

**Lucent Technologies**  
Bell Labs Innovations



# UCS 1000

## R4.6

### Speech Development, Processing, and Recognition

585-313-223  
Comcode 108762063  
August 2000  
Issue 1



Copyright © 2000 by Lucent Technologies. All rights reserved.

For trademark, regulatory compliance, and related legal information, see the copyright and legal notices section of this document.

# Copyright and Legal Notices

## Copyright

Copyright © 2000 by 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.

## Lucent Technologies Web Page

The world wide web home page for Lucent Technologies is:  
<http://www.lucent.com>

## Preventing Toll Fraud

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

## Lucent Technologies Fraud Intervention

If you suspect you are being victimized by toll fraud and you need technical support or assistance, call the appropriate BCS National Customer Care

Center telephone number. Users of the MERLIN®, PARTNER®, and System 25 products should call 1 800 628-2888. Users of the System 75, System 85, DEFINITY® Generic 1, 2 and 3, and DEFINITY® ECS products should call 1 800 643-2353.

## Providing Telecommunication Security

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

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

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

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

- Utilization (of capabilities special to the accessed equipment)
- Theft (such as, of intellectual property, financial assets, or toll-facility access)

- Eavesdropping (privacy invasions to humans)
- Mischief (troubling, but apparently innocuous, tampering)
- Harm (such as harmful tampering, data loss or alteration, regardless of motive or intent)

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

### **Your Responsibility for Your Company's Telecommunication Security**

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

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

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

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

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

**Part 15: Class A Statement.**

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

### **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. For the CWB2/CYD2, this number is AS5USA-27438-XD-E; registration for the CWB10/CYD10 is pending at the time of this publication.

### **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

## Industry Canada (IC) Interference Information

This digital apparatus does not exceed the Class A limits for radio noise emissions set out in the radio interference regulations of Industry Canada.

Le Présent Appareil Numérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la class A prescrites dans le règlement sur le brouillage radioélectrique édicté par le Industrie Canada.

## Trademarks

Lucent Technologies has made every effort to supply the following trademark information about company names, products, and services mentioned in the UCS 1000 R4.6 documentation library:

- Adobe Systems, Inc. — Trademarks: Adobe, Acrobat.
- Enhanced Software Technologies, Inc. — Trademark: Quickstart.
- Equinox Systems, Inc — Registered trademark: Equinox
- Hewlett Packard Corporation — Registered trademarks: Hewlett-Packard and HP
- Intel Corporation — Registered trademarks: Pentium.
- International Business Machines Corporation — Registered trademarks: IBM, VTAM.
- Lucent Technologies — Registered trademarks: 4ESS, 5ESS, AUDIX, CONVERSANT, DEFINITY, Voice Power. Trademarks: FlexWord, Intuity, Lucent.

- Microsoft Corporation — Registered trademarks: Excel, Internet Explorer, Microsoft, MS, MS-DOS, Windows, Windows NT.
- Mylex Corporation — Registered trademark: Mylex.
- Novell, Inc. — Registered trademarks: Novell.
- Oracle Corporation — Trademarks: OBJECT\*SQL, ORACLE, ORACLE\*Terminal, PRO\*C, SQL\*FORMS, SQL\*Menu, SQL\*Net, SQL\*Plus, SQL\*ReportWriter.
- PCI Industrial Computer Manufacturers Group — Registered trademarks: CompactPCI and PICMG.
- Santa Cruz Operation, Inc. — Registered trademarks: UnixWare.
- Sun Microsystems — Registered trademarks: Sun, Sun Microsystems, Sun Workstation, Solaris (computer and peripherals). Trademarks: Solaris (operating system utilities) and Java
- UNIX System Laboratories, Inc. — Registered trademarks: UNIX.
- Xerox Corporation — Trademarks: Ethernet.

## Ordering Information

### Call or Write

Lucent Technologies Publications Center  
2855 N. Franklin Road  
Indianapolis, IN 46219 U.S.A

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

Documents can also be ordered from the Customer Information Centre in Malmesbury, England.

Voice	44 1666 83-2900	FAX	44 1666 83-2213
-------	-----------------	-----	-----------------

For additional documents, refer to the section in “About This Document” entitled “Related Resources.”

You can be placed on a standing order list for this and other documents you may need. 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.

## Obtaining Products

To learn more about Lucent Technologies products and to order products, contact Lucent Direct, the direct-market organization of Lucent Technologies Business Communications Systems. Access their Web site at [www.lucent.direct.com](http://www.lucent.direct.com). Or call the following numbers: customers 1 800 451-2100, account executives 1 888 778-1880 (voice) or 1 888 778-1881 (fax).

<b>Warranty</b>	Lucent Technologies provides a limited warranty on this product. Refer to the "Limited Use Software License Agreement" card provided with your package.
<b>European Union Declaration of Conformity</b>	<p>The "CE" mark affixed to the equipment means that it conforms to the directives below. Lucent Technologies Business Communications Systems declares that the UCS 1000 equipment specified in this document conforms to the referenced European Union (EU) Directives and Harmonized Standards listed below:</p> <ul style="list-style-type: none"><li>• EMC Directive 89/336/EEC</li><li>• Low-Voltage Directive 73/23/EEC</li></ul>
<b>Comments</b>	To comment on this document, return the comment card at the front of the document.
<b>Acknowledgment</b>	This document was prepared by Product Documentation Development, Lucent Technologies, Columbus, OH.



## Copyright and Legal Notices

iii

Copyright . . . . .	iii
Notice . . . . .	iii
Lucent Technologies Web Page . . . . .	iii
Preventing Toll Fraud . . . . .	iii
Lucent Technologies Fraud Intervention . . . . .	iii
Providing Telecommunication Security . . . . .	iv
Your Responsibility for Your Company's Telecommunication Security . . . . .	v
Federal Communications Commission Statement . . . . .	vi
Industry Canada (IC) Interference Information . . . . .	viii
Trademarks . . . . .	viii
Ordering Information . . . . .	x
Obtaining Products . . . . .	x
Warranty . . . . .	xi
European Union Declaration of Conformity. . . . .	xi
Comments . . . . .	xi
Acknowledgment. . . . .	xi

## About This Book

xxviii

Overview . . . . .	xxviii
Intended Audiences . . . . .	xxix
Updates to the Product . . . . .	xxix
How This Book Is Organized. . . . .	xxx
Conventions Used in This Book . . . . .	xxxii
Terminology. . . . .	xxxii
Keyboard and Telephone Keypad Representations . . . . .	xxxvii
Screen Displays . . . . .	xxxviii
Some Screen Simulations. . . . .	xxxix
Items That May or May Not Appear . . . . .	xxxix
Other Typography . . . . .	xl
Command Text . . . . .	xl
Cross-References and Hypertext . . . . .	xli
Safety and Security Alert Labels. . . . .	xli
Related Resources. . . . .	xlii
Updates to the Product . . . . .	xlii
Training . . . . .	xlii
Documentation . . . . .	xlii
Obtaining Printed Versions of the Documentation . . . . .	xlii
Using the CD-ROM Documentation. . . . .	xliii
Setting the Default Magnification . . . . .	xliii
Adjusting the Window Size . . . . .	xliii

Hiding and Displaying Bookmarks . . . . .	xliii
Using the Button Bar . . . . .	xliv
Using Hypertext Links . . . . .	xliv
Navigating with Double Arrow Keys . . . . .	xliv
Searching for Topics . . . . .	xliv
Displaying Figures. . . . .	xliv
Printing the Documentation. . . . .	xliv
How to Comment on This Book . . . . .	xlvi

## **1 Overview of Speech 1**

Overview . . . . .	1
The Speech File System . . . . .	2
Speech Encoding and Storage . . . . .	2
Default Speech Directory . . . . .	4
Specifying a Speech Directory Procedure . . . . .	4
Saving and Restoring Speech Files . . . . .	5
Adding a Supplemental Speech Disk . . . . .	5
Recovering from a Corrupted Speech Disk. . . . .	6
Defining Phrases. . . . .	6
Defining TSM Phrase Numbers . . . . .	6
Defining Talkfiles . . . . .	6
Defining Speech Files . . . . .	7
Defining the Speech File System . . . . .	7

Speech Development Features . . . . .	7
The irAPI Application Development Tool . . . . .	8
Text-to-Speech Features . . . . .	9
Standard Text-to-Speech (TTS) . . . . .	9
Proxy Text-to-Speech (PTTS) . . . . .	12
Speech Recognition Features . . . . .	14
WholeWord Speech Recognition . . . . .	15
FlexWord Speech Recognition . . . . .	17
Dial Pulse Recognition . . . . .	18
The SSP Circuit Card . . . . .	19
Description . . . . .	19
Capacity . . . . .	19
LSPS II Circuit Card . . . . .	22
Description . . . . .	22
Capacity . . . . .	22

## **2 Developing Speech 25**

Overview . . . . .	25
Determining the Transaction . . . . .	27
Background . . . . .	27
References . . . . .	27
Planning the Voice Script . . . . .	28

Writing the Voice Script . . . . .	30
Using Framing in Voice Scripts . . . . .	31
Analyzing Speech Inflections . . . . .	32
Placing Frame Words . . . . .	32
Selecting Speech Sounds for Framing . . . . .	33
Including Voiceless Speech Sounds . . . . .	33
Selecting a Speech Development Method . . . . .	34
Hiring a Professional Speaker . . . . .	35
Purchasing a Lucent Technologies Custom Speech Package . . . . .	36
Using TTS (Text-to-Speech) . . . . .	37
Recording Speech . . . . .	39
Working with a Professional Speaker . . . . .	39
Working with Lucent Technologies to Develop a Custom Speech Package . . . . .	42
Encoding Speech Phrases . . . . .	43
Installing Speech . . . . .	44
Installing Speech in an IRAPI Application . . . . .	44
<b>3 Recognizing WholeWord Speech Input on the SSP</b>	<b>46</b>
Overview . . . . .	46
How WholeWord Speech Recognition Works . . . . .	47
Overview . . . . .	47

Caller Response . . . . .	47
The Recognizer . . . . .	48
Computational Processing . . . . .	48
Further Processing . . . . .	49
WholeWord Hardware and Software . . . . .	49
Languages Supported . . . . .	49
Bilingual Capabilities . . . . .	50
Standard Vocabulary . . . . .	50
Standard WholeWord Speech Recognition Types . . . . .	52
Description . . . . .	52
Models and Speaker Independence . . . . .	52
Grammars . . . . .	53
Keyword Recognition Types . . . . .	53
Connected-Digit Recognition Types . . . . .	54
Connected-Digit Recognition . . . . .	55
Fixed-Length versus Variable-Length Connected Digits . . . . .	55
Recognition DIPS . . . . .	56
Word Spotting . . . . .	57
Phrase Screening . . . . .	58
Recognition Confirmation . . . . .	58
Barge-In . . . . .	59
Description . . . . .	59
Enabling or Disabling Barge-In . . . . .	60

Discouraging Barge-In . . . . .	60
Encouraging Barge-In . . . . .	61
System Response for Barge-In . . . . .	61
Dual Tone Multifrequency (DTMF) Support . . . . .	62
Custom Grammars. . . . .	63
Custom Vocabulary . . . . .	64
Custom DIPs. . . . .	65
WholeWord Speech Recognition Accuracy . . . . .	65
Overview . . . . .	65
Positive Influences on WholeWord Speech Recognition Accuracy . . . . .	65
Negative Influences on WholeWord Speech Recognition Accuracy . . . . .	69
The Recognizer. . . . .	73
Computational Processing . . . . .	73

## **4 Recognizing FlexWord™ Speech Input on the SSP Circuit**

### **Card**

**74**

Overview . . . . .	74
Description . . . . .	75
How FlexWord Speech Recognition Works . . . . .	75
FlexWord Capacity. . . . .	78
Vocabularies . . . . .	78
Keyword Spotting . . . . .	79

Phrase Screening . . . . .	79
Barge-in . . . . .	80
Designing a FlexWord Speech Recognition Application . . . . .	80
Defining Wordlists . . . . .	80
Using Wordlists in Scripts . . . . .	81
FlexWord Software and Hardware . . . . .	84
FlexWord Recognition Vocabulary Administration . . . . .	85
Copying Wordlists . . . . .	85
Installing and Activating Wordlists from Diskette . . . . .	86
Activating Wordlists That Reside on the System . . . . .	87
Deactivating Wordlists . . . . .	88
Removing Wordlists . . . . .	89
FlexWord Speech Recognition Accuracy . . . . .	90
Positive Influences on FlexWord Speech Recognition Accuracy . . . . .	90
Negative Influences on FlexWord Speech Recognition Accuracy . . . . .	95

## **5 Recognizing Dial Pulse Input 97**

Overview . . . . .	97
Hardware . . . . .	98
Simultaneous Dial Pulse Capabilities . . . . .	98
Dial Pulse Recognition Types . . . . .	98

Grammars . . . . .	100
Recognition Confirmation . . . . .	101
Barge-In . . . . .	102
How Dial Pulse Recognition Works . . . . .	102
The Recognizer . . . . .	102
Recognition Accuracy . . . . .	103
Training . . . . .	103
Limitations . . . . .	104
DPR Troubleshooting . . . . .	105
Common Failure Modes . . . . .	106

## **6 Putting It Together with the SSP Circuit Card 108**

Overview . . . . .	108
Using WholeWord Recognition and Dial Pulse Recognition Together . . . . .	109
Getting the Most Out of FlexWord Speech Recognition . . . . .	109
Choosing a FlexWord Speech Recognition Vocabulary . . . . .	110
Caller Error . . . . .	110
Word Length . . . . .	111
Word Choice . . . . .	111
Prompting for Input . . . . .	111
Word Sounds . . . . .	111
Synonyms . . . . .	111

Examples of FlexWord Speech Recognition Applications . . . . .	112
Getting the Most out of Dial Pulse Recognition . . . . .	114
Getting the Most out of Text-to-Speech . . . . .	115
Using FlexWord Speech Recognition and Text-to-Speech Together . . . . .	116
Using Text-to-Speech and Prerecorded Speech Together. . . . .	117
Using WholeWord and FlexWord Speech Recognition Together . . . . .	118
WholeWord Speech Recognition Versus FlexWord Speech Recognition Accuracy	120
Comparison of Recognition Types. . . . .	121

## **7 Recognizing WholeWord Speech Input on the LSPS II 123**

Overview . . . . .	123
How WholeWord Speech Recognition Works . . . . .	124
Overview . . . . .	124
Caller Response . . . . .	124
The Recognizer. . . . .	125
Computational Processing . . . . .	126
WholeWord Hardware and Software. . . . .	126
Languages Supported . . . . .	126
Standard Vocabulary. . . . .	127
Standard WholeWord Speech Recognition Types. . . . .	128
Description . . . . .	128
Models and Speaker Independence . . . . .	129

Grammars . . . . .	129
Keyword Recognition Types . . . . .	130
Connected-Digit Recognition Types . . . . .	131
Connected-Digit Recognition . . . . .	132
Fixed-Length versus Variable-Length Connected Digits . . . . .	132
Recognition DIPs . . . . .	133
Word Spotting . . . . .	134
Phrase Screening . . . . .	134
Recognition Confirmation . . . . .	135
Barge-In . . . . .	135
Description . . . . .	135
Enabling or Disabling Barge-In . . . . .	136
Discouraging Barge-In . . . . .	137
Encouraging Barge-In . . . . .	137
System Response for Barge-In . . . . .	138
Dual LSPS Tone Multifrequency (DTMF) Support . . . . .	138
Custom Grammars. . . . .	139
Custom Vocabulary . . . . .	140
Custom DIPs. . . . .	141
WholeWord Speech Recognition Accuracy . . . . .	141
Overview . . . . .	141
Positive Influences on WholeWord Speech Recognition Accuracy . . . . .	141

## **8 Recognizing FlexWord™ Speech Input on the LSPS II Circuit Card** **149**

Introduction . . . . .	149
Description . . . . .	150
How FlexWord Speech Recognition Works . . . . .	151
Phonemes . . . . .	151
FlexWord Capacity . . . . .	153
Creating a Grammar . . . . .	155
Vocabularies . . . . .	155
Keyword Spotting . . . . .	155
Phrase Screening . . . . .	155
Barge-in . . . . .	156
Designing a Grammar . . . . .	156
FlexWord Software and Hardware . . . . .	156
Configuring the LSPS II Circuit Card . . . . .	157
Writing the Grammar . . . . .	158
Compiling the Grammar . . . . .	160
Editing Board Type Files . . . . .	161
Assigning the Basic Flex to the LSPS II Circuit Card . . . . .	162
Assigning the Basic ASR Grammar with Flex . . . . .	172
Assigning the Basic ASR Grammar with Flex and TTS . . . . .	182

Assigning Basic Flex and TTS . . . . .	192
LSPS Usage Data Files . . . . .	202
Usage Data Files That are Changeable . . . . .	203
Editing a Usage Data File . . . . .	205
Example . . . . .	205

## **9 Putting It Together with the LSPS II Circuit Card 208**

Overview . . . . .	208
Using WholeWord Recognition . . . . .	209
Getting the Most Out of FlexWord Speech Recognition . . . . .	209
Choosing a FlexWord Speech Recognition Vocabulary . . . . .	210
Caller Error . . . . .	210
Word Length . . . . .	211
Word Choice . . . . .	211
Prompting for Input . . . . .	211
Word Sounds . . . . .	211
Synonyms . . . . .	211
Examples of FlexWord Speech Recognition Applications . . . . .	212
Getting the Most out of Text-to-Speech . . . . .	214
Using FlexWord Speech Recognition and Text-to-Speech Together . . . . .	215
Using Text-to-Speech and Prerecorded Speech Together. . . . .	216
Using WholeWord and FlexWord Speech Recognition Together . . . . .	217

WholeWord Speech Recognition Versus FlexWord Speech Recognition Accuracy	219
Comparison of Recognition Types . . . . .	220

## **Appendix A: Speech File Formats** **221**

Overview . . . . .	221
Speech File Formats . . . . .	222
PCM Encoding Formats . . . . .	223
ADPCM Encoding Formats . . . . .	223
Speech Files . . . . .	225

## **Appendix B: Advanced Text-to-Speech Features** **228**

Overview . . . . .	228
Escape Sequences Used by the SSP Circuit Card . . . . .	229
Inserting Silent Delays and Pauses . . . . .	229
Examples for Silence— Using Text . . . . .	231
Examples for Silence— Using a Variable . . . . .	232
Changing the Gender of the Speaking Voice . . . . .	233
Changing the Rate of Speech . . . . .	234
Setting Text Classes and Detectors . . . . .	235
Background . . . . .	235
Classes . . . . .	235
Class Detector Modes . . . . .	235
Guidelines . . . . .	236

Escape Sequences Used by the LSPS II Circuit Card. . . . .	240
Changing Text Class Detection Mode. . . . .	240
Switching to Special Translation Mode. . . . .	242
Changing Text Expansion Modes. . . . .	243
Operating on User Dictionaries. . . . .	244
Using Other Escape Sequences. . . . .	244

<b>Glossary</b>	<b>248</b>
-----------------	------------

<b>Index</b>	<b>322</b>
--------------	------------

## Overview

This document describes the procedures needed to perform full system administration for the following areas:

- The UNIX operating system
- The voice system
- Databases
- Switch interfaces

This document also describes the procedures needed to create and access system reports and to monitor the system.

## Intended Audiences

The primary audience for this document are system administrators. This includes:

- On-site technicians who perform system administration at the customer site
- End customers who choose to administer their own systems

Secondary audiences include the field support personnel.

We assume that the primary users of this book have completed the UCS 1000 R4.6 hardware installation and maintenance training course. See [Using the CD-ROM Documentation on page xliii](#) for more information.

## Updates to the Product

The following Web site displays any updates or exceptions to the product that have occurred after the publication of this document:

**<http://glsdocs.lucent.com>**

# How This Book Is Organized

This document is designed to take you step by step through the system administration process. Each chapter contains procedures for a specific product area that requires administration.

This book is contains the following sections:

- [Chapter 1, Overview of Speech](#) — Provides an overview of speech on the server, information about the speech file system and a discussion of all of the speech-related features available for the development of voice response applications.
- [Chapter 2, Developing Speech](#) — Describes the procedures involved in producing speech, from determining the transaction and planning the script to recording and encoding the speech.
- [Chapter 3, Recognizing WholeWord Speech Input on the SSP](#) — Discusses the concepts behind WholeWord speech recognition and includes sections on creating WholeWord vocabularies and improving the accuracy of speech recognition as it pertains to the SSP circuit card.
- [Chapter 4, Recognizing FlexWord™ Speech Input on the SSP Circuit Card](#) — Includes a detailed discussion of FlexWord speech recognition and its features as it pertains to the SSP circuit card.

- [Chapter 5. Recognizing Dial Pulse Input](#) — Discusses the recognition of dial pulses on rotary or push button telephones using the SSP circuit card.
- [Chapter 6. Putting It Together with the SSP Circuit Card](#) — Discusses how to make the various speech-related features and work well together in a voice response application using the SSP circuit card.
- [Chapter 7. Recognizing WholeWord Speech Input on the LSPS II](#) — Discusses the concepts behind WholeWord speech recognition and includes sections on creating WholeWord vocabularies and improving the accuracy of speech recognition using the LSPS II circuit card.
- [Chapter 8. Recognizing FlexWord™ Speech Input on the LSPS II Circuit Card](#) — Includes a detailed discussion of FlexWord speech recognition and its features using the LSPS II circuit card.
- [Chapter 9. Putting It Together with the LSPS II Circuit Card](#) — Discusses how to make the various speech-related features and work well together in a voice response application using the LSPS II circuit card.
- [Appendix A. Speech File Formats](#) — Includes information about speech files and speech file formats, including pulse code modulation (PCM) and adaptive differential pulse code modulation (ADPCM).
- [Appendix B. Advanced Text-to-Speech Features](#) — Discusses how to customize synthesized speech by using escape sequences to add silence delays, change the speaking rate, select the speaking voice gender, and mark text as belonging to a specific text category.

- [Glossary](#) — Defines the terms, abbreviations, and acronyms used in system documentation.
- [Index](#) — Alphabetically lists the principal subjects covered in the book.

## Conventions Used in This Book

This section describes the typography and other conventions that are used in this book.

**Note:** The screens shown in this book are examples only. The screens you see on your system may be similar, but not exactly the same in all cases.

### Terminology

- The word “type” means to press the key or sequence of keys specified. For example, an instruction to type the letter “y” is shown as  
Type **y** to continue.
- The word “enter” means to type a value and then press the **ENTER** key on the keyboard. For example, an instruction to type the letter “y” and press **ENTER** is shown as  
Enter **y** to continue.

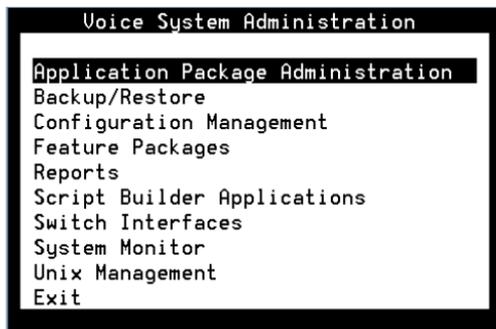
- The word “select” means to move the cursor to the desired menu item and then press **ENTER**. For example, an instruction to move the cursor to the Configuration Management option on the Voice Administration System screen and then press **ENTER** is shown as

Select:

```
> Configuration Management
```

- The system displays menus, screens, and windows. Menus ([Figure 1](#)) present options from which you can choose to view another menu, or a screen or window. Screens and windows both show ([Figure 2 on page xxxv](#) and [Figure 3 on page xxxvi](#)) and request ([Figure 4 on page xxxvi](#) and [Figure 5 on page xxxvii](#)) system information.

Figure 1. Example of a UCS 1000 R4.6 Menu



**Figure 2. Example of a UCS 1000 R4.6 Screen Showing Information**

```
UnixWare Installation           Primary Hard Disk Partitioning

In order to install LINCOS, you should reserve a UNIX system
partition (a portion of your hard disk's space) containing 100%
of the space on your primary hard disk. 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 partitions(s).

The UNIX system partition that you intend to use on the primary
hard disk must be at least 4200 MBs and labeled "ACTIVE."

Press 'ENTER' to continue
```

**Figure 3. Example of a UCS 1000 R4.6 Window Showing Information**

```
3 Shut Down the Machine
Users currently logged on:
root      console    Jul  8 06:39
root      pts000     Jul  8 08:48
```

**Figure 4. Example of a UCS 1000 R4.6 Screen Requesting Information**

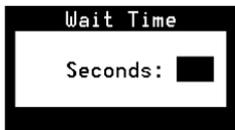
```
UNIX System Installation                               Set Slice Sizes

Please select whether you would like the recommended slice
sizes or would like to customize the slice sizes.

Your choices are:
1. Recommended Slice Sizes
2. Customize Slice Sizes

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

Figure 5. Example of a UCS 1000 R4.6 Window Requesting Information



### Keyboard and Telephone Keypad Representations

- Keys that you press on your terminal or PC are represented as capitalized **BOLD** text. For example, an instruction to press the Enter key is shown as  
Press **ENTER**.
- Two or three keys that you press at the same time on your terminal or PC (that is, you hold down the first key while pressing the second and/or third key) are represented as a series of small, capitalized **BOLD** text separated by the plus sign (+). 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 capitalized **BOLD** text followed by the function or value of that key enclosed in parentheses. For example, an instruction to press function key 2 is shown as  
Press **F2** (Choices).

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

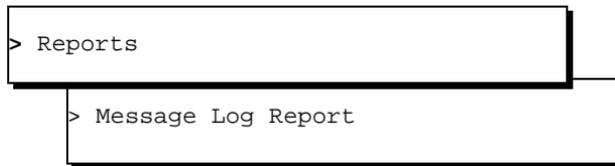
Press **1** to record a message.

## Screen Displays

- System messages, field names, and prompts that appear on the screen are shown in `typewriter text`, as shown in the following examples:
  - ~ Enter the number of ports to be dedicated to outbound traffic in the `Maximum Simultaneous Ports` field.
  - ~ Enter **y** in the `Message Transfer?` field.
  - ~ The system displays the following message:  
`Installation in progress.`
- The sequence of menu options that you must select to display a specific screen or submenu appears in a series of boxes.

Example:

Start at the Voice System Administration menu and select:



In this example, you would access the Voice System Administration menu and select the Reports menu. From the Reports menu, you would then select the Message Log Report window.

### Some Screen Simulations

Text in a simulated screen display appears in `typewriter text`.

Example:

```
QuickStart - Data Recovery Rescue
Copyright(c) 1997-1999 by Enhanced Software Technologies, Inc.
Serial# 8200-999                               Version: 1.3.17

Backup System  Verify System  Recover System  Configure QuickStart  Exit and Reboot
```

### Items That May or May Not Appear

**Grayed-out** type represents optional items that may or may not appear in a given display.

Example:

Once the backup is complete, the system displays a message similar to the following:

The Differential UNIX backup is now complete. Please remove the tape and label it as "Differential UNIX Backup, created **August 30, 1999.**"

## Other Typography

### Command Text

- Literal values, commands, and text that 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 in italics when they are not part of the command line, for example:

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

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

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

**Cross-References  
and Hypertext**

[Blue, underlined](#) type indicates a cross reference or hypertext link that takes you to another location in the document when you click on it.

## Safety and Security Alert Labels

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

 **CAUTION:**

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

 **WARNING:**

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

 **DANGER:**

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

**SECURITY ALERT:**

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

## Related Resources

### Updates to the Product

The following Web site displays any updates or exceptions to the product that have occurred after the publication of this document:

**<http://glsdocs.lucent.com>**

### Training

For more information on UCS 1000 R4.6 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) 256-8988

### Documentation

Appendix A, “Documentation Guide,” in *UCS 1000 R4.6 System Description*, 585-313-222, provides an overview of all the books in the UCS 1000 R4.6 library that are mentioned in this book.

### Obtaining Printed Versions of the Documentation

See [Printing the Documentation on page xliiv](#) for information on how to print this document.

You can also order the printed documents by calling 1-888-582-3688 or visiting the Customer Information Center (CIC) website at:

[http://www.lucent.com/cgi-bin/CIC\\_store.cgi](http://www.lucent.com/cgi-bin/CIC_store.cgi)

## Using the CD-ROM Documentation

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

### Setting the Default Magnification

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

### Adjusting the Window Size

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

### Hiding and Displaying Bookmarks

By default, the document appears with bookmarks displayed on the left side of the screen. The bookmarks serve as a hypertext table of contents for the

chapter you are viewing. You can control the appearance of bookmarks by selecting **View | Page Only** or **View | Bookmarks and Page**.

### Using the Button Bar

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

### Using Hypertext Links

Hypertext-linked text appears in blue, italics, and underlined. These links are shortcuts to other sections or books.

### Navigating with Double Arrow Keys

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

### Searching for Topics

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

### Displaying Figures

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

### Printing the Documentation

**Note:** For information on ordering printed copies of the documents, see [Obtaining Printed Versions of the Documentation on page xlii](#).

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

### Printing an Entire Document

To print an entire document:

- 1 From the documentation main menu screen, select one of the print-optimized documents. Print-optimized documents print two-screens to a side, both sides of the sheet on 8.5x11-in or A4 paper.
- 2 Select **File | Print**.
- 3 Enter the page range you want to print, or select **All**. Note that the print page range is different from the page numbers on the documents (they print two to a page).
- 4 After the document is printed, close the file. Do not leave this file open while viewing the electronic documents.

### Printing Part of a Document

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

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

The document is printed, one screen per side, two sides per sheet.

## How to Comment on This Book

A comment card is included at the back of this book. While we have tried to make this document fit your needs, we are interested in your suggestions for improving it and urge you to complete and return a comment card to us. If the comment card has been removed from this book, send your comments to the address below. Be sure to include the name and document number of this book, *UCS 1000 R4.6 Administration*, 585-313-509.

Lucent Technologies  
GLS Information Development Division  
Room 22-2H15  
11900 North Pecos Street  
Denver, Colorado 80234-2703 US

You may also fax your comments to the attention of the Lucent Technologies UCS 1000 R4.6 writing team at (303) 538-1741.

# 1 Overview of Speech

## Overview

This chapter provides:

- Information about the speech file system
- Information about speech terminology
- An overview of the features available for speech development

This chapter is also designed to ensure that you:

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

## The Speech File System

All speech to be played as part of an application resides as UnixWare files in a mounted UnixWare file system. By default, speech file systems reside in **/voice1**, which is linked to **/home2**. These two file systems can be used interchangeably. With the UCS 1000 R4.2 system, you can define where in **/voice1** or **/home2** speech files are stored.

### Speech Encoding and Storage

Each speech phrase requires a minimum of 8 Kbytes of space. These 8-Kbyte blocks hold different lengths of speech depending on the encoding rate, which is in turn dependent on which circuit card you are using to encode the speech. Currently, three circuit cards are in use for speech coding and playback on UCS 1000 R4.6 systems:

- Speech and Signal Processor (SSP) circuit card
- Lucent Speech Processing Solutions (LSPS II) circuit card
- Quad circuit card

See [Table 1](#) to estimate how much speech is in each speech phrase block for the SSP and Quad circuit cards.

**Table 1. SSP and Quad Circuit Card Encoding Rates and Speech Blocks**

Coding Rate	Seconds of Speech per 8-Kbyte Block
ADPCM32	2.0
ADPCM64	1.0
CELP16	4.0

See [Table 2](#) to estimate how much speech is in each speech phrase block for the LSPS II circuit card.

**Table 2. LSPS II Encoding Rates and Speech Phrase Blocks**

Coding Rate	Seconds of Speech per 8-Kbyte Block
OKI ADPCM32	2.0
MuLaw	1.0

See [Appendix A. Speech File Formats](#) for detailed information about coding rates. For information about the encoding methods and capacities, see “Speech Storage Capacities” and “Speech Administration Capacities” in Chapter 4, “Features,” of *UCS 1000 R4.6 System Description*, 585-313-222.

### 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 Procedure

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

- 1 Stop the voice system.

For the procedure to do so, see “Common System Procedures” in *UCS 1000 R4.6 System Reference*, 585-313-221.

- 2 Access the **/vs/data/irAPI.rc** file.

- 3 Add the following entry:

**SPEECHDIR=directory**

where *directory* is the full path of the new directory where you want to store speech files.

The SPEECHDIR variable specifies the new directory and contains default coding style for all speech.

- 4 For LSPS II, add only *one* of the following entries as a global play and record default:

**SPEECH\_ALGO=IRA\_MULAW**  
**SPEECH\_ALGO=IRA\_OKIPCM32**

### 5 Restart the voice system.

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

**Note:** These speech administration tools are not supported for the LSPS II circuit card.

### Saving and Restoring Speech Files

Speech files are backed up when a **mkimage** system backup is performed. The **spsav** (save) and **spres** (restore) commands are also available for backing up speech. See *UCS 1000 R4.6 Administration*, 585-313-509, for information about these commands.

### CAUTION:

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

### Adding a Supplemental Speech Disk

Depending on your needs and system configuration, there may be situations in which you want to add a supplemental hard disk specifically for storing speech. In this case, the additional disk in the system is used to store speech file systems that reside in **/home3**.

For the instructions to install such a supplemental hard disk, see "Adding a Hard Disk Drive for Speech" in "Replacing the Hard Disk Drive Assembly" in *UCS 1000 R4.6 Maintenance*, 585-313-154.

### Recovering from a Corrupted Speech Disk

For information on recovering from a corrupted speech disk, see "Replacing a Hard Disk Drive" in *UCS 1000 R4.6 Maintenance*, 585-313-154.

### Defining Phrases

A *phrase* is a unit of speech, such as letter, number, word, sentence, or paragraph, that a speech application script speaks to a caller. Examples of phrases include a welcome message, a bank balance, or the name of a month. The application speaks a phrase to callers by referencing either TSM phrase number, filename, or an irAPI filename. See [Defining TSM Phrase Numbers on page 6](#) for more information.

### Defining TSM Phrase Numbers

A *TSM 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 the phrase number. A phrase number is assigned to a phrase when you add the phrase to an application. See *UCS 1000 R4.6 Application Development with Advanced Methods*, 585-313-225, for more information.

### Defining Talkfiles

A *talkfile* is a directory. The default path is **/home2/vfs/talkfiles**. In this directory you will find subdirectories representing all the talkfiles. The actual phrases are located in **/home2/vfs/talkfiles/<talkfile#>**, where *talkfile#* represents the actual talkfile.

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

### Defining Speech Files

A *speech file* is a file containing an encoded speech phrase. Speech files can be stored anywhere, although the default speech file system is located in the **/home2/vfs/talkfiles** directory.

### Defining the Speech File System

A *speech file system* is a mounted UnixWare file system where speech resides and is defined in the **irAPI.rc** file. Only one speech file system can be active at a given time. See the various topics under [The Speech File System on page 2](#) for more information.

## Speech Development Features

Nearly all applications involve playing speech to a caller. The irAPI application development tool is available to use for the creation, editing, recognition, and inclusion of speech in an application.

**Note:** See *UCS 1000 R4.6 Application Development with Advanced Methods*, 585-313-225, for further details on the irAPI application development tool.

The following speech development features are available for the inclusion, editing, and recognition of speech in an application.

- The irAPI Application Development Tool
- Text-to-Speech Features
  - ~ Standard Text-to-Speech (TTS)
  - ~ Proxy Text-to-Speech (PTTS)
- Speech Recognition Features
  - ~ WholeWord Speech Recognition
  - ~ FlexWord™ Speech Recognition
  - ~ Dial Pulse Recognition (SSP only)

## The irAPI Application Development Tool

For information about developing applications using the irAPI application development tool, see *UCS 1000 R4.6 Application Development with Advanced Methods*, 585-313-225.

**Note:** The LSPS II circuit card supports no more than 64 calls to irSay(3IRAPI) or irPlay(3IRAPI) functions. The application must call the irEnd(3IRAPI) after 64 irSay(3IRAPI) or irPlay(3IRAPI) calls, or unexpected results can occur.

## Text-to-Speech Features

The UCS 1000 R4.6 system has a number of features available that can greatly enhance your ability to use it effectively. Among these are features that allow you to convert text input to spoken output (standard Text-to-Speech and Proxy Text-to-Speech)

### Standard Text-to-Speech (TTS)

The Text-to-Speech (TTS) feature allows you to include speech in an application by providing text as input and allowing the speech and signal processing (SSP) circuit card or the Lucent Speech Processing Solutions (LSPS II) circuit card to produce synthesized speech. Currently TTS is available only for US English (whereas Proxy Text-to-Speech provides similar capabilities both for US English and for other languages — for more information, see the *Proxy Text-Speech (PTTS) User Guide*, 585-350-115). The default gender for the speaking voice is male, although a female voice can be specified.

Text that TTS speaks can either be typed at the keyboard or obtained from a file or database.

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

- Filters the text to identify sentence and phrase boundaries
- Identifies words as a parts of speech (noun, verb, preposition, and so on)

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

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

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

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

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

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

### Requirements

The TTS feature requires:

- One of the following Text-to-Speech feature software packages:
  - ~ For the SST circuit card, **tts**
  - ~ For the LSPS II circuit card, **ttsenna**
- An SSP or LSPS II circuit card with TTS functionality assigned

### References for use with the SSP Circuit Card

- To use TTS with TAS applications, see Appendix B, "Summary of TAS Script Instructions," in *UCS 1000 R4.6 Application Development with Advanced Methods*, 585-313-225.

- To install the TTS feature software for use with the SSP circuit card, see “Installing the Text-to-Speech Package” in Chapter 7, “Installing the Optional Feature Software” in *UCS 1000 R4.6 Maintenance*, 585-313-154.
- For information on SSP circuit card functions, see [The SSP Circuit Card on page 19](#).

### References for use with the LSPS II Circuit Card

- To use TTS with TAS applications, see Appendix B, “Summary of TAS Script Instructions,” in *UCS 1000 R4.6 Application Development with Advanced Methods*, 585-313-225.
- The TTS software package for the LSPS II circuit card is included in the LSPSset software package. See Chapter 9, “Installing LSPS II Optional Software Packages,” of *UCS 1000 R4.6 Maintenance*, 585-313-154.
- For information on the LSPS II circuit card functions, see [Commands for SSP Circuit Card Only on page 21](#)

### Proxy Text-to-Speech (PTTS)

In applications where the demand for TTS is very high or where a language is needed that is not supported on the SSP or LSPS II circuit card, text-to-speech processing must be done using one or more auxiliary computers connected to the UCS 1000 R4.6 system in a client/server configuration. This capability is called Proxy Text-to-Speech (PTTS).

The current release of the PTTS feature for UCS 1000 R4.6 supports only the L&H Microsoft Speech Application Programming Interface (SAPI)-compliant languages, which typically include English and most western European and Latin American languages

A complete list of the currently supported languages is included with the PTTS software package.

With the open architecture provided by this feature, you can also add other customized languages, possibly with the assistance of an independent software vendor (ISV).

The PTTS feature consists of software that is installed on a UCS 1000 R4.6 client system and on one or more Ziatech PTTS server cards running the Windows NT 4.0 operating system. These systems communicate through socket interface connections, called PTTS connections, over an Ethernet Local Area Network (LAN). Multiple UCS 1000 R4.6 clients and their associated PTTS servers can be placed on a single LAN.

The UCS 1000 R4.6 client establishes the PTTS connections and submits text to a PTTS server for conversion. The PTTS server converts the text to speech encoded in the PCM64 format (pulse code modulation at 64 kbps using Mu-law encoding) and returns the encoded speech to the UCS 1000 R4.6 client.

The PTTS feature supports multiple languages, and even the use of multiple languages within a single IVR application. Characteristics of the PTTS speaking voice, such as gender, rate of speech, volume, pitch, and intonation

can be customized. Barge-in (talkoff) can also be enabled to allow a caller to interrupt speech playback.

PTTS is licensed on a per-connection basis. The PTTS and TTS features share the same licensing, which means that, if you already have TTS licenses, you can use them for this feature as well.

### Requirements

The PTTS feature requires:

- The Proxy Text-to-Speech feature software package
- One or more Ziatech PTTS server cards
- TTS licensing assigned

### Reference

For more information about using PTTS, see the *Proxy Text-to-Speech (PTTS) User Guide*, 585-310-115.

## Speech Recognition Features

Speech recognition is a feature that allows the system to recognize and respond to spoken voice responses from the caller. The UCS 1000 R4.6 system offers both WholeWord and FlexWord speech recognition.

WholeWord speech recognition and FlexWord speech recognition can be added on a per-channel basis.

### WholeWord Speech Recognition

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

WholeWord speech recognition is currently available only for US English and UK English.

### Standard Versus Custom Vocabulary

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

- The digits 0 to 9 and “oh”
- The words “yes” and “no”

The standard WholeWord recognition vocabulary also provides connected-digit recognition (the use of two or more digits in a sequence, such as an account number).

The standard speech vocabulary is best suited for applications that require callers to respond with “yes” or “no” (such as a survey application) or for number-intensive applications (such as a banking application that requires callers to enter account numbers).

You can purchase a custom vocabulary if the standard WholeWord speech recognition vocabulary does not meet your needs.

### Requirements

WholeWord speech recognition requires:

- The WholeWord speech recognition base package and one or two language package(s)

**Note:** For use with the LSPS II circuit card, WholeWord speech recognition (**asrenna**) is included with the LSPSset software package.

- WholeWord functionality assigned to either SSP *or* LSPS II circuit cards

**Note:** If your system is equipped with both SSP and LSPS II circuit cards, be careful to assign the functionality to either the SSP *or* the LSPS II circuit cards, not to both.

### References

For more information about WholeWord speech recognition and related topics, see the following:

- To install the WholeWord feature software for use with the SSP circuit card, see “Installing the WholeWord Recognition Packages” in “Installing Optional Feature System Software” in *UCS 1000 R4.6 Maintenance*, 585-313-154.

- To install the WholeWord feature software (part of **asrenna** in the LSPSset) for use with the LSPS II circuit card, see Chapter 9, "Installing the LSPS II Optional Software Packages," in *UCS 1000 R4.6 Maintenance*, 585-313-154.
- For information on:
  - ~ SSP circuit card functions, see [The SSP Circuit Card on page 19](#)
  - ~ LSPS II circuit card functions, see [Commands for SSP Circuit Card Only on page 21](#)

## FlexWord Speech Recognition

FlexWord speech recognition recognizes phonemes or parts of words. It is available for use in the US English language.

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

### Requirements

FlexWord speech recognition requires:

- FlexWord speech recognition base package and one language package

- SSP or LSPS II circuit card with FlexWord functionality assigned

### References

For more information about FlexWord speech recognition and related topics, see the following:

- To install the FlexWord feature software, see “Installing the FlexWord Speech Recognition Package” in “Installing Optional Feature System Software” in *UCS 1000 R4.6 Maintenance*, 585-313-154.
- For information on:
  - ~ SSP circuit card functions, see [The SSP Circuit Card on page 19](#)
  - ~ LSPS II circuit card functions, see [Commands for SSP Circuit Card Only on page 21](#)

### Dial Pulse Recognition

Dial pulse recognition (DPR) allows users with rotary telephones or push-button telephones that generate dial pulses to respond to the system. DPR converts the “pops” and “clicks” on the line to dial pulses. DPR supports digits 0-9 on analog and digital interfaces.

**Note:** DPR is available only with the SSP circuit card. For more information on DPR, see [Chapter 5, Recognizing Dial Pulse Input](#).

### Requirements

DPR requires:

- Dial Pulse Recognition software package
- An SSP circuit card with DPR functionality assigned

### References

For more information about DPR and related topics, see the following:

- To install the DPR feature software, see “Installing the Dial Pulse Recognition Package” in “Installing the Optional Feature Software” in *UCS 1000 R4.6 Maintenance*, 585-313-154.
- For information on SSP circuit card functions, see [The SSP Circuit Card on page 19](#).

## The SSP Circuit Card

### Description

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

### Capacity

[Table 3 on page 20](#) lists the SSP speech channel capacities.

Channel counts assume that the entire SSP circuit card is dedicated to the specified feature. For SSP circuit card channel capacities for features other

than those listed here, see “Optional System Hardware” in Chapter 2, “Hardware,” in *UCS 1000 R4.6 System Description*, 585-313-222.

**Table 3. SSP Circuit Card Channel Capacities**

Feature	Simultaneous Transactions Supported
DPR	60
Echo cancellation	60
FAX	90
FlexWord speech recognition, without barge-in *	15
FlexWord speech recognition, with play and barge-in *	12
TTS	60
Playback/record in 16-Kbps CELP	120/60
Playback/record in 64-Kbps PCM (Mu-law)	120/120
<i>1 of 2</i>	

Table 3. SSP Circuit Card Channel Capacities

Feature	Simultaneous Transactions Supported
Playback/record in 16- or 32-Kbps ADPCM	120/120
WholeWord speech recognition, without barge-in*	15
WholeWord speech recognition, with play and barge-in*	12
<i>2 of 2</i>	

\* Echo cancellation is required for barge-in

### References

- To install an SSP circuit card, see “Installing or Replacing Circuit Cards” in *UCS 1000 R4.6 Maintenance*, 585-313-154.
- To assign SSP functionality to an SSP circuit card, see Chapter 3, “Voice System Administration” in *UCS 1000 R4.6 Administration*, 585-313-509.

### Commands for SSP Circuit Card Only

The commands listed here are designed only for speech data used with the SSP circuit card; although some of the commands may work with the LSPS II circuit card, the commands are intended for use only with the SSP circuit

card: **add**, **addhdr**, **codetype**, **list**, **copy**, **decode**, **encode**, **erase**, **spres**, **spsav**, **striphdr**, and **vusage**.

## LSPS II Circuit Card

### Description

The Lucent Speech Processing Solutions (LSPS II) circuit card is a high-performance signal processor circuit card capable of simultaneous support for various speech technologies. A single LSPS II circuit card can run Text-to-Speech, WholeWord speech recognition, FlexWord speech recognition, echo cancellation, and voice code and play back.

**Note:** On the LSPS II circuit card, echo cancellation and coding cannot both be turned on at the same time. If echo cancellation is on while coding, only silence is recorded.

### Capacity

[Table 4 on page 23](#) lists the LSPS II speech channel capacities.

Channel counts assume that the entire LSPS II circuit card is dedicated to the specified feature. For LSPS II circuit card channel capacities for features other than those listed here, see “Speech and Signal Processor Card” in Chapter 2, “Hardware,” in *UCS 1000 R4.6 System Description*, 585-313-222.

Table 4. LSPS II Circuit Card Channel Capacities

Feature	Simultaneous Transactions Supported
Echo cancellation	64
FlexWord speech recognition	24
FlexWord speech recognition with Text-to-Speech	14
TTS	64
TTS with FlexWord and/or WholeWord	20
WholeWord speech recognition	30
WholeWord speech recognition with Text-to-Speech	18
Playback/record in 64-Kbps PCM (Mu-law)	64/64
Playback/record in 32-Kbps OKI ADPCM	64/64

### References

- To install an LSPS II circuit card, see “Installing or Replacing Circuit Cards” in *UCS 1000 R4.6 Maintenance*, 585-313-154.
- To assign LSPS II functionality to an LSPS II circuit card, see Chapter 3, “Voice System Administration” in *UCS 1000 R4.6 Administration*, 585-313-509.

# 2 Developing Speech

## Overview

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 any of the following elements:

- A complete sentence
- A single word
- A silence period of specified duration
- Music
- A tone (for example, a “beep”)

Speech phrases are typically specific to a single application. You determine the speech phrase content based on the application requirements. However, some speech phrases, such as generic greetings or prompts, may be used in multiple applications.

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

This chapter provides background information necessary to complete the process of creating speech on the system. Topics covered include:

- [Determining the Transaction on page 27](#)
- [Planning the Voice Script on page 28](#)
- [Writing the Voice Script on page 30](#)
- [Selecting a Speech Development Method on page 34](#)
- [Recording Speech on page 39](#)
- [Encoding Speech Phrases on page 43](#)
- [Installing Speech on page 44](#)

## Determining the Transaction

### Background

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

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

### References

For information on:

- Planning a voice response application, see *Application Design Guidelines*, 585-310-670 in the language translation of your choice.
- Developing an application using Intuity Response Application Programming Interface (IRAPI), the Transaction State Machine (TSM) process script level language, or C language, see *UCS 1000 R4.6 Application Development with Advanced Methods*, 585-313-225.
- LSPS II circuit card applications, print the LSPS Software Development Kit resident on the system by moving the file **/usr/lsp/doc/pdf/sdk.pdf** from the system to a location that can be accessed by acrobat reader.

## Planning the Voice Script

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

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

- Use the following types of phrases in your voice script:
  - ~ Long phrases that stand alone, for example,  
“Welcome to the LINC Server 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 you will use repeatedly.
- Anticipate the environment in which the phrases will be used — that is, whether the phrase will be used at the beginning, in the middle, or at the end of a sentence. The following example shows each use of the word “enter.”
  - “*Enter* the pound key.” (phrase at the beginning of the sentence)
  - “You need to *enter* the pound key.” (phrase in the middle of the sentence)
  - “Please press *enter*.” (phrase at the end of the sentence)

If your application contains a phrase that is used in more than one environment, you must record the phrase separately for each one. In the above example, you would need three recordings of the phrase “enter” —

one phrase with rising inflection for use at the beginning of a sentence, one with medial inflection for use in the middle of a sentence, and one with falling inflection for use at the end of a sentence. Recording words with the proper emphasis is discussed below in [Writing the Voice Script on page 30](#).

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

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

The following is a better example of how to word this instruction:

“In the schedule book for course offerings, find the specific course you are interested in. To the left of the title is a 5-digit number used for class scheduling. Press the keys or say that number now.”

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

## Writing the Voice Script

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

**Using Framing in Voice Scripts**

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

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

“Enter” the pound sign.

Please press “enter.”

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

“Welcome to our telephone information service.”

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

“This amount represents” the total balance.

“Please enter” two oh one.

You have “a balance of” two hundred dollars.

You can deposit “up to” five hundred dollars.

### Analyzing Speech Inflections

Three types of inflection exist with speech phrases:

- Rising inflection

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

- Medial inflection

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

- Falling inflection

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

### Placing Frame Words

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

Installing “and” verifying  
Cutting “and” pasting

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

### Selecting Speech Sounds for Framing

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

December “eighth”

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

August “eighth”

### Including Voiceless Speech Sounds

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

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

## Selecting a Speech Development Method

As an application developer, you have several options from which to choose for including speech in your application.

Options that require you to record speech are:

- [Hiring a Professional Speaker on page 35](#)
- [Purchasing a Lucent Technologies Custom Speech Package on page 36](#)

Options that do not require you to record speech are:

- [Using TTS \(Text-to-Speech\) on page 37](#)

**Hiring a Professional Speaker**

Hiring a professional speaker, such as an actor or an announcer, gives you recorded speech of high quality. Consider the following 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 on page 28](#) and [Writing the Voice Script on page 30](#) for guidelines.
- Audition several speakers of both sexes. Record and digitize their voices to evaluate the encoded quality.
- Ensure that the speaker is able to maintain the following:
  - ~ Constant speaking rhythm and general intonation throughout the recording session (this ensures that phrases spoken early in the session result in normal-sounding speech when they are concatenated with phrases spoken later in the session)
  - ~ Constant acceptable level of volume
  - ~ Clear pronunciation
  - ~ Constant orientation and distance from the microphone
- Ensure that alphabetic and numeric characters that are to be recorded with rising, medial, and falling inflections are spoken with appropriate inflection.
- Use the same speaker for all speech associated with a specific application.

If you hire a professional speaker, you can edit the speech phrases for the application script.

See [Recording Speech on page 39](#) below for information on setting up a recording session with a professional speaker.

### 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 EBS (Enhanced Basic Speech) phrases. Therefore, a continuity can exist among scripts that use both custom and enhanced basic speech.

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

**Using TTS  
(Text-to-Speech)**

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

**Note:** Standard TTS is available for US English only. However, TTS is available for other languages using the Proxy Text-to-Speech (PTTS) feature package, which allows the TTS to be processed by an external (proxy) server. For more details, see the *Proxy Text-to-Speech (PTTS) User Guide*, 585-350-115.

TTS converts text to speech in the following manner:

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

TTS constructs speech by concatenating units of speech. When constructing speech from these units, the TTS feature adjusts parameters, such as pitch

and duration, to make the outcome sound natural. Text filtering is critical because it:

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

TTS functionality is supported through TSM script instruction. The talkoff function and other system features for voice response work with TTS as they work with other speech files.

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

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

For information on using TTS in a TAS application, see Appendix B, “Summary of TAS Script Instructions,” in *UCS 1000 R4.6 Application Development with Advanced Methods*, 585-313-225.

## Recording Speech

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

- [Working with a Professional Speaker on page 39](#)
- [Working with Lucent Technologies to Develop a Custom Speech Package on page 42](#)

### Working with a Professional Speaker

The following sections provide information on the considerations involved in planning a recording session with a professional speaker and recording the speech on magnetic tape:

- [Environmental Conditions on page 40](#)
- [Equipment Specifications on page 40](#)
- [Recording Natural Sounding Speech on page 41](#)
- [Recording Sets of Related Words on page 41](#)
- [Recording Alphabetic and Numeric Characters with Inflections on page 42](#)

### 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 carpeted floors.

### Equipment Specifications

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

The recording apparatus and medium should provide the following:

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

### Recording Natural Sounding Speech

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

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

When this sentence is encoded, the phrase “The current temperature is” can be encoded as one phrase and “degrees Fahrenheit” can be encoded as a second phrase. The speech phrase “sixty-seven” should be removed because “sixty-seven” is a combination of two phrases that are recorded separately and concatenated later (numbers and alphabetic characters are recorded as separate phrases). See [Recording Alphabetic and Numeric Characters with Inflections on page 42](#) below for tips to record alphabetic and numeric characters.

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

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

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

### Recording Sets of Related Words

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

The movie for “[the name of day]” is \_\_\_\_\_.

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

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

### Recording Alphabetic and Numeric Characters with Inflections

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

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

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

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

- A clear indication of the phrase tags that corresponds to each phrase
- Specifications of what type of speech encoding should be used

**Note:** The choice of formats depends in part on the amount of speech necessary for the applications on the system. Some coding rates requires more storage space than others. See [Table 1 on page 3](#) in [Chapter 1, Overview of Speech](#) for a comparison of the storage requirements for various coding rates.

- Indication of whether each phrase is to be recorded in a male or female voice

For additional information on the custom speech recording service, contact the Speech Coordinator for the UCS 1000 R4.6 at the following number: 614-860-2260

## Encoding Speech Phrases

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

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

## Installing Speech

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

### Installing Speech in an IRAPI Application

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

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

For example, if you are using the default speech filesystem, enter:

```
cd /voice1/vfs/talkfiles
```

- 3 Making sure to use a talkfile number not already in use, enter:

```
mkdir talkfile#
```

where *talkfile#* is the talkfile number.

For example, if you are using talkfile100, enter **mkdir 100**

- 4 Copy the speech phrases to the directory specified above.

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

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

With an IRAPI application, you can put speech anywhere on your system. For information on accessing speech files in an IRAPI application, see Chapter 5, “IRAPI Programming,” in *UCS 1000 R4.6 Application Development with Advanced Methods*, 585-313-225.

# 3 Recognizing WholeWord Speech Input on the SSP

## Overview

**Note:** Information in this chapter pertains to the feature working only on the SSP circuit card.

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

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

This chapter describes the capabilities of WholeWord speech recognition and the factors that influence the accuracy of recognition. This background information is necessary to use the WholeWord speech recognition package for optimal accuracy. Topics covered include:

- The process by which WholeWord speech recognition works on the SSP circuit card
- Languages supported

- Standard vocabulary
- Speech recognition types
- Barge-in
- Positive and negative influences on speech recognition accuracy

# How WholeWord Speech Recognition Works

## Overview

WholeWord speech recognition is available as an optional installable package on the UCS 1000 R4.6 system. WholeWord speech recognition allows you to write applications that prompt for and understand spoken input from callers. This feature supports a standard vocabulary that includes:

- The digits 0 to 9 and "oh"
- The words "yes" and "no"

Processing involved during WholeWord speech recognition on the UCS 1000 R4.6 system includes a caller response, the action of the recognizer, and processing.

## Caller Response

During a prompt and collect action, the system prompts the caller for a response. If the prompt and collect action allows for voice input, the system locates and reserves a free recognizer resource. If the caller responds using touch tones, resources of the SSP circuit card are not required for this

particular prompt and the resource is freed. If the caller responds using voice input, the input is sampled at 8000 samples per second, digitized, and then transferred to the SSP circuit card via the H.110 bus. The H.110 bus provides a communication link between the SSP circuit card and the telephone interface cards (E1/T1/Quad) and also allows SSP circuit card resources to be shared across all incoming voice channels. This makes available more channels of recognition, since there is no need to dedicate certain incoming channels to only WholeWord speech recognition when accepting both touch tone and voice input. Thus channels can share resources.

#### The Recognizer

The SSP circuit card is loaded with the WholeWord speech recognition software, also called the *recognizer*. The recognizer compares the incoming speech sample to the word models indicated by the recognition type chosen in the Prompt and Collect action. The recognizer computes a score for each of the models. The score represents the likelihood that the incoming speech matches a word model indicated by the recognition type. The recognizer returns the most likely candidate to the variable specified:

#### Computational Processing

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

The SSP circuit card uses only 19 watts per circuit card. For more information on the SSP channel capacities, see [Capacity on page 19](#) 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 [Recognition DIPs on page 56](#), for further information on **recog\_dip** and other recognition post-processing functions.

## WholeWord Hardware and Software

For a list of the required software and hardware, see [WholeWord Speech Recognition](#) in [Chapter 1, Overview of Speech](#).

For the procedure to install the WholeWord software, see “Installing the WholeWord Recognition Package” in “Installing Optional Feature System Software” in *UCS 1000 R4.6 Maintenance*, 585-313-154.

## Languages Supported

WholeWord speech recognition provides standard vocabulary support for US English and UK English.

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

## Bilingual Capabilities

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

## Standard Vocabulary

[Table 5](#) lists the standard vocabulary supported for the US English language in WholeWord speech recognition.

Table 5. Standard Vocabulary for WholeWord Speech Recognition

Caller Input Field	US & UK English
0	zero, oh
1	one
2	two
3	three
4	four
5	five
6	six
7	seven
8	eight
9	nine
yes	yes
no	no

## Standard WholeWord Speech Recognition Types

### Description

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

- Mathematical models of the words, see [Models and Speaker Independence on page 52](#).
- A grammar, see [Grammars on page 53](#).
- Specifications of maximum and minimum numbers of digits, see [Keyword Recognition Types on page 53](#).
- A recognition data interface process (DIP) — optional (discussed below) The standard WholeWord speech recognition package provides several commonly used recognition types. See [Recognition DIPs on page 56](#).

### Models and Speaker Independence

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

accents. The recognition type determines which models the incoming speech is compared to.

## Grammars

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

## Keyword Recognition Types

Keyword recognition types are used for isolated word recognition, that is, responses of a single word or digit. [Table 6](#) shows the standard WholeWord speech recognition types for US English, which are used when prompting the caller to speak a response of one word or digit. The US\_1\_3 recognition type, for example, means that callers are expected to say a single word, either “one,” “two,” or “three.” A prompt that might use the 1\_3 digit grammar type

could be structured as follows: “For checking account balance, say ‘one’. For savings account balance, say ‘two’. For interest rates, say ‘three’.”

**Table 6. US English Standard WholeWord Speech Recognition Types on the SSP Circuit Card**

Prompt	Recognition Type
"Yes" or "No"	US_YN
One digit 1_3	US_1_3
One digit 1_3 or "No"	US_1_3N
One digit 1_5	US_1_5
One digit 1_5 or "No"	US_1_5N
One digit 0_9 or "Oh"	US_DIG

## Connected-Digit Recognition Types

*Connected-digit recognition types* are used for connected-word recognition, that is, responses of more than one word or more than one digit string. 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- to 10-digit fixed-length and

the 1- to 24-digit variable-length strings. [Table 6 on page 54](#) lists the connected-digit recognition types.

**Note:** With US English, the recognition type is US\_DIG and use of a custom grammar is recommended.

For better accuracy, you must specify the desired string length when selecting the recognition type for a fixed-length string. A US telephone 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.

#### Connected-Digit Recognition

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

#### Fixed-Length versus Variable-Length Connected Digits

The capability to recognize fixed-length strings of 1 to 10 digits is provided with WholeWord speech recognition packages (see [Table 6 on page 54](#)). A *fixed-length string* is a string of digits that is always made up of the same number of digits. For example, a U.S. Social Security number is a fixed-length number, since it always consists of nine digits. By contrast, a *variable-*

*length string* indicates a string of digits that varies, for example, a street address.

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

Recognition accuracy is best when you use fixed-length strings. For strings longer than 10 digits, use of a custom grammar is recommended (see [Custom Grammars on page 63](#)). Using the provided variable-length grammar as an alternative results in decreased accuracy. If your application must accept a variable-length digit string, you may be able to increase recognition accuracy by using a two-step entry process. First prompt the caller for the number of digits, and then prompt for the digits. For example:

“How many digits are in the next code?”

“Please say the three-digit code now.”

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

## Recognition DIPs

A *recognition data interface process* (DIP) is used to help increase an application’s accuracy. For example, certain digit strings, such as credit card numbers, have check digits built into them. Verifying a check digit is done most efficiently through a DIP. However, a DIP is not necessary for many recognition types. If there is no further information to consider for a string

other than what is in the grammar, a DIP is not needed. Specifically, none of the standard WholeWord speech recognition types, such as “yes” and “no,” needs a DIP.

For some recognition types, there may be tradeoffs that determine how much of the structure of the input is built into the grammar and how much should be left for the DIP. For more information about custom DIPs, see “Writing the DIP” in Chapter 4, “Data Interface Processes,” in *UCS 1000 R4.6 Application Development with Advanced Methods*, 585-313-225. Also see [Custom DIPs on page 65](#).

## Word Spotting

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

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

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

## 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. The application can be designed to 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. Although the system can screen phrases, recognition accuracy is best when the application is written to encourage the caller to speak only what is required.

## Recognition Confirmation

*Recognition confirmation* is the ability of the system to repeat the caller’s spoken response back to the customer for confirmation. For example, a script prompts a caller, “Please say your area code,” and the caller says, “Six, one, four.” Using recognition confirmation, the system repeats what it recognizes the caller’s response to be and then prompts, “If this is correct, say ‘yes’, if

not say ‘no’.” The caller then confirms whether the system recognized the spoken input properly.

## Barge-In

### Description

*Barge-in*, also called “allow interrupt” or “recognize during prompt,” is the ability of the system to allow callers to interrupt or barge in during voice playback by speaking a vocabulary word. Speech recognition accepts either speech or touchtone input in response to a prompt. Barge-in operates for speech much like the talkoff option does for touchtone input, where a caller can interrupt the prompt by pressing a touchtone key on the telephone keypad. For more information on touchtone interrupts, see [Dual Tone Multifrequency \(DTMF\) Support on page 62](#).

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.

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

For isolated word recognition, the prompt does not stop until the system recognizes a valid vocabulary word. Once the prompt completes playback,

the initial timeout field eventually ends the recognition if no valid input is received.

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

#### Enabling or Disabling Barge-In

At the beginning of an application, after the answer instruction, you can place an instruction to enable barge-in using the **sr\_talkoff** (TAS instruction). If a barge-in resource is available, it is reserved for this call. Otherwise the return code is negative, and barge-in will not be available. For more information see Appendix B, "Summary of TAS Script Instructions," in *UCS 1000 R4.6 Application Development with Advanced Methods*, 585-313-225.

#### Discouraging Barge-In

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

“For sales say, ‘one’.” (short pause)

“For service, say, ‘two’.” (short pause)

“To speak with a representative, say ‘three’.”

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

“You can order up to five copies. Please say how many copies you want.”

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

#### Encouraging Barge-In

The following prompts are designed to encourage callers to barge in when barge-in is enabled. Note that the pauses are longer than in the examples in [Discouraging Barge-In on page 60](#) above, so callers are encouraged to respond after hearing the desired option and the word “now.”

“For sales say, ‘one’ now.” (pause of 1-1.5 seconds)

“For service, say, ‘two’ now.” (pause of 1-1.5 second)

“For a representative, say ‘three’ now.” (pause of 1-1.5 seconds)

#### System Response for Barge-In

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

- When a fixed-length digit recognition type is used on the SSP circuit card, the prompt can be interrupted after  $N-2$  ( $N$  minus 2) digits are spoken plus a system delay of 0.75 second, where  $N$  is the total number of digits to be collected. For example, for the US\_9DIG recognition type, the prompt can be interrupted after seven digits.
- When short prompts are used, callers might not detect delays in the barge-in point because the prompt frequently completes before the caller speaks much input.

## Dual Tone Multifrequency (DTMF) Support

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

Often the response time for talkoff is faster than for barge-in. This is because touch tones were designed to be recognized by machines and are rarely confused with other sound or voice input. However, WholeWord speech recognition must perform the more complex task of separating the desired input from other sound or voice input that may simultaneously be present. The system does not terminate a prompt until it processes enough input to ensure with relatively high confidence that the input is valid. For a prompt requiring single-digit input, talkoff and barge-in may appear equally fast since the input is completed with a single touch tone or a single spoken digit. Prompts for multidigit input reveal the difference between the response times for talkoff and barge-in. See [Barge-In on page 59](#) for information comparing and contrasting talkoff and barge-in.

## Custom Grammars

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

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

For more information, contact your Lucent Technologies representative.

## Custom Vocabulary

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

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

A custom vocabulary requires application analysis by Lucent Technologies, speech data collection, model creation, and custom grammar work. Depending on your application, you may want to consider FlexWord™ speech recognition as a more cost-effective alternative. For more information, see [Chapter 4, Recognizing FlexWord™ Speech Input on the SSP Circuit Card](#).

For more information, contact your Lucent Technologies representative.

## Custom DIPs

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

## WholeWord Speech Recognition Accuracy

### Overview

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

### Positive Influences on WholeWord Speech Recognition Accuracy

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

### **Isolated Word Recognition**

Isolated word recognition is very high. The smaller the number of choices in an isolated word recognition type, the better the accuracy. For example, “US English Digits 1 to 3” is more accurate than “US English Digits 1 to 5”, which in turn is more accurate than using “US English 1 digit (0-9 and ‘oh’)”.

### **Fixed-Length Digit String**

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

### **Validation of Data**

Try to verify the recognized result against a database or a host field. This helps improve the overall accuracy of an application, especially when a longer string is input.

### **Reprompt**

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

### Prompt Structure

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

- Menu prompts

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

*<desired result> <action required>*

Examples:

“To hear your checking account balance, say 1.”

“To hear your savings account balance, say 2.”

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

- Yes and no prompts

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

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

This wording is more natural than the following:

“To hear your order again, say ‘yes’. Otherwise, say ‘no’.”

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

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

#### **Calling Experience and Informative Prompts**

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

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

#### Custom Grammars and DIPS

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

#### Negative Influences on WholeWord Speech Recognition Accuracy

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

#### Environment

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

#### Extraneous Words Within Responses

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

### Information Type

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

### Regional and National Accents and Dialects

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

### Connected-Digit String Length

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

digit accuracies, as well as overall string accuracies, vary according to the language and noise conditions of different national networks.

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

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

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

#### Application-Related Limitations

The capability of the system and WholeWord speech recognition is application dependent. If the system is underengineered for a particular application, it may not perform satisfactorily.

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

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

#### The Recognizer

**Note:** If your system is equipped with both the LSPS II and the SSP circuit cards, assign the function to either the SSP or LSPS II circuit card, not both.

The SSP circuit card is loaded with the WholeWord speech recognition software, also called the *recognizer*. The recognizer compares the incoming speech sample to the word models indicated by the recognition type chosen in the prompt and collect action. The recognizer computes a score for each of the models. The score represents the likelihood that the incoming speech matches a word model indicated by the recognition type.

See *UCS 1000 R4.6 Application Development with Advanced Methods*, 585-313-225 for more information.

#### Computational Processing

A single SSP circuit card has the ability to run Text-to-Speech (TTS), speech recognition, dial pulse recognition (DPR), FAX, voice code and playback, and full call classification analysis (CCA) simultaneously.

The SSP circuit card uses only 19 watts per circuit card. For more information on the SSP channel capacities, see [The SSP Circuit Card on page 19 in Chapter 1, Overview of Speech](#).

# 4 Recognizing FlexWord™ Speech Input on the SSP Circuit Card

## Overview

**Note:** Information in this chapter pertains to the feature working only on the SSP circuit card.

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

- Background information about the capabilities of FlexWord speech recognition
- Guidelines for designing FlexWord wordlists for use in applications
- Information on positive and negative influences on the accuracy of FlexWord speech recognition

## Description

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

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

## How FlexWord Speech Recognition Works

FlexWord speech recognition uses subword technology. Subword technology relies on phonemic recognition for analyzing and recognizing words.

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

Either the speech and signal processor (SSP) circuit card or the Lucent Speech Processing Solutions (LSPSII) circuit card is required for all languages.

Illustrated in [Figure 6](#) are FlexWord speech recognition phonemes for the SSP circuit card in the US English language. A word accompanies each phoneme to illustrate its sound.

Figure 6. US English Phonemes for the SSP Circuit Card

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

## FlexWord Capacity

Every entry in a wordlist is counted as a distinct and separate word. The maximum number of words that can be loaded onto the FlexWord speech recognizer is 2000. (FlexWord speech recognition supports up to 200 wordlists and 500 words per wordlist, as shown in [Table 7](#).) For example, if the word “help” appears as an entry in 10 of your wordlists, “help” is counted as 10 separate words. Phrases such as “loan information” count as only one word. Each wordlist must be given a name tag consisting of 1 to 14 uppercase characters. For definitions of word, wordlist, or vocabulary terms, see the [Glossary](#) at back of this book.

**Table 7. FlexWord Maximum Capacity**

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

### Vocabularies

A FlexWord speech recognition vocabulary can include a maximum of 2000 words or phrases per system, allowing up to 200 wordlists. Each wordlist can

include up to 500 words. (For an application that requires more than 2000 words, contact your Lucent Technologies representative.)

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

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

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

### Keyword Spotting

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

### 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 says the word “banking”, the application can be designed to inform the caller that the response is invalid and to replay the prompt.

### Barge-in

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

## Designing a FlexWord Speech Recognition Application

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

### Defining Wordlists

The first step in designing a wordlist is to define the words you want the system to recognize and group these words into wordlists. As [Figure 7 on page 82](#) 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 7 on page 82](#) and [Figure 8 on page 83](#) contain groups of possible words from which callers can choose. All of the wordlists for all of your applications constitute the vocabulary.

### Using Wordlists in Scripts

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

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

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

For more information on how to create effective wordlists, see [Positive Influences on FlexWord Speech Recognition Accuracy on page 90](#).

Figure 7. FlexWord Speech Application Example Using Menu Prompts

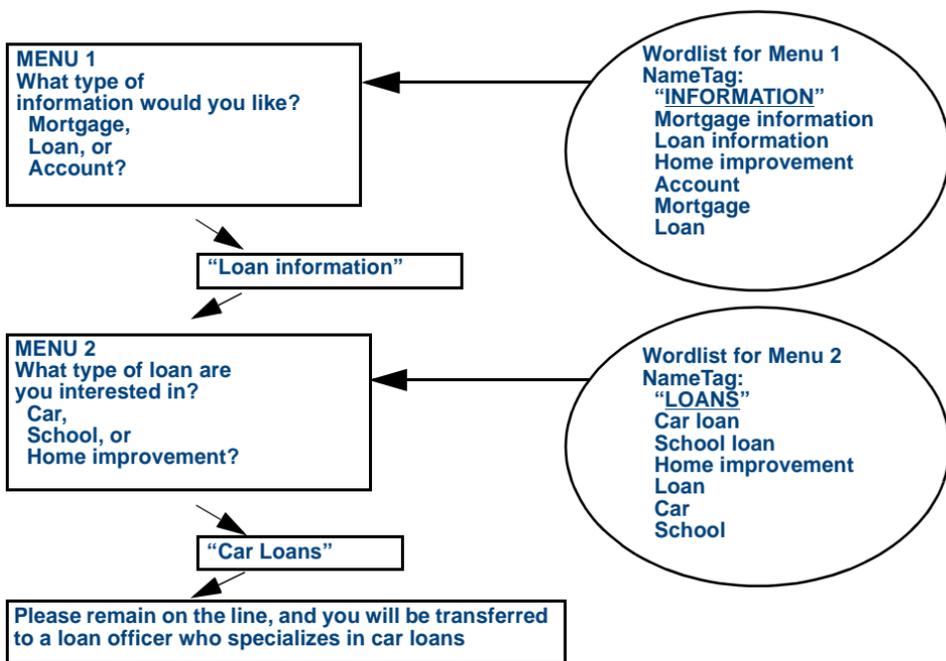
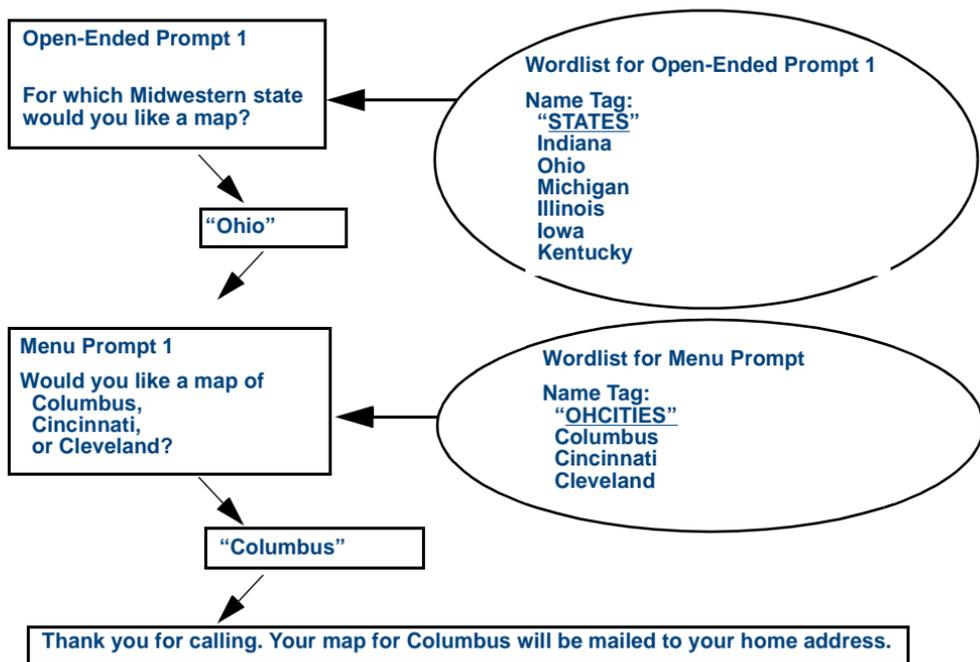


Figure 8. FlexWord Speech Application Example Using Open-Ended and Menu Prompts



# FlexWord Software and Hardware

For a list of the required software and hardware, see [FlexWord Speech Recognition on page 17](#) in [Chapter 1, Overview of Speech](#).

To install the FlexWord feature software, see “Installing the FlexWord Speech Recognition Package” in “Installing Optional Feature System Software” in *UCS 1000 R4.6 Maintenance*, 585-313-154.

## FlexWord Recognition Vocabulary Administration

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

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

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

- 1 Go to the directory where the wordlists are located.
- 2 Insert a diskette into the diskette drive.
- 3 Enter: **wl\_copy filename1 filename2...filenameN**  
where *filename1 filename2...filenameN* are the names of wordlists.

**Note:** Since they will be used to load the wordlists onto a FlexWord recognition system, filenames must be relative, not absolute pathnames.

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

### Installing and Activating Wordlists from Diskette

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

- 1 Enter: **wl\_install**
- 2 Insert the diskette containing the wordlists (made in the procedure in [Copying Wordlists on page 85](#)) into the diskette drive when prompted.

The system copies the wordlists to the **inactive** directory (**/att/asr/wordlists/inactive**) and prompts as to whether any of the wordlists should be activated and the **wl\_gen** command run.

- 3 Designate, if appropriate, the wordlists that should be activated.

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

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

### Activating Wordlists That Reside on the System

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

**Note:** Do not use this procedure if you have already activated the wordlists through the **wl\_install** command. See [Installing and Activating Wordlists from Diskette on page 86](#).

- 1 Make sure the wordlist you want to activate is currently in the **inactive** directory (**/att/asr/wordlists/inactive**)
- 2 From the **inactive** directory, enter **In *filename* /att/asr/wordlists/active** where *filename* is the name of the wordlist you want to activate.

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

- 3 Enter: **wl\_gen *language***

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

The **wl\_gen** command verifies the format of all active wordlists and creates the data files needed for FlexWord speech recognition. If the command finds more than 38 phonemes, including the underscore (**\_**), within any entry in a wordlist, you receive an error message. Once **wl\_gen** finds an error in a wordlist, it quits looking at that wordlist, and the wordlist containing the error is not used when generating the FlexWord

recognition data files. Therefore, if you receive an error message, you may have to run **wl\_gen** several times to locate each error.

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

See *UCS 1000 R4.6 Administration*, 585-313-509, for more information about the **diagnose card** command.

- 5 Assign the FlexWord function to the SSP circuit card.

For the procedure, see Chapter 3, “Voice System Administration” in *UCS 1000 R4.6 Administration*, 585-313-509.

### Deactivating Wordlists

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

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

For more information, see Chapter 3, “Voice System Administration” in *UCS 1000 R4.6 Administration*, 585-313-509.

- 2 Make sure you are in the **/att/asr/wordlists/active** directory.
- 3 Enter: **rm filename**

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

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

### 4 Enter: **wl\_gen language**

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

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

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

#### 1 Deactivate the wordlist.

See [Deactivating Wordlists on page 88](#) for the procedure.

#### 2 Go to the **/att/asr/wordlists/inactive** directory.

#### 3 Enter **rm filename** where *filename* is the name of the wordlist to be deleted.

## FlexWord Speech Recognition Accuracy

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

### Positive Influences on FlexWord Speech Recognition Accuracy

Factors that positively affect accuracy rates include:

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

See also [Chapter 6. Putting It Together with the SSP Circuit Card](#) for information about getting the most out of a FlexWord speech recognition application.

### Effective Wordlists

Choose words for your wordlist that have different sounds and are of medium length to help increase recognition accuracy. For example, using the last name and the first name on wordlists of a name dialer application (that is, an

application that allows employees to speak a colleague's name rather than dial an extension number) increases recognition accuracy. In other types of applications, syllabic and vowel similarities may impair recognition. For example, "women's wear" and "men's clothing" are more effective wordlist entries than "women's clothing" and "men's clothing." The latter pair sound too much alike. Both have "men" embedded in the word and both share the word "clothing."

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

### Including Out-of-Vocabulary Terms

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

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

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

### Experienced Calling Population

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

### Prompt with Structure and Good Design

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

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

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

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

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

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

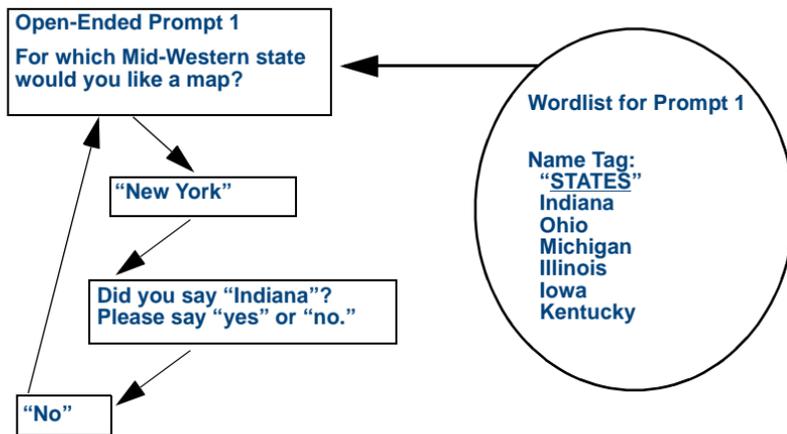
### Confirmation and Reprompt Steps

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

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

With this prompt structure, callers can make sure their words match the recognized response. [Figure 9](#) provides an illustration of a confirmation reprompt.

Figure 9. Example of a Reprompt in a FlexWord Application



### Informative Prompts

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

**Negative Influences on FlexWord Speech Recognition Accuracy**

Low recognition accuracy rates are usually due to:

- Inexperienced callers
- Environmental factors
- Ineffective wordlists

**Inexperienced Callers**

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

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

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

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

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

### Environment

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

### Ineffective wordlists

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

- **Wordlist size** — As the size of wordlists increase, accuracy decreases. The best accuracy results can be achieved by structuring an application to make use of several smaller wordlists rather than one large wordlist.
- **Rhyming words** — Wordlists that include words with similar rhyming vowel sounds can cause a decrease in recognition accuracy.
- **Short words** — One-syllable words that have the same vowel sounds are difficult to recognize. For example, “on” and “off” both share the short “o” sound. (Long “o” vowel sounds appear in words like “oh” and “no”— words in which you can hear the letter “o.”) Short words should be used in moderation.

# 5 Recognizing Dial Pulse Input

## Overview

**Note:** The information and procedures in this chapter pertain only to the SSP circuit card.

Dial Pulse Recognition (DPR) allows a user with a rotary or a push-button telephone that generates only dial pulses to respond to the UCS 1000 R4.6 system. DPR supports digits 0 (zero) through 9 on analog and digital interfaces.

Topics covered include:

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

- Troubleshooting guidelines

## Hardware

DPR is available on the speech and signal processor (SSP) circuit card. No other special hardware is needed to use it.

## Simultaneous Dial Pulse Capabilities

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

## Dial Pulse Recognition Types

Certain digit strings may require custom recognition types as a technique used to obtain acceptable recognition accuracy. Examples include a credit card number, a merchant number, and a 10-digit telephone number. These digit strings have specific limitations on the position of certain digits within the

string (for example, a credit card whose first number must be 3 or 5, and whose second number must be 1 or 7).

[Table 8](#) shows the standard dial pulse recognition types.

**Table 8. Standard Dial Pulse Recognition Types**

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

## Grammars

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

Grammars for DPR are modeled after grammars for speech recognition. DPR currently uses grammars only to do input restriction such as “ignore digit one” and pulse-to-digit mapping.

## Recognition Confirmation

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

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

## Barge-In

*Barge-in*, also known as "Allow Interrupt", is the ability of the system to allow callers to interrupt during prompts.

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

## How Dial Pulse Recognition Works

Dial-pulse digits that a user enters are collected by the algorithm on the SSP circuit card and processed according to the selected recognition type. A maximum of 60 channels are available per SSP circuit card.

For DPR to run on the SSP circuit card, it must be assigned through the `cvis_menu` screens. For information on assigning SSP functionality, see Chapter 3, See "Assign SSP Functions" in Chapter 3, "Voice System Administration" in *UCS 1000 R4.6 Administration*, 585-313-509.

### The Recognizer

The SSP circuit card is loaded with the DPR software package, also called the *recognizer*.

For a prompt of "2, 5, 8," the application developer designates the DP258 recognition type, which is used for a response consisting of "2," "5," or "8." The caller's response to the prompt is compared to the numbers 2, 5, or 8. The number of pulses that most closely matches the input determines the

recognized result. A value of 2, 5, or 8 is returned in \$CI\_VALUE. The application decides what to do next, based on the value of \$CI\_VALUE.

## Recognition Accuracy

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

### Training

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

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

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

### Limitations

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

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

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

## DPR Troubleshooting

Use [Table 9](#) to troubleshoot problems with DPR.

**Table 9. DPR Troubleshooting**

Problem	Action
DPR does not appear to be working.	A path is provided in the <b>feature_tst</b> script delivered with the system. Load this package by selecting DPR as a test path. Make sure DPR is assigned to the appropriate circuit card and is in service. The test asks you to dial any four dials pulse digits and speaks the result back to you.
DPR does not recognize digits.	A number of problems could cause DPR not to work. <ul style="list-style-type: none"><li data-bbox="314 542 1108 573">• Ensure all administration and assignments for DPR are correct.</li><li data-bbox="314 591 1108 653">• Use the <b>feature_tst</b> script to determine if it is an application or system issue.</li><li data-bbox="314 671 1108 733">• Check the system error logs and take appropriate corrective action.</li></ul> If these actions do not reveal the problem, contact your Support Hotline 1-303-538-8748.

1 of 2

Table 9. DPR Troubleshooting

Problem	Action
DPR mis-recognizes digits.	If digits are consistently mis-recognized from multiple callers, some local tuning may be necessary. Contact your Support Hotline 1-303-538-8748.

2 of 2

### Common Failure Modes

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

#### Noise on the Line

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

#### Missing or Extra Pulse

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

There are standard recognition types for single-digit prompts requiring four or fewer choices that provide good recognition accuracy. This is accomplished

by separating the valid choices by three pulses and mapping all invalid choices to their closest neighbor.

# 6 Putting It Together with the SSP Circuit Card

## Overview

**Note:** Information in this chapter pertains to the features that work only on the SSP circuit card.

This chapter describes the following considerations for using speech-related features with optimal effectiveness in applications:

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

Recognition types, including FlexWord™ speech recognition, WholeWord speech recognition, and dial pulse recognition (DPR—on SSP circuit cards only) are discussed, as well as the TTS (Text-to-Speech) feature.

**Note:** In order for barge-in to work, recognition and echo cancellation must be assigned on the same card. Otherwise a non-barge-in prompt is displayed.

## Using WholeWord Recognition and Dial Pulse Recognition Together

WholeWord speech recognition can be used to recognize a limited set of words. However, WholeWord speech recognition is most successful when you use it to augment a touchtone application to handle callers who are not using touchtone telephones.

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

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

## Getting the Most Out of FlexWord Speech Recognition

FlexWord speech recognition recognizes the caller speaking words from a vocabulary that you define. It is, therefore, specifically tailored to your application. Allowing callers to speak the options they want rather than speaking a number assigned to the option can make the interaction more natural and easy to use. Included below are guidelines for getting the most out of FlexWord speech recognition.

### Choosing a FlexWord Speech Recognition Vocabulary

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

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

### Caller Error

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

- Word Length** Words with more than one syllable are recognized more reliably than shorter words. For example, the phrase “add entry” would be better to use than the word “add.” Phrases can be fairly long, although the longer the phrase, the greater the chance of callers forgetting it or making mistakes while speaking.
- Word Choice** Choose words and phrases that would occur naturally to your typical caller. One way to determine this is to survey some representative callers and ask what words or phrases they would use for actions in your script. Do not rely on just your knowledge, because your impressions may be different than those of your callers.
- Prompting for Input** Because FlexWord speech recognition does not support barge-in, structure your prompts to encourage callers to wait until the end of the prompt before speaking. For example, use a prompt such as, “Please say the department name, now,” as opposed to “Now say the department name, please.” This is good practice for all applications, whether or not barge-in is supported.
- Word Sounds** Use words in your wordlists that do not have a similar sound. To the recognizer, vowels are more important than consonants. Avoid using words with similar vowel sounds. Words that differ in only one or two consonants are difficult for the recognizer to distinguish between. For example, the words “connect” and “comment” sound similar to the recognizer.
- Synonyms** If you find that callers often use more than one word for the same thing, you may want to add both words to your list.

### Examples of FlexWord Speech Recognition Applications

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

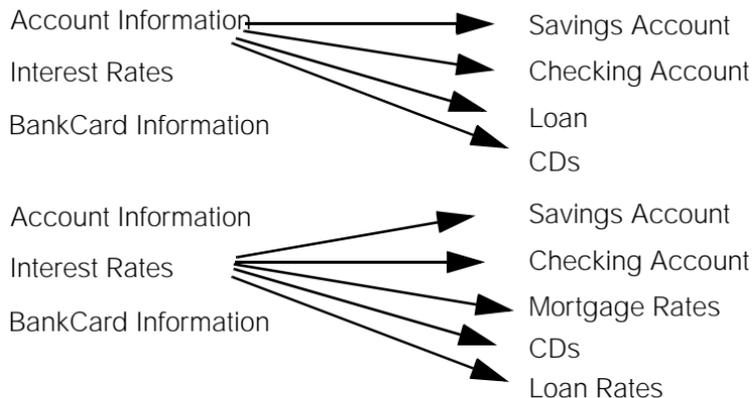
#### Applications Requiring Large Wordlists

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

#### Menu-Based Applications

Menu-based applications that use words and phrases native to your business environment are suggested for use with FlexWord speech recognition. Your application flows more smoothly if it uses the same words that would occur naturally in an interaction between an agent and the customer. [Figure 10 on page 113](#) shows an example of a menu-based application.

**Figure 10. Example of Menu-Based Application**



## Getting the Most out of Dial Pulse Recognition

As discussed in [Chapter 5, Recognizing Dial Pulse Input](#), the DPR feature recognizes caller pulse inputs from a rotary telephone or from a push button telephone that generates dial pulses. The following tips can help you get the most out of the DPR feature in a Script Builder application.

- Design your application so that the first digit is 5 or greater to allow the system to train to the caller's telephone.
- Use the standard recognition types/grammars available.
- 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, FlexWord, or Natural Language speech recognition.
- Use confirmation to improve script performance.
- When planning an application, if possible, avoid using the digits 1 or 2.
- For simple menu items, use recognition types DP258 or DPR1470 for best performance.

## Getting the Most out of Text-to-Speech

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

- When TTS prompts are used with touchtone 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.
- When using TTS prompts with FlexWord speech recognition input, barge-in is not available. Therefore, when accepting FlexWord speech recognition input, structure your TTS prompts so that they encourage callers to wait until the end before responding. Note that this is good practice for all applications, not just those for which barge-in is not supported.

**Note:** Standard TTS is available for US English only. However, TTS is available for other languages using the Proxy Text-to-Speech (PTTS) feature package, which allows the TTS to be processed by an external (proxy) server. For more details, see *Proxy Text-to-Speech (PTTS) User Guide*, 585-350-115.

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

Since FlexWord speech recognition provides the ability to recognize thousands of 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 `SCI_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:** Standard TTS is available for US English only. However, TTS is available for other languages using the Proxy Text-to-Speech (PTTS) feature package, which allows the TTS to be processed by an external (proxy) server. For more details, see *Proxy Text-to-Speech (PTTS) User Guide*, 585-350-115.

## Using Text-to-Speech and Prerecorded Speech Together

A UCS 1000 R4.6 system application can speak prompts and announcements in prerecorded speech only, TTS only (including PTTS), or a combination of prerecorded speech and TTS.

For example, in one application, recorded speech is used for all phrases except the customer's name. TTS allows the system to speak the contents of a "cust\_name" variable, which contains the customer's name as a character string. To have this name spoken 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.

Furthermore, each time you add a customer's name to the list, you would have to update your application, record the new name, recompile your application, transfer, and re-install it on the UCS 1000 R4.6 system. Obviously, this would involve a huge amount of work just to maintain the names in the system. In cases like this, it might make more sense to use TTS to speak the customer's name.

## Using WholeWord and FlexWord Speech Recognition Together

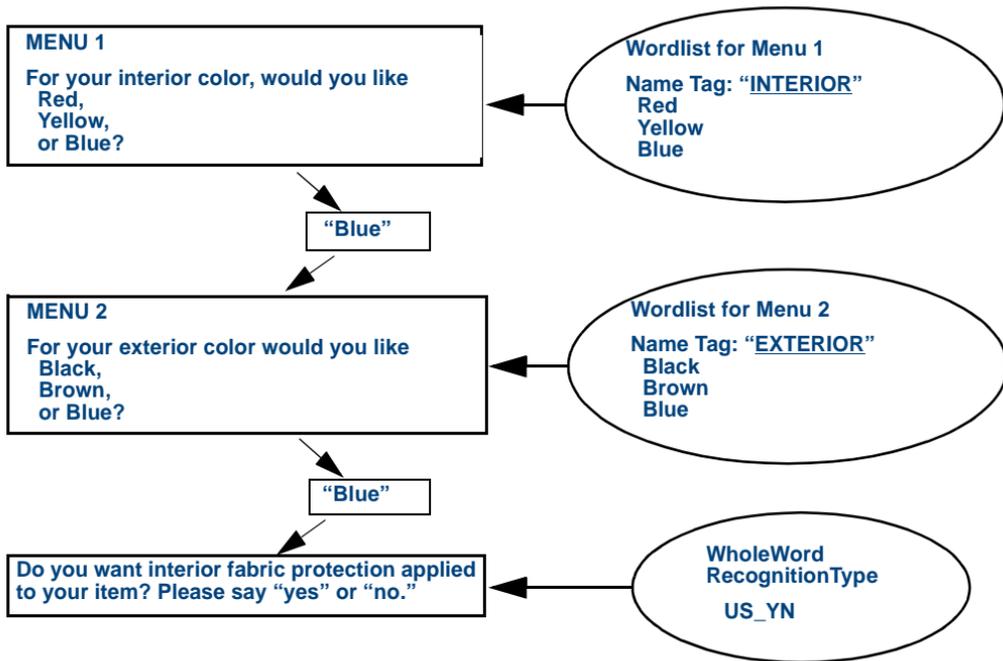
WholeWord and FlexWord speech recognition are used in either a TAS or irAPI script. See *UCS 1000 R4.6 Application Development with Advanced Methods*, 585-313-225 for TAS and irAPI information.

- If you want callers to say “yes” or “no,” a series of digits, or a single digit, choose a WholeWord speech recognition.
- If you want callers to speak a word or phrase from your custom vocabulary, choose a FlexWord speech recognition.

**Note:** Both recognizers cannot be used in the same prompt.

The example in [Figure 11 on page 119](#) shows a model of a retail application that combines FlexWord wordlists and the WholeWord “yes and no” speech recognition feature. In the example, the word “blue” must appear on both the “INTERIOR” wordlist and the “EXTERIOR” wordlist. Since the word “blue” appears on two wordlists, it counts as two words. If the word “blue” is not on the “INTERIOR” wordlist, this feature will not look to the “EXTERIOR” wordlist to find it. The feature returns the word in the wordlist that it thinks is the most suitable match. Since the feature can only look at one wordlist prompt, each wordlist must contain *all* possible choices for a single prompt.

Figure 11. Illustration of the Use of FlexWord and WholeWord Speech Recognition in a Single Application Script



## **WholeWord Speech Recognition Versus FlexWord Speech Recognition Accuracy**

For most applications, the accuracy of Whole Word recognition is higher than for FlexWord recognition. The following list shows the accuracy rates, arranged in order from highest to lowest:

- 1** WholeWord speech recognition isolated digit
- 2** WholeWord speech recognition connected digit
- 3** FlexWord speech recognition isolated word

## Comparison of Recognition Types

[Table 10](#) summarizes the similarities and differences between WholeWord speech recognition, FlexWord speech recognition, and DPR when assigned to an SSP circuit card.

**Table 10. Comparison of Speech Recognition Types on an SSP Circuit Card**

<b>WholeWord Recognition</b>	<b>FlexWord Recognition</b>	<b>Dial Pulse Recognition</b>
Word-based	Phoneme-based	Pulse-based
Requires data collection in model building for vocabulary words other than digits “zero” through “nine,” common synonyms for those words, and the words “yes” and “no”	No data collection required in model building	No data collection required in model building
Connected digits	Single word or phrase	Connected digits
Standard and custom grammars	Customized wordlists	Standard grammars

1 of 2

Table 10. Comparison of Speech Recognition Types on an SSP Circuit Card

<b>WholeWord Recognition</b>	<b>FlexWord Recognition</b>	<b>Dial Pulse Recognition</b>
Barge-in supported	Barge-in not supported	Barge-in not supported
Phrase screening supported	Phrase screening supported	Phrase screening supported
Limited vocabulary	2000-word vocabulary	Limited vocabulary
Word spotting supported	Word spotting supported	Word spotting not supported

2 of 2

# 7 Recognizing WholeWord Speech Input on the LSPS II

## Overview

**Note:** Information in this chapter pertains to the feature working only on the LSPS II circuit card.

WholeWord speech recognition is part of the *LSPSset* installable software package. See Chapter 9, "Installing LSPS II Optional Software Packages," in *UCS 1000 R4.6 Maintenance*, 585-313-154 for procedures to install the *LSPSset* software package.

WholeWord speech recognition allows you to write applications that prompt for and understand spoken input from callers. This feature supports a standard vocabulary that includes:

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

This chapter describes the capabilities of WholeWord speech recognition and the factors that influence the accuracy of recognition. This background information is necessary to use the WholeWord speech recognition package for optimal accuracy. Topics covered include:

- How WholeWord speech recognition works on the LSPS II circuit card

- Languages supported
- Standard vocabulary
- Speech recognition types
- Barge-in
- Positive and negative influences on speech recognition accuracy

## How WholeWord Speech Recognition Works

### Overview

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

### Caller Response

During a prompt and collect action, the system prompts the caller for a response. If the prompt and collect action allows for voice input, the system locates and reserves a free recognizer resource. If the caller responds using touch tones, resources of the LSPS II circuit card are not required for this particular prompt and the resource is freed. If the caller responds using voice input, the input is sampled at 8000 samples per second, digitized, and then transferred to the LSPS II circuit card via the H.110 bus.

The H.110 bus provides a communication link between the LSPS II circuit card and the telephone interface cards (E1/T1/Quad) and also allows LSPS II circuit card resources to be shared across all incoming voice channels. This makes available more channels of recognition, since there is no need to

dedicate certain incoming channels to only WholeWord speech recognition when accepting both touch tone and voice input. Thus channels can share resources.

### The Recognizer

**Note:** If your system is equipped with both the LSPS II and the SSP circuit cards, assign the function to either the SSP or LSPS II circuit card, not both.

The LSPS II circuit card is loaded with the WholeWord speech recognition software, also called the *recognizer*. The recognizer compares the incoming speech sample to the word models indicated by the recognition type chosen in the Prompt and Collect action. The recognizer computes a score for each of the models. The score represents the likelihood that the incoming speech matches a word model indicated by the recognition type.

See *UCS 1000 R4.6 Application Development with Advanced Methods*, 585-313-225 for more information.

**Computational Processing**

A single LSPS II circuit card has the ability to run Text-to-Speech (TTS), speech recognition, voice code and playback, and echo cancellation, simultaneously.

The LSPS II circuit card uses approximately 25 watts per circuit card. For more information on the LSPS II channel capacities, see [Commands for SSP Circuit Card Only on page 21](#) in [Chapter 1, Overview of Speech](#).

## WholeWord Hardware and Software

For a list of the required software and hardware, see [WholeWord Speech Recognition](#) in [Chapter 1, Overview of Speech](#).

For the procedure to install the LSPSset software package, see Chapter 9, "Installing LSPS II Optional Software Packages" in *UCS 1000 R4.6 Maintenance*, 585-313-154.

## Languages Supported

WholeWord speech recognition provides standard vocabulary support for the US English language.

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

## Standard Vocabulary

[Table 11](#) lists the standard vocabulary supported for the US English language in WholeWord speech recognition.

**Table 11. Standard Vocabulary for WholeWord Speech Recognition**

Caller Input Field	US English
0	zero, oh
1	one
2	two
3	three
4	four
5	five
6	six
7	seven
8	eight
9	nine

1 of 2

Table 11. Standard Vocabulary for WholeWord Speech Recognition

Caller Input Field	US English
yes	yes
no	no

2 of 2

## Standard WholeWord Speech Recognition Types

### Description

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

- Mathematical models of the words, see [Models and Speaker Independence on page 129](#).
- A grammar, see [Grammars on page 129](#).
- Specifications of maximum and minimum numbers of digits, see [Keyword Recognition Types on page 130](#)
- A recognition data interface process (DIP) — optional (discussed below). The standard WholeWord speech recognition package provides several commonly used recognition types. See [Recognition DIPs on page 133](#)

### Models and Speaker Independence

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

### Grammars

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

## Keyword Recognition Types

Keyword recognition types are used for isolated word recognition, that is, responses of a single word or digit. [Table 12](#) shows the standard WholeWord speech recognition types for US English, which are used when prompting the caller to speak a response of one word or digit. The CD\_3 recognition type, for example, means that callers are expected to say three successive digits. A prompt that might use the CD\_3 grammar type could be structured as follows: “Please say your three-digit telephone area code.”

**Table 12. US English Standard WholeWord Speech Recognition Types on the LSPS II Circuit Card**

Prompt	Recognition Type
"Yes" or "No"	IW_YESNO
Connected digits: telephone numbers	CD_PHONE; CD_PHONE+
Connected digits: 14-digit authorization code	CD_AUTH14
Natural numbers: dollar amounts	NN_DOLLAR
Natural numbers: dates	NN_DATE
Natural numbers: telephone numbers	NN_PHONE+

1 of 2

Table 12. US English Standard WholeWord Speech Recognition Types on the LSPS II Circuit Card

Prompt	Recognition Type
Natural numbers: time	NN_TIME
Connected digits: known length	CD_1 through CD_16
Connected digits: unknown length	CD_UNKNOWN

2 of 2

## Connected-Digit Recognition Types

*Connected-digit recognition types* are used for connected-word recognition, that is, responses of more than one word or more than one digit string. The spoken string can be fixed in length from 1 to 16 digits, or it can be of a variable length. Grammars are provided for the 1- to 16-digit fixed-length while CD\_UNKNOWN provides the variable-length strings. [Table 12 on page 130](#) lists the connected-digit recognition types.

For better accuracy, you must specify the desired string length when selecting the recognition type for a fixed-length string. A US 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.

**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 (1–16 digits) or, for US English only, a variable-length (1–24 digits). Number entry is an essential component if you want callers to input numbers with more than one digit, (for example, if using code CD\_1, the system looks for numbers 0 through 9; if using CD-3 it looks for numbers 000 through 999). 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 1–16 digits is provided with WholeWord speech recognition packages (see [Table 12 on page 130](#)). A fixed-length string is a string of digits that is always made up of the same number of digits. For example, a US social security number is a fixed-length number, since it always consists of nine digits. By contrast, a variable-length string indicates a string of digits that varies. For example, a street address is a variable-length string.

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 16 digits, use of a custom grammar is recommended (see [Custom Grammars on page 139](#) below). Using the provided variable-length grammar as an alternative results in decreased accuracy. If your application must accept a variable-length digit string, you may be able to increase recognition accuracy by using a two-step entry process. First prompt the caller for the number of digits, and then prompt for the digits.

For example:

“How many digits are in the next code?”

“Please say the three-digit code now.”

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

## Recognition DIPs

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

For some recognition types, there may be tradeoffs that determine how much of the structure of the input is built into the grammar and how much should be left for the DIP. For more information about custom DIPs, see “Writing the DIP” in Chapter 4, “Data Interface Processes,” in *UCS 1000 R4.6 Application Development with Advanced Methods*, 585-313-225. Also see [Custom Vocabulary on page 140](#).

## Word Spotting

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

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

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

## 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. The application can be designed to 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. Although the system can screen phrases, recognition accuracy is best when the application is written to encourage the caller to speak only what is required.

## Recognition Confirmation

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

## Barge-In

### Description

*Barge-in*, also called "recognize during prompt," is the ability of the system to allow callers to interrupt or barge in during voice playback by speaking a vocabulary word. Speech recognition accepts either speech or touchtone input in response to a prompt. Barge-in operates for speech much like the talkoff option does for touch-tone input, where a caller can interrupt the prompt by pressing a touch tone. See [Dual LSPS Tone Multifrequency \(DTMF\) Support on page 138](#) for more information on interrupts.

Experienced callers appreciate being able to shorten the transaction time by not being required to listen completely to each prompt. You can enable or disable barge-in for any of the prompts in your application.

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

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

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

### Enabling or Disabling Barge-In

At the beginning of an application, after the answer instruction, you can place an instruction to enable barge-in using the **sr\_talkoff** (TAS instruction). If a barge-in resource is available, it is reserved for this call. Otherwise the return code is negative, and barge-in will not be available. For more information see Appendix B, "Summary of TAS Script Instructions," in *UCS 1000 R4.6 Application Development with Advanced Methods*, 585-313-225.

**Discouraging Barge-In**

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

“For sales say, ‘one’.” (short pause)

“For service, say, ‘two’.” (short pause)

“To speak with a representative, say ‘three’.”

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

“You can order up to five copies. Please say how many copies you want.”

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

**Encouraging Barge-In**

The following prompts are designed to encourage callers to barge in when barge-in is enabled. Note that the pauses are longer than in the examples in [Discouraging Barge-In on page 137](#) above, so callers are encouraged to respond after hearing the desired option and the word “now.”

“For sales say, ‘one’ now.” (pause of 1-1.5 seconds)

“For service, say, ‘two’ now.” (pause of 1-1.5 second)

“For a representative, say ‘three’ now.” (pause of 1-1.5 seconds)

### System Response for Barge-In

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

- When a fixed-length digit recognition type is used on the SSP circuit card, the prompt can be interrupted after  $N-2$  ( $N$  minus 2) digits are spoken plus a system delay of 0.75 second, where  $N$  is the total number of digits to be collected. For example, for the US\_9DIG recognition type, the prompt can be interrupted after seven digits.
- When short prompts are used, callers might not detect delays in the barge-in point because the prompt frequently completes before the caller speaks much input.

## Dual LSPS Tone Multifrequency (DTMF) Support

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

Often the response time for talkoff is faster than for barge-in. This is because touch tones were designed to be recognized by machines and are rarely

confused with other sound or voice input. However, WholeWord speech recognition must perform the more complex task of separating the desired input from other sound or voice input that may simultaneously be present. The system does not terminate a prompt until it processes enough input to ensure with relatively high confidence that the input is valid. For a prompt requiring single-digit input, talkoff and barge-in may appear equally fast since the input is completed with a single touch tone or a single spoken digit. Prompts for multidigit input reveal the difference between the response times for talkoff and barge-in. See [Barge-In on page 135](#) above for information comparing and contrasting talkoff and barge-in.

## Custom Grammars

The system uses the recognition type, along with other fields, 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 LSPS II circuit card, which results in better accuracy.

For more information, contact your Lucent Technologies representative.

## Custom Vocabulary

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

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

A custom vocabulary requires application analysis by Lucent Technologies, speech data collection, model creation, and custom grammar work. Depending on your application, you may want to consider FlexWord™ speech recognition as a more cost-effective alternative. For more information, see [Chapter 8, Recognizing FlexWord™ Speech Input on the LSPS II Circuit Card](#).

For more information, contact your Lucent Technologies representative.

## Custom DIPs

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

## WholeWord Speech Recognition Accuracy

### Overview

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

### Positive Influences on WholeWord Speech Recognition Accuracy

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

**Note:** For more information on WholeWord Speech Recognition using the LSPS II circuit card, print the LSPS Software Development Kit resident on the system by moving the file **/usr/lsp/doc/pdf/sdk.pdf** from the system to a location that can be accessed by Acrobat Reader<sup>®</sup>.

### Isolated Word Recognition

Isolated word recognition is very high. The smaller the number of choices in an isolated word recognition type, the better the accuracy. For example, IW\_YES\_NO is either yes or no, which yields a very high recognition rate.

### Fixed-Length Digit String

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

### Validation of Data

Try to verify the recognized result against a database or a host field. This helps improve the overall accuracy of an application, especially when a longer string is input. The system uses a series of "prompt and collect" TAS instructions. The system responds with its "best guess" of what the user said to the TAS instructions below:

- talk or say set of instructions (play or TTS)
- **recog\_start** and **getinput**

**Note:** See Appendix B, "Summary of TAS Script Instruction," in *UCS 1000 R4.6 Application Development with Advanced Methods*, 585-313-225 for more information on the **recog\_start** and **getinput** instructions.

### Reprompt

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

### Prompt Structure

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

- Menu prompts

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

*<desired result> <action required>*

Examples:

“To hear your checking account balance, say 1.”

“To hear your savings account balance, say 2.”

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

- Yes and no prompts

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

“Would you like to hear your order again?”

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

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

This wording is more natural than the following:

“To hear your order again, say ‘yes’. Otherwise, say ‘no’.”

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

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

### Calling Experience and Informative Prompts

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

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

### Custom Grammars and DIPS

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

### Negative Influences on WholeWord Speech Recognition Accuracy

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

#### Environment

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

### Extraneous Words Within Responses

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

### Information Type

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

### Regional and National Accents and Dialects

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

### Connected-Digit String Length

Connected-digit string recognition can be thought of as a sequence of single-digit recognitions performed as one operation. For example, assume that the per-digit accuracy is  $X\%$  and that a digit string of one digit will be correct  $X\%$  of the time. Taking into consideration that this is a probabilistic, exponential model, when longer digit strings are used the overall expected accuracy will be  $X^n\%$ . Therefore, a two-digit string will have an overall expected accuracy

of  $X^2\%$  and a 10-digit string will have an overall accuracy of  $X^{10}\%$ . As a result, string accuracies are affected by the length of the string. Shorter string lengths are more accurate than longer string lengths. In addition, individual digit accuracies, as well as overall string accuracies, vary according to the language and noise conditions of different national networks.

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

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

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

### Application-Related Limitations

The capability of the system and WholeWord speech recognition is application dependent. If the system is underengineered for a particular application, it may not perform satisfactorily.

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

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

# 8 Recognizing FlexWord™ Speech Input on the LSPS II Circuit Card

## Introduction

**Note:** Information in this chapter pertains to the feature working only on the LSPS II circuit card.

This chapter provides the information necessary to create grammars and use them in applications. It includes:

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

**Note:** For more information on Grammar Recognition using the LSPS II circuit card, print the LSPS Software Development Kit document resident on the system by moving the file **/usr/lsp/doc/pdf/sdk.pdf** from the system to a location that can be accessed by Acrobat Reader®.

For more information about Advanced Speech Recognition (ASR) and Text-to-Speech using the LSPS II circuit card, print the LSPS ASR and TTS document resident on the system by moving the file **/usr/lsp/doc/pdf/asr\_tts.pdf** from the system to a location that can be accessed by Acrobat Reader.

## Description

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* (sometimes also called a *grammar*) is a set of entries, comprised of single words, phrases, or even whole sentences, that can be recognized with FlexWord speech recognition.
- A *vocabulary* is a set of wordlists associated with a particular FlexWord speech recognition application.

## How FlexWord Speech Recognition Works

FlexWord speech recognition uses subword technology. Subword technology relies on phonemic recognition for analyzing and recognizing words.

Either the speech and signal processor (SSP) circuit card *or* the Lucent Speech Processing Solutions (LSPSII) circuit card is required for all languages. If assigning FlexWord to the Lucent Speech Processing Solutions (LSPSII) circuit card you can not assign FlexWord to the SSP circuit card.

### 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 in the language. The word “sales,” for example, consists of four phonemes: s-A-l-z. Other languages use different phonemes.

Illustrated in [Table 13](#) are FlexWord speech recognition phonemes for the LSPS II circuit card in the US English language. A word accompanies each phoneme to illustrate its sound.

Table 13. US English Phonemes for the LSPS II Circuit Card

Phoneme	Word Example	Phoneme	Word Example
a	<i>father</i>	l	<i>led</i>
@	<i>bat</i>	m	<i>mom</i>
^	<i>but</i>	n	<i>no</i>
c	<i>bought</i>	G	<i>sing</i>
W	<i>bough</i>	o	<i>boat</i>
x	<i>again</i>	O	<i>boy</i>
Y	<i>bite</i>	p	<i>pop</i>
b	<i>bob</i>	r	<i>red</i>
C	<i>church</i>	s	<i>sis</i>
d	<i>dad</i>	S	<i>shoe</i>
D	<i>they</i>	t	<i>tot</i>
E	<i>bet</i>	T	<i>thief</i>
R	<i>bird</i>	U	<i>book</i>
e	<i>baït</i>	u	<i>boot</i>

1 of 2

Table 13. US English Phonemes for the LSPS II Circuit Card

Phoneme	Word Example	Phoneme	Word Example
f	<i>fief</i>	v	<i>very</i>
g	<i>gag</i>	w	<i>wet</i>
h	<i>hag</i>	y	<i>yet</i>
l	<i>bit</i>	z	<i>zoo</i>
i	<i>beat</i>	Z	<i>measure</i>
J	<i>judge</i>	#	<i>silence</i>
k	<i>kick</i>		

2 of 2

## FlexWord Capacity

Every entry in a wordlist is counted as a distinct and separate word. When using the LSPS II circuit card, and depending upon the lengths of the entries in the wordlists, the maximum number of words that can be loaded onto your system is in the range of 32,000.

Channel capacity of the LSPS II circuit card for FlexWord depends largely upon the complexity of the wordlist(s) being used. A software tool is provided

with the LSPS II circuit card to help estimate the channel capacity of an LSPS II circuit card for a given FlexWord wordlist. This tool, called **asr\_capacity**, is located in the **/usr/lsp/bin** directory.

To use this tool, at the system prompt, enter the following command:

```
asr_capacity -t <boardtype> -a <identifier>
```

where:

- *boardtype* is the type of circuit card file installed (see list of possible board types in [FlexWord Software and Hardware](#) below)
- *identifier* is the grammar's name as identified by **grammarName** in [Writing the Grammar](#) below.

The system returns a message telling you what the estimated channel capacity is for that grammar per LSPS II circuit card.

**Note:** Before you can use this tool, you must first create and compile your grammar using the **gc\_make** command (see [Compiling the Grammar](#)); and then assign it using the **mkboardtype** command. For the procedure, see the appropriate section from the following:

- [Assigning the Basic Flex to the LSPS II Circuit Card](#)
- [Assigning the Basic ASR Grammar with Flex](#)
- [Assigning the Basic ASR Grammar with Flex and TTS](#)
- [Assigning Basic Flex and TTS](#)

## Creating a Grammar

FlexWord supports fast, low-cost delivery of speech recognition vocabularies on a custom basis.

### Vocabularies

A FlexWord speech recognition vocabulary can include a maximum of 10,000 words or phrases per system. This is true whether you have one or more LSPS II circuit card installed. However, your system could become degraded if you use only one LSPS II circuit card and the maximum number of words. (For an application that requires more than 10,000 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," the system can recognize the word "checking" when a caller says "checking account please". However, to maintain high recognition accuracy, it is advisable to structure the application to prompt the caller for the required information only.

### 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 says the word "banking", the application can be

designed to inform the caller that the response is invalid and to replay the prompt.

### Barge-in

The LSPS II circuit does support barge-in for FlexWord speech recognition.

## Designing a Grammar

The LSPS II circuit card allows you to construct words and phrases by stringing phonemes together. The correct software and hardware are both necessary, and the hardware must be properly configured before you can create, edit, and delete wordlists.

## FlexWord Software and Hardware

For a list of the required software and hardware, see [FlexWord Speech Recognition on page 17](#) in [Chapter 1, Overview of Speech](#).

To install FlexWord (LSPSset software), see "Installing LSPS II Optional Software Packages" in *UCS 1000 R4.6 Maintenance*, 585-313-154.

**Note:** If you reinstall the LSPS Set (specifically, the LSPS II Integration package (**cslsps**)), it is possible that you will have to recreate your assignments for the following grammar files:

**CS\_BASIC\_FLEX-6UB5**

**CS\_BASIC\_ASR\_FLEX-6UB5**

**CS\_BASIC\_ASR\_FLEX\_TTS-6UB5**

**CS\_BASIC\_FLEX\_TTS-6UB5**

Currently, if you reinstall the **csIsps** package, the above files are saved as *o.filename* (such as **o.CS\_BASIC\_FLEX-6UB5**). These files can be used as a record of the previous grammar assignments.

- To configure the video controller circuit card, see “Setting Up the Monitor” in “Installing Base System Software” in *UCS 1000 R4.6 Maintenance*, 585-313-154.

## Configuring the LSPS II Circuit Card

Configuring the LSPS II circuit card for FlexWord functionality involves several steps. First, you must determine that you want to assign the function to an LSPS II circuit card. Then you must determine which LSPS II circuit card, if there is more than one, should be assigned FlexWord functionality. Use the following procedure to make the determination.

### 1 Enter **display card Isps**

The system displays output in the format shown in [Figure 12 on page 158](#).

### 2 Make a note of the card number.

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

See “Assign LSPS Functions” in Chapter 3, “Voice System Administration” in *UCS 1000 R4.6 Administration*, 585-313-509.

**Figure 12. Output from the Display Card Command After Assigning TTS Functionality**

```
CARD 12 STATE:Inserv CLASS: Signal_Processor (LSPS)O.S.INDEX:
12
      NAME: 6UB5  OPTIONS: slave,tdml
      FUNCTION: CS_BASIC_FLEX_TTS_6UB5+code+echocan+play+
                text2speech+flexword
```

Preparing a grammar-based application entails several steps and using the wordlists in scripts:

- 1 [Writing the Grammar on page 158](#)
- 2 [Compiling the Grammar on page 160](#)
- 3 [Assigning the Basic Flex to the LSPS II Circuit Card on page 162](#)

## Writing the Grammar

The first step in designing a wordlist is to create, or write the grammar file. The grammar file has a *grammarName* that should reflect the contents of the wordlist, i.e., *biglist*. Use a text editor to create the wordlist file in the following directory:

```
/vs/cslsps/asr/grammars/<grammarName>.gram
```

The following is the simple template for a typical **.gram** file, which is a Java Speech Grammar Format (JSGF) file:

```
#JSGF V1.0;  
grammar grammarName;  
public <grammarName>= GRAMMAR;
```

**Note:** A complete definition of the Java Speech Grammar Format (JSGF) is available at the following URL:

<http://java.sun.com/products/java-media/speech>

When creating this grammar file, keep the following in mind:

- The file name must have a **.gram** extension.
- The first line of the file must be: **#JSGF V1.0;**
- The second line of the file must be: **grammar *grammarName*;** where *grammarName* is the same as the file name minus the **.gram** extension.
- All statements must end with a semi-colon (;).
- Avoid underscores (for example, **ohio\_state**).
- Sub-grammar names must be enclosed with angle brackets (< >).
- Optional items must be enclosed with square brackets ( [ ] ).
- The "pipe symbol" ( | ) is used to designate OR options.

- Parentheses ( ) can be used to group items, as in the following example:  
**<foo>=(this | that) thing;**

An example of the contents for a **.gram** file is the following:

```
#JSGF V1.0;
grammar months;
public <months>=<monthlist>;
<monthlist>=
january |
february |
march |
april |
may |
june |
july |
august |
september |
october |
november |
december;
```

## Compiling the Grammar

The next step is to run the **gc\_make <grammarName>** command. The grammar compiler converts the contents of the file into a graph that is used by the LSPS II ASR resource. This script uses programs to expand macros, convert region markers into grammar tags, and invokes the compiler to compile the grammar.

Running the **gc\_make** command creates two more files, one file with a **.mv** extension and one file with a **.lex** extension. Following this step, you should see three files:

- **<grammarName>.gram**
- **<grammarName>.mv**
- **<grammarName>.lex**

The **.lex** file contains the actual lexicon of phonemic representations of the entries in the grammar file. If necessary, you can open and edit this file to achieve better recognition results. To edit this file, use the phoneme guide as found in [Table 13 on page 152](#).

## Editing Board Type Files

There are four board type files you must edit so that any combination that uses FlexWord the system will recognize the associated grammars. The files are the following and must be updated each time new grammars are added:

- **CS\_BASIC\_FLEX-6UB5**: This file includes the play and code functionality and all FlexWord grammars on the system. See [Assigning the Basic Flex to the LSPS II Circuit Card on page 162](#).
- **CS\_BASIC\_ASR\_FLEX-6UB5**: This file includes the play and code functionality and all FlexWord and WholeWord grammars on the system. See [Assigning the Basic ASR Grammar with Flex on page 172](#).

- **CS\_BASIC\_ASR\_FLEX\_TTS-6UB5:** This file includes the play and code functionality and all FlexWord, WholeWord, and TTS grammars on the system. See [Assigning the Basic ASR Grammar with Flex and TTS on page 182](#).
- **CS\_BASIC\_FLEX\_TTS-6UB5:** This file includes the play and code functionality and all FlexWord and TTS grammars on the system. See [Assigning Basic Flex and TTS on page 192](#).

### Assigning the Basic Flex to the LSPS II Circuit Card

Following is an example for assigning the basic flex grammar to the LSPS II circuit card:

#### 1 Enter **mkboardtype -f -t CS\_BASIC\_FLEX-6UB5**

The system displays the following message:

```
Board Model Count Description
1) Off AYC51 or BYC51 Model Boards
2) Off 9UD3 Model Boards
3) Off 6UB5 Model Boards
```

Select 1-3, or Next?

#### 2 Enter **3**

The system displays the following message:

```
Board Model Count Description
1) Off AYC51 or BYC51 Model Boards
```

- 2) Off 9UD3 Model Boards
- 3) Off 6UB5 Model Boards

Select 1-3, or Next?

### 3 Enter n

The system displays the following message:

```
Base Configuration (Select only one) Count Description
1) Off Basic configuration for the 6UB5
2) Off ASR configuration for the 6UB5
3) Off TTS configuration for the 6UB5
4) Off Combined ASR & TTS configuration for the 6UB
Select 1-4, or Next?
```

### 4 Enter 2

The system displays the following message:

```
Base Configuration (Select only one) Count Description
1) Off Basic configuration for the 6UB5
2) Off ASR configuration for the 6UB5
3) Off TTS configuration for the 6UB5
4) Off MLTTS configuration for the 6UB
Select 1-4, or Next?
```

### 5 Enter n

The system displays the following message:

```
ASR Configuration for the 6UB5 Count Description
1) 64 Play Module
2) 64 Record Module
3) 64 Dtmf DetectionModule
4) 64 Connect TimeslotModule
5) 64 TDD DetectionModule
6) 64 Tone Generation Module
7) 64 Call Progress Detection Module
8) 64 Translate coded speech Module
9) 36 ASR Module (select ONLY the quantity that is
    required!)
Select 1-9, or Next?
```

## 6 Enter n

The system displays the following message:

```
ASR Language Options (Select a max of 4)Count Description
1) Off North American English ASR Models

Select 1 or Next?
```

## 7 Enter 1

The system displays the following message:

```
ASR Language Options (Select a max of 4)Count Description
1) On North American English ASR Models
Select 1 or Next?
```

**8 Enter n**

The system displays the following message:

```
ASR North American English Options Count Description
1) On English Connected Digits
2) Off English Natural Digits
3) Off English Name Training
4) Off English Custom Grammar ASR
Select 1-4, or Next?
```

**9 Enter 2**

The system displays the following message:

```
ASR North American English Options Count Description
1) On English Connected Digits
2) On English Natural Digits
3) Off English Name Training
4) Off English Custom Grammar ASR
Select 1-4, or Next?
```

**10 Enter 4**

The system displays the following message:

```
ASR North American English Options Count Description
1) On English Connected Digits
2) On English Natural Digits
3) Off English Name Training
4) On English Custom Grammar ASR
Select 1-4, or Next?
```

**11 Enter n**

The system displays the following message:

```
ASR North American English Connected Digit Options Count
Description
1) On      English Isolated Digits
2) On      English Connected Digits of known length 2
3) On      English Connected Digits of known length 3
4) On      English Connected Digits of known length 4
5) On      English Connected Digits of known length 5
6) On      English Connected Digits of known length 6
7) On      English Connected Digits of known length 7
8) On      English Connected Digits of known length 8
9) On      English Connected Digits of known length 9
10) On     English Connected Digits of known length 10
11) On     English Connected Digits of known length 11
12) On     English Connected Digits of known length 12
```

- 13) On English Connected Digits of known length 13
- 14) On English Connected Digits of known length 14
- 15) On English Connected Digits of known length 15
- 16) On English Connected Digits of known length 16
- 17) On English Connected Digits of known length PHONE
- 18) On English Connected Digits of known length PHONE+
- 19) On English Connected Digits of unknown length
- 20) On English Yes/No Keywords

Select 1-20, or Next?

## 12 Enter n

The system displays the following message:

```
ASR North American English Natural Number Options Count
Description
1) Off English Natural Date Recognition
2) Off English Natural Dollar Recognition
3) Off English Natural Phone Number Recognition
4) Off English Natural Time Recognition
5) Off English Natural Credit Card Number Recognition
6) Off English Natural Credit Card Expiration Recognition
Select 1-4, or Next?
```

**13 Enter 1**

The system displays the following message:

```
ASR North American English Natural Number Options Count
Description
1) On English Natural Date Recognition
2) Off English Natural Dollar Recognition
3) Off English Natural Phone Number Recognition
4) Off English Natural Time Recognition
5) Off English Natural Credit Card Number Recognition
6) Off English Natural Credit Card Expiration Recognition
Select 1-4, or Next?
```

**14 Repeat [step 13](#) for each Natural Number Options listed.**

After entering **6**, the system displays the following message:

```
ASR North American English Natural Number Options
Count Description
1) On English Natural Date Recognition
2) On English Natural Dollar Recognition
3) On English Natural Phone Number Recognition
4) On English Natural Time Recognition
5) On English Natural Credit Card Number Recognition
6) On English Natural Credit Card Expiration Recognition
Select 1-4, or Next?
```

## 15 Enter n

The system displays the following message:

```
ASR North American English Custom Grammar Options
```

```
Count Description
```

```
1) Off English Custom Grammar 1
2) Off English Custom Grammar 2
3) Off English Custom Grammar 3
4) Off English Custom Grammar 4
5) Off English Custom Grammar 5
6) Off English Custom Grammar 6
7) Off English Custom Grammar 7
8) Off English Custom Grammar 8
9) Off English Custom Grammar 9
10) Off English Custom Grammar 10
11) Off English Custom Grammar 11
12) Off English Custom Grammar 12
13) Off English Custom Grammar 13
14) Off English Custom Grammar 14
15) Off English Custom Grammar 15
16) Off English Custom Grammar 16
17) Off English Custom Grammar 17
18) Off English Custom Grammar 18
19) Off English Custom Grammar 19
Select 1-19, or Next?
```

**16** Enter **1**

The system displays the following message:

```
Custom Base Grammar Specification
Enter Custom ASR Name Identifier (such as GC_MYGRAM1)?
```

**17** Enter the **grammarName** identified earlier.

The system displays the following message:

```
Grammar Type Options:
1) gc
2) free
3) namelist
Select 1 - 3 ?
```

**18** Enter **1**

The system displays the following message:

```
Enter Compiled Grammar Filename ?
```

**19** Enter **/vs/cslsps/asr/grammars/grammarName/grammarName.mv**

The system displays the following message:

```
Enter Lexicon Filename (optional) ?
```

**20** Enter **/vs/cslsps/asr/grammars/grammarName/grammarName.lex**

The system displays the following message:

```
Grammar Type: gc
Grammar File: /vs/cslsps/asr/grammars/grammarName/grammarName.mv
Grammar Lexicon:
/vs/cslsps/asr/grammars/grammarName/grammarName.lex
Is this correct (y/n)?
```

## 21 Enter y

The system displays the following message:

```
Would you like to add a sub grammar (y/n)?
```

## 22 Enter n

The system displays the following message:

```
ASR North American English Custom Grammar Options Count
Description
1) On English Custom Grammar 1
2) Off English Custom Grammar 2
3) Off English Custom Grammar 3
4) Off English Custom Grammar 4
5) Off English Custom Grammar 5
6) Off English Custom Grammar 6
7) Off English Custom Grammar 7
8) Off English Custom Grammar 8
9) Off English Custom Grammar 9
10) Off English Custom Grammar 10
11) Off English Custom Grammar 11
```

```
12) Off English Custom Grammar 12
13) Off English Custom Grammar 13
14) Off English Custom Grammar 14
15) Off English Custom Grammar 15
16) Off English Custom Grammar 16
17) Off English Custom Grammar 17
18) Off English Custom Grammar 18
19) Off English Custom Grammar 19
Select 1-19, or Next?
```

### 23 Enter n

The system displays the system prompt.

## Assigning the Basic ASR Grammar with Flex

To assign the basic ASR grammar with Flex, do the following:

### 1 Enter **mkboardtype -f -t CS\_BASIC\_ASR\_FLEX-6UB5**

The system displays the following message:

```
Board Model Count Description
1) Off AYC51 or BYC51 Model Boards
2) Off 9UD3 Model Boards
3) Off 6UB5 Model Boards
Select 1-3, or Next?
```

### 2 Enter 3

The system displays the following message:

```
Board Model Count Description
1) Off   AYC51 or BYC51 Model Boards
2) Off   9UD3 Model Boards
3) On    6UB5 Model Boards
```

Select 1-3, or Next?

### 3 Enter n

The system displays the following message:

```
Base Configuration (Select only one) Count Description
1) Off   Basic configuration for the 6UB5
2) Off   ASR configuration for the 6UB5
3) Off   TTS configuration for the 6UB5
4) Off   Combined ASR & TTS configuration for the 6UB5
```

Select 1-4, or Next?

### 4 Enter 2

The system displays the following message:

```
Base Configuration (Select only one) Count Description
1) Off   Basic configuration for the 6UB5
2) On    ASR configuration for the 6UB5
3) Off   TTS configuration for the 6UB5
4) Off   Combined ASR & TTS configuration for the 6UB5
```

Select 1-4, or Next?

## 5 Enter n

The system displays the following message:

```
ASR configuration for the 6UB5 Count Description
1) 64    Play Module
2) 64    Record Module
3) 64    Dtmf Detection Module
4) 64    Connect Timeslot Module
5) 64    TDD Detection Module
6) 64    Tone Generation Module
7) 64    Call Progress Detection Module
8) 64    Translate coded speech Module
9) 36    ASR Module (Select ONLY the quantity that is
          required!)
```

Select 1-9, or Next?

## 6 Enter n

The system displays the following message:

```
ASR Language Options (Select a max of 4) Count Description
1) Off   North American English ASR Models

Select 1, or Next?
```

**7 Enter 1**

The system displays the following message:

```
ASR Language Options (Select a max of 4) Count Description
1) On    North American English ASR Models

Select 1, or Next?
```

**8 Enter n**

The system displays the following message:

```
ASR North American English Options Count Description
1) On    English Connected Digits
2) Off   English Natural Numbers
3) Off   English Name Training
4) Off   English Custom Grammar ASR

Select 1-4, or Next?
```

**9 Enter 2**

The system displays the following message:

```
ASR North American English Options Count Description
1) On    English Connected Digits
2) On    English Natural Numbers
3) Off   English Name Training
4) Off   English Custom Grammar ASR
```

Select 1-4, or Next?

## 10 Enter 4

The system displays the following message:

ASR North American English Options Count Description

- 1) On English Connected Digits
- 2) On English Natural Numbers
- 3) Off English Name Training
- 4) On English Custom Grammar ASR

Select 1-4, or Next?

## 11 Enter n

The system displays the following message:

ASR North American English Connected Digit Options Count  
Description

- 1) On English Isolated Digits
- 2) On English Connected Digits of known length 2
- 3) On English Connected Digits of known length 3
- 4) On English Connected Digits of known length 4
- 5) On English Connected Digits of known length 5
- 6) On English Connected Digits of known length 6
- 7) On English Connected Digits of known length 7
- 8) On English Connected Digits of known length 8
- 9) On English Connected Digits of known length 9

- 10) On English Connected Digits of known length 10
- 11) On English Connected Digits of known length 11
- 12) On English Connected Digits of known length 12
- 13) On English Connected Digits of known length 13
- 14) On English Connected Digits of known length 14
- 15) On English Connected Digits of known length 15
- 16) On English Connected Digits of known length 16
- 17) On English Connected Digits of known length PHONE
- 18) On English Connected Digits of known length PHONE+
- 19) On English Connected Digits of unknown length
- 20) On English Yes/No Keywords

Select 1-20, or Next?

## 12 Enter n

The system displays the following message:

```
ASR North American English Natural Number Options Count
Description
1) Off English Natural Date Recognition
2) Off English Natural Dollar Recognition
3) Off English Natural Phone Number Recognition
4) Off English Natural Time Recognition
5) Off English Natural Credit Card Number Recognition
6) Off English Natural Credit Card Expiration Recognition
```

Select 1-6, or Next?

**13 Enter 1**

The system displays the following message:

```
ASR North American English Natural Number Options Count
Description
1) On      English Natural Date Recognition
2) Off     English Natural Dollar Recognition
3) Off     English Natural Phone Number Recognition
4) Off     English Natural Time Recognition
5) Off     English Natural Credit Card Number Recognition
6) Off     English Natural Credit Card Expiration Recognition
```

Select 1-6, or Next?

**14 Repeat [step 13](#) for each Natural Number Options listed.**

After entering **6**, the system displays the following message:

```
ASR North American English Natural Number Options Count
Description
1) On      English Natural Date Recognition
2) On      English Natural Dollar Recognition
3) On      English Natural Phone Number Recognition
4) On      English Natural Time Recognition
5) On      English Natural Credit Card Number Recognition
6) On      English Natural Credit Card Expiration Recognition
Select 1-6, or Next?
```

## 15 Enter n

The system displays the following message:

```
ASR North American English Custom Grammar Options Count  
Description
```

```
1) Off English Custom Grammar 1  
2) Off English Custom Grammar 2  
3) Off English Custom Grammar 3  
4) Off English Custom Grammar 4  
5) Off English Custom Grammar 5  
6) Off English Custom Grammar 6  
7) Off English Custom Grammar 7  
8) Off English Custom Grammar 8  
9) Off English Custom Grammar 9  
10) Off English Custom Grammar 10  
11) Off English Custom Grammar 11  
12) Off English Custom Grammar 12  
13) Off English Custom Grammar 13  
14) Off English Custom Grammar 14  
15) Off English Custom Grammar 15  
16) Off English Custom Grammar 16  
17) Off English Custom Grammar 17  
18) Off English Custom Grammar 18  
19) Off English Custom Grammar 19
```

```
Select 1-19, or Next?
```

**16 Enter 1**

The system displays the following message:

```
Custom Base Grammar Specification
Enter Custom ASR Name Identifier (such as GC_MYGRAM1)?
```

**17 Enter the *grammarName* identified earlier.**

The system displays the following message:

```
Grammar Type Options:
1) gc
2) free
3) namelist
Select 1 - 3 ?
```

**18 Enter 1**

The system displays the following message:

```
Enter Compiled Grammar Filename ?
```

**19 Enter */vs/cslsps/asr/grammars/grammarName/grammarName.mv***

The system displays the following message:

```
Enter Lexicon Filename (optional) ?
```

**20 Enter */vs/cslsps/asr/grammars/grammarName/grammarName.lex***

The system displays the following message:

```
Grammar Type: gc
Grammar File: /vs/cslsps/asr/grammars/grammarName/grammarName.mv
Grammar Lexicon:
/vs/cslsps/asr/grammars/grammarName/grammarName.lex
Is this correct (y/n)?
```

## 21 Enter y

The system displays the following message:

```
Would you like to add a sub grammar (y/n)?
```

## 22 Enter n

The system displays the following message:

```
ASR North American English Custom Grammar Options Count
Description
1) On      English Custom Grammar 1
2) Off     English Custom Grammar 2
3) Off     English Custom Grammar 3
4) Off     English Custom Grammar 4
5) Off     English Custom Grammar 5
6) Off     English Custom Grammar 6
7) Off     English Custom Grammar 7
8) Off     English Custom Grammar 8
9) Off     English Custom Grammar 9
10) Off    English Custom Grammar 10
11) Off    English Custom Grammar 11
12) Off    English Custom Grammar 12
```

```
13) Off      English Custom Grammar 13
14) Off      English Custom Grammar 14
15) Off      English Custom Grammar 15
16) Off      English Custom Grammar 16
17) Off      English Custom Grammar 17
18) Off      English Custom Grammar 18
19) Off      English Custom Grammar 19
Select 1-19, or Next?
```

### 23 Enter n

The system displays the system prompt.

#### Assigning the Basic ASR Grammar with Flex and TTS

To assign the basic ASR grammar with Flex and TTS, do the following:

#### 1 Enter **mkboardtype -f -t CS\_BASIC\_ASR\_FLEX\_TTS-6UB5**

The system displays the following message:

```
Board Model Count Description
1) Off      AYC51 or BYC51 Model Boards
2) Off      9UD3 Model Boards
3) Off      6UB5 Model Boards

Select 1-3, or Next?
```

#### 2 Enter 3

The system displays the following message:

```
Board Model Count Description
1) Off      AYC51 or BYC51 Model Boards
```

- 2) Off 9UD3 Model Boards
- 3) On 6UB5 Model Boards

Select 1-3, or Next?

### 3 Enter n

The system displays the following message:

```
Base Configuration (Select only one) Count Description
1) Off Basic configuration for the 6UB5
2) Off ASR configuration for the 6UB5
3) Off TTS configuration for the 6UB5
4) Off Combined ASR & TTS configuration for the 6UB5
```

Select 1-4, or Next?

### 4 Enter 4

The system displays the following message:

```
Base Configuration (Select only one) Count Description
1) Off Basic configuration for the 6UB5
2) Off ASR configuration for the 6UB5
3) Off TTS configuration for the 6UB5
4) On Combined ASR & TTS configuration for the 6UB5
```

Select 1-4, or Next?

### 5 Enter n

The system displays the following message:

Combines ASR & TTS configuration for the 6UB5 Count  
Description

- 1) 64 Play Module
- 2) 64 Record Module
- 3) 64 Dtmf Detection Module
- 4) 64 Connect Timeslot Module
- 5) 64 TDD Detection Module
- 6) 64 Tone Generation Module
- 7) 64 Call Progress Detection Module
- 8) 64 Translate coded speech Module
- 9) 30 Text to Speech Module
- 10) 24 ASR Module (Select ONLY the quantity that is  
required!)

Select 1-10, or Next?

## 6 Enter n

The system displays the following message:

ASR Language Options (Select a max of 4) Count Description

- 1) Off North American English ASR Models

Select 1, or Next?

## 7 Enter 1

The system displays the following message:

ASR Language Options (Select a max of 4) Count Description

- 1) On North American English ASR Models

Select 1, or Next?

### 8 Enter n

The system displays the following message:

```
ASR North American English Options Count Description
1) On      English Connected Digits
2) Off     English Natural Numbers
3) Off     English Name Training
4) Off     English Custom Grammar ASR
```

Select 1-4, or Next?

### 9 Enter 2

The system displays the following message:

```
ASR North American English Options Count Description
1) On      English Connected Digits
2) On      English Natural Numbers
3) Off     English Name Training
4) Off     English Custom Grammar ASR
```

Select 1-4, or Next?

### 10 Enter 4

The system displays the following message:

```
ASR North American English Options Count Description
1) On English Connected Digits
2) On English Natural Numbers
3) Off English Name Training
4) On English Custom Grammar ASR
```

Select 1-4, or Next?

## 11 Enter n

The system displays the following message:

```
ASR North American English Connected Digit Options Count
Description
1) On English Isolated Digits
2) On English Connected Digits of known length 2
3) On English Connected Digits of known length 3
4) On English Connected Digits of known length 4
5) On English Connected Digits of known length 5
6) On English Connected Digits of known length 6
7) On English Connected Digits of known length 7
8) On English Connected Digits of known length 8
9) On English Connected Digits of known length 9
10) On English Connected Digits of known length 10
11) On English Connected Digits of known length 11
12) On English Connected Digits of known length 12
```

- 13) On English Connected Digits of known length 13
- 14) On English Connected Digits of known length 14
- 15) On English Connected Digits of known length 15
- 16) On English Connected Digits of known length 16
- 17) On English Connected Digits of known length PHONE
- 18) On English Connected Digits of known length PHONE+
- 19) On English Connected Digits of unknown length
- 20) On English Yes/No Keywords

Select 1-20, or Next?

## 12 Enter n

The system displays the following message:

ASR North American English Natural Number Options Count  
Description

- 1) Off English Natural Date Recognition
- 2) Off English Natural Dollar Recognition
- 3) Off English Natural Phone Number Recognition
- 4) Off English Natural Time Recognition
- 5) Off English Natural Credit Card Number Recognition
- 6) Off English Natural Credit Card Expiration Recognition

Select 1-6, or Next?

## 13 Enter 1

The system displays the following message:

```
ASR North American English Natural Number Options Count
Description
1) On      English Natural Date Recognition
2) Off     English Natural Dollar Recognition
3) Off     English Natural Phone Number Recognition
4) Off     English Natural Time Recognition
5) Off     English Natural Credit Card Number Recognition
6) Off     English Natural Credit Card Expiration Recognition
```

Select 1-6, or Next?

- 14** Repeat [step 13](#) for each Natural Number Options listed.

After entering **6**, the system displays the following message:

```
ASR North American English Natural Number Options Count
Description
1) On      English Natural Date Recognition
2) On      English Natural Dollar Recognition
3) On      English Natural Phone Number Recognition
4) On      English Natural Time Recognition
5) On      English Natural Credit Card Number Recognition
6) On      English Natural Credit Card Expiration Recognition
Select 1-6, or Next?
```

- 15** Enter n

The system displays the following message:

```
ASR North American English Custom Grammar Options Count
Description
1) Off English Custom Grammar 1
2) Off English Custom Grammar 2
3) Off English Custom Grammar 3
4) Off English Custom Grammar 4
5) Off English Custom Grammar 5
6) Off English Custom Grammar 6
7) Off English Custom Grammar 7
8) Off English Custom Grammar 8
9) Off English Custom Grammar 9
10) Off English Custom Grammar 10
11) Off English Custom Grammar 11
12) Off English Custom Grammar 12
13) Off English Custom Grammar 13
14) Off English Custom Grammar 14
15) Off English Custom Grammar 15
16) Off English Custom Grammar 16
17) Off English Custom Grammar 17
18) Off English Custom Grammar 18
19) Off English Custom Grammar 19
```

Select 1-19, or Next?

## 16 Enter 1

The system displays the following message:

```
Custom Base Grammar Specification
Enter Custom ASR Name Identifier (such as GC_MYGRAM1)?
```

- 17 Enter the **grammarName** identified earlier.

The system displays the following message:

```
Grammar Type Options:
1) gc
2) free
3) namelist
Select 1 - 3 ?
```

- 18 Enter 1

The system displays the following message:

```
Enter Compiled Grammar Filename ?
```

- 19 Enter **/vs/cslsps/asr/grammars/grammarName/grammarName.mv**

The system displays the following message:

```
Enter Lexicon Filename (optional) ?
```

- 20 Enter **/vs/cslsps/asr/grammars/grammarName/grammarName.lex**

The system displays the following message:

```
Grammar Type: gc
Grammar File: /vs/cslsps/asr/grammars/grammarName/grammarName.mv
Grammar Lexicon:
```

```
/vs/cslsps/asr/grammars/grammarName/grammarName.lex  
Is this correct (y/n)?
```

## 21 Enter y

The system displays the following message:

```
Would you like to add a sub grammar (y/n)?
```

## 22 Enter n

The system displays the following message:

```
ASR North American English Custom Grammar Options Count  
Description  
1) On      English Custom Grammar 1  
2) Off     English Custom Grammar 2  
3) Off     English Custom Grammar 3  
4) Off     English Custom Grammar 4  
5) Off     English Custom Grammar 5  
6) Off     English Custom Grammar 6  
7) Off     English Custom Grammar 7  
8) Off     English Custom Grammar 8  
9) Off     English Custom Grammar 9  
10) Off    English Custom Grammar 10  
11) Off    English Custom Grammar 11  
12) Off    English Custom Grammar 12  
13) Off    English Custom Grammar 13  
14) Off    English Custom Grammar 14  
15) Off    English Custom Grammar 15
```

```
16) Off      English Custom Grammar 16
17) Off      English Custom Grammar 17
18) Off      English Custom Grammar 18
19) Off      English Custom Grammar 19
Select 1-19, or Next?
```

### 23 Enter n

The system displays the system prompt.

## Assigning Basic Flex and TTS

To assign the basic Flex and TTS, do the following:

#### 1 Enter **mkboardtype -f -t CS\_BASIC\_FLEX\_TTS-6UB5**

The system displays the following message:

```
Board Model Count Description
1) Off      AYC51 or BYC51 Model Boards
2) Off      9UD3 Model Boards
3) Off      6UB5 Model Boards

Select 1-3, or Next?
```

#### 2 Enter 3

The system displays the following message:

```
Board Model Count Description
1) Off      AYC51 or BYC51 Model Boards
2) Off      9UD3 Model Boards
3) On       6UB5 Model Boards
```

Select 1-3, or Next?

### 3 Enter n

The system displays the following message:

```
Base Configuration (Select only one) Count Description
1) Off   Basic configuration for the 6UB5
2) Off   ASR configuration for the 6UB5
3) Off   TTS configuration for the 6UB5
4) Off   Combined ASR & TTS configuration for the 6UB5
```

Select 1-4, or Next?

### 4 Enter 4

The system displays the following message:

```
Base Configuration (Select only one) Count Description
1) Off   Basic configuration for the 6UB5
2) Off   ASR configuration for the 6UB5
3) Off   TTS configuration for the 6UB5
4) On    Combined ASR & TTS configuration for the 6UB5
```

Select 1-4, or Next?



**7 Enter 1**

The system displays the following message:

```
ASR Language Options (Select a max of 4) Count Description
1) On    North American English ASR Models

Select 1, or Next?
```

**8 Enter n**

The system displays the following message:

```
ASR North American English Options Count Description
1) On    English Connected Digits
2) Off   English Natural Numbers
3) Off   English Name Training
4) Off   English Custom Grammar ASR

Select 1-4, or Next?
```

**9 Enter 2**

The system displays the following message:

```
ASR North American English Options Count Description
1) On    English Connected Digits
2) On    English Natural Numbers
3) Off   English Name Training
4) Off   English Custom Grammar ASR
```

Select 1-4, or Next?

## 10 Enter 4

The system displays the following message:

```
ASR North American English Options Count Description
1) On      English Connected Digits
2) On      English Natural Numbers
3) Off     English Name Training
4) On      English Custom Grammar ASR
```

Select 1-4, or Next?

## 11 Enter n

The system displays the following message:

```
ASR North American English Connected Digit Options Count
Description
1) On      English Isolated Digits
2) On      English Connected Digits of known length 2
3) On      English Connected Digits of known length 3
4) On      English Connected Digits of known length 4
5) On      English Connected Digits of known length 5
6) On      English Connected Digits of known length 6
7) On      English Connected Digits of known length 7
8) On      English Connected Digits of known length 8
```

- 9) On English Connected Digits of known length 9
- 10) On English Connected Digits of known length 10
- 11) On English Connected Digits of known length 11
- 12) On English Connected Digits of known length 12
- 13) On English Connected Digits of known length 13
- 14) On English Connected Digits of known length 14
- 15) On English Connected Digits of known length 15
- 16) On English Connected Digits of known length 16
- 17) On English Connected Digits of known length PHONE
- 18) On English Connected Digits of known length PHONE+
- 19) On English Connected Digits of unknown length
- 20) On English Yes/No Keywords

Select 1-20, or Next?

## 12 Enter n

The system displays the following message:

```
ASR North American English Natural Number Options Count
Description
1) Off English Natural Date Recognition
2) Off English Natural Dollar Recognition
3) Off English Natural Phone Number Recognition
4) Off English Natural Time Recognition
5) Off English Natural Credit Card Number Recognition
6) Off English Natural Credit Card Expiration Recognition
```

Select 1-6, or Next?

**13 Enter 1**

The system displays the following message:

```
ASR North American English Natural Number Options Count
Description
1) On      English Natural Date Recognition
2) Off     English Natural Dollar Recognition
3) Off     English Natural Phone Number Recognition
4) Off     English Natural Time Recognition
5) Off     English Natural Credit Card Number Recognition
6) Off     English Natural Credit Card Expiration Recognition
```

Select 1-6, or Next?

**14 Repeat [step 13](#) for each Natural Number Options listed.**

After entering **6**, the system displays the following message:

```
ASR North American English Natural Number Options Count
Description
1) On      English Natural Date Recognition
2) On      English Natural Dollar Recognition
3) On      English Natural Phone Number Recognition
4) On      English Natural Time Recognition
5) On      English Natural Credit Card Number Recognition
6) On      English Natural Credit Card Expiration Recognition
Select 1-6, or Next?
```

## 15 Enter n

The system displays the following message:

```
ASR North American English Custom Grammar Options Count  
Description
```

```
1) Off English Custom Grammar 1  
2) Off English Custom Grammar 2  
3) Off English Custom Grammar 3  
4) Off English Custom Grammar 4  
5) Off English Custom Grammar 5  
6) Off English Custom Grammar 6  
7) Off English Custom Grammar 7  
8) Off English Custom Grammar 8  
9) Off English Custom Grammar 9  
10) Off English Custom Grammar 10  
11) Off English Custom Grammar 11  
12) Off English Custom Grammar 12  
13) Off English Custom Grammar 13  
14) Off English Custom Grammar 14  
15) Off English Custom Grammar 15  
16) Off English Custom Grammar 16  
17) Off English Custom Grammar 17  
18) Off English Custom Grammar 18  
19) Off English Custom Grammar 19
```

Select 1-19, or Next?

**16 Enter 1**

The system displays the following message:

```
Custom Base Grammar Specification
Enter Custom ASR Name Identifier (such as GC_MYGRAM1)?
```

**17 Enter the *grammarName* identified earlier.**

The system displays the following message:

```
Grammar Type Options:
1) gc
2) free
3) namelist
Select 1 - 3 ?
```

**18 Enter 1**

The system displays the following message:

```
Enter Compiled Grammar Filename ?
```

**19 Enter */vs/cslsps/asr/grammars/grammarName/grammarName.mv***

The system displays the following message:

```
Enter Lexicon Filename (optional) ?
```

**20 Enter */vs/cslsps/asr/grammars/grammarName/grammarName.lex***

The system displays the following message:

```
Grammar Type: gc
Grammar File: /vs/cslsps/asr/grammars/grammarName/grammarName.mv
Grammar Lexicon:
/vs/cslsps/asr/grammars/grammarName/grammarName.lex
Is this correct (y/n)?
```

## 21 Enter y

The system displays the following message:

```
Would you like to add a sub grammar (y/n)?
```

## 22 Enter n

The system displays the following message:

```
ASR North American English Custom Grammar Options Count
Description
1) On      English Custom Grammar 1
2) Off     English Custom Grammar 2
3) Off     English Custom Grammar 3
4) Off     English Custom Grammar 4
5) Off     English Custom Grammar 5
6) Off     English Custom Grammar 6
7) Off     English Custom Grammar 7
8) Off     English Custom Grammar 8
9) Off     English Custom Grammar 9
10) Off    English Custom Grammar 10
11) Off    English Custom Grammar 11
12) Off    English Custom Grammar 12
```

```
13) Off      English Custom Grammar 13
14) Off      English Custom Grammar 14
15) Off      English Custom Grammar 15
16) Off      English Custom Grammar 16
17) Off      English Custom Grammar 17
18) Off      English Custom Grammar 18
19) Off      English Custom Grammar 19
Select 1-19, or Next?
```

### 23 Enter n

The system displays the system prompt.

## LSPS Usage Data Files

The Resource Manager (RM) uses data to determine resource usage requirements for functions that can be assigned to the LSPS II circuit card. Normally, data contained in the usage files do not require any changes because of applications being used by the LSPS II circuit card.

However, there are possible exceptions that require changing the usage data file contents. Possible exceptions are:

- Custom Flex grammar required more resource than the default value. This could occur if large vocabulary grammars are used. In this instance, Usage0 should be changed to a value that represents the portion of the

LSPS CPU capacity that is consumed. If different sizes of grammars are used, the usage data should be set based on the capacity of the most expensive grammar.

- Measured occupancy of the LSPS II circuit card shows significant difference from the TM view of the board load.

### Usage Data Files That are Changeable

[Table 14 on page 203](#) contains the fields in the LSPS II usage data files that can be changed, when necessary, and their meanings:

**Table 14. LSPS II Changeable Usage Data Fields**

Field*	Definition
MAX_CONCURRENT	Indicates the maximum number of times this function can be allocated on any LSPS II circuit card.
MAX_WITH_TTS	Applies only to ASR functions and is used to indicate when the TTS function is also assigned to this card.
MAX_WITH_ASR	Applies only to the TTS function and is used to indicate a lower MAX_CONCURRENT capacity for this TTS function when an ASR function is also assigned to this card.

1 of 2

Table 14. LSPS II Changeable Usage Data Fields

Field*	Definition
USAGE0	Indicated the portion of the total board CPU and memory capacity that is consumed for each occurrence of the functions. The total board capacity is 10,000 so a value of 100 represents 1% of the board's capacity.
USAGE1	Indicates the portion of the Input (coding) I/O capacity that is consumed for each occurrence of the function. A value such as 312 for the code function implies that 1/21 of the board capacity is used.
USAGE2	Indicates the portion of the Output (Play) I/O capacity that is consumed for each occurrence of the function. A value such as 312 for the play function implies that 1/32 of the board capacity is used.
USAGE3	Indicates the portion of the PCI I/O capacity that is consumed for each occurrence of the function. A value of 100 implies that 1/100 of the board capacity is used.

2 of 2

\* Fields not listed in this table must not be changed.

### Editing a Usage Data File

Default values for the LSPS II capacities can be found in Chapter 2, "Hardware," in *UCS 1000 R4.6 System Description*, 585-313-222. A limited listing is also located in [Commands for SSP Circuit Card Only on page 21](#).

The data usage is found in */vs/cslsps/usage/function* where *function* may be one of the following:

- play
- code
- wholeword
- flexword
- text2speech
- LSPS\_DATA\_COLLECT
- BUS\_DIAG

### Example

**Note:** The **cslsps** software package *will* erase any changes to a data usage file if **cslsps** is reinstalled after modifications are made. Ensure that the data usage file is updated properly after any reinstallation of the **cslsps** software package.

For example, a user may decide that it is necessary to modify data in the **flexword** data usage file. A few circumstances for this modification are:

- Using small grammars, connected digits and less than 50 word grammars, means that one can increase the LSPS II channel capacity.
- Using large or very large vocabulary grammars, the channel capacity must be reduced.

Continuing with the example, to modify the **flexword** file to accommodate very large grammars (those over 2000 words), do the following:

- 1 Open the following file in a text editor: **/vs/cslsps/usage/flexword**
- 2 Change the following lines:

```
MAX_CONCURRENT=6  
MAX_WITH_TTS=4  
USAGE0=1666
```

- 3 Write and exit the file.

In the above example, very large wordlists are accommodated but the channel capacity of the LSPS II circuit card is changed to 6 channels for FlexWord speech recognition or 4 channels for FlexWord speech recognition with Text-to-Speech assigned to the same card.

[Table 15 on page 207](#) contains some suggested starting values for various size FlexWord grammars:

Table 15. Suggested Start Values for FlexWord Grammars

Function	MAX_CURRENT	MAX_WITH_TTS	USAGE0
Default	24	14	416
FlexWord with connected digit or less than 50 word grammars	30	18	333
FlexWord with 50 to 2000 word grammars	15	8	666
FlexWord with over 2000 word grammars	6	4	1666

**Note:** Application developers are responsible for choosing the appropriate values and validating them as part of using a custom grammar. For further information, print the LSPS Software Development Kit resident on the system by moving the file **/usr/lsp/doc/pdf/sdk.pdf** from the system to a location that can be accessed by Acrobat Reader.

# 9 Putting It Together with the LSPS II Circuit Card

## Overview

**Note:** Information in this chapter pertains to the features working only on the LSPS II circuit card.

This chapter describes the following considerations for using speech-related features with optimal effectiveness in applications:

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

Recognition types, including FlexWord™ speech recognition and WholeWord speech recognition are discussed, as well as the TTS (Text-to-Speech) feature.

**Note:** In order for barge in to work, recognition and echo cancellation must be assigned on the same card.

## Using WholeWord Recognition

WholeWord speech recognition can be used to recognize a limited set of words, as discussed in [Chapter 7. Recognizing WholeWord Speech Input on the LSPS II](#). However, WholeWord speech recognition is most successful when you use it to augment a touchtone application to handle callers who do not have or for some reason do not want to use touch-tone telephones.

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

## Getting the Most Out of FlexWord Speech Recognition

FlexWord speech recognition recognizes the caller speaking words from a vocabulary that you define. It is, therefore, specifically tailored to your application. Allowing callers to speak the options they want rather than speaking a number assigned to the option can make the interaction more natural and easy to use. Included below are guidelines for getting the most out of FlexWord speech recognition.

### Choosing a FlexWord Speech Recognition Vocabulary

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

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

### Caller Error

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

- Word Length** Words with more than one syllable are recognized more reliably than shorter words. For example, the phrase “add entry” would be better to use than the word “add.” Phrases can be fairly long, although the longer the phrase, the greater the chance of callers forgetting it or making mistakes while speaking.
- Word Choice** Choose words and phrases that would occur naturally to your typical caller. One way to determine this is to survey some representative callers and ask what words or phrases they would use for actions in your script. Do not rely on just your knowledge, because your impressions may be different than those of your callers.
- Prompting for Input** Because FlexWord speech recognition does not support barge-in, structure your prompts to encourage callers to wait until the end of the prompt before speaking. For example, use a prompt such as, “Please say the department name, now,” as opposed to “Now say the department name, please.” This is good practice for all applications, whether or not barge-in is supported.
- Word Sounds** Use words in your wordlists that do not have a similar sound. To the recognizer, vowels are more important than consonants. Avoid using words with similar vowel sounds. Words that differ in only one or two consonants are difficult for the recognizer to distinguish between. For example, the words “connect” and “comment” sound similar to the recognizer.
- Synonyms** If you find that callers often use more than one word for the same thing, you may want to add both words to your list.

## Examples of FlexWord Speech Recognition Applications

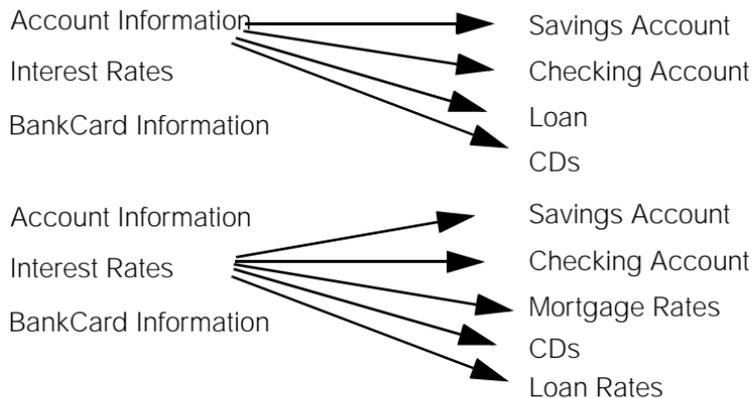
Two examples of applications suited to FlexWord speech recognition are described below.

### Applications Requiring Large Wordlists

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

### Menu-Based Applications

Menu-based applications that use words and phrases native to your business environment are suggested for use with FlexWord speech recognition. Your application flows more smoothly if it uses the same words that would occur naturally in an interaction between an agent and the customer. [Figure 13 on page 213](#) shows an example of a menu-based application.

**Figure 13. Example of Menu-Based Application**

## Getting the Most out of Text-to-Speech

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

- When TTS prompts are used with touch-tone input, callers can interrupt the prompt just as they can when the prompt is prerecorded.
- When using TTS prompts with FlexWord speech recognition input, barge-in is not available. Therefore, when accepting FlexWord speech recognition input, structure your TTS prompts so that they encourage callers to wait until the end before responding. Note that this is good practice for all applications, not just those for which barge-in is not supported.

**Note:** Standard TTS is available for US English only. However, TTS is available for other languages using the Proxy Text-to-Speech (PTTS) feature package, which allows the TTS to be processed by an external (proxy) server. For more details, see *Proxy Text-to-Speech (PTTS) User Guide*, 585-350-115.

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

Since FlexWord speech recognition provides the ability to recognize up to 10,000 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 `SCI_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:** Standard TTS is available for US English only. However, TTS is available for other languages using the Proxy Text-to-Speech (PTTS) feature package, which allows the TTS to be processed by an external (proxy) server. For more details, see *Proxy Text-to-Speech (PTTS) User Guide*, 585-350-115.

# Using Text-to-Speech and Prerecorded Speech Together

A UCS 1000 R4.6 system application can speak prompts and announcements in prerecorded speech only, TTS only (including PTTS), or a combination of prerecorded speech and TTS.

For example, in one application, recorded speech is used for all phrases except the customer's name. TTS allows the system to speak the contents of a "cust\_name" variable, which contains the customer's name as a character string. To have this name spoken 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.

Furthermore, each time you add a customer's name to the list, you would have to update your application, record the new name, recompile your application, transfer, and re-install it on the UCS 1000 R4.6 system. Obviously, this would involve a huge amount of work just to maintain the names in the system. In cases like this, it might make more sense to use TTS to speak the customer's name.

## Using WholeWord and FlexWord Speech Recognition Together

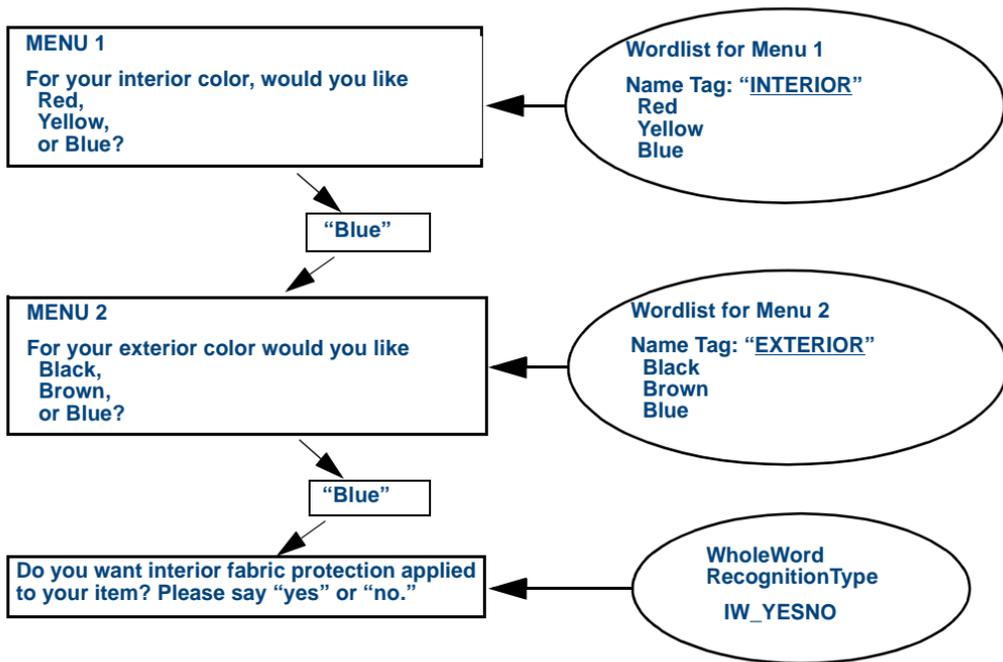
WholeWord and FlexWord speech recognition are used in either a TAS or irAPI script. See *UCS 1000 R4.6 Application Development with Advanced Methods*, 585-313-225 for TAS and irAPI information.

- If you want callers to say “yes” or “no,” a series of digits, or a single digit, choose a WholeWord speech recognition.
- If you want callers to speak a word or phrase from your custom vocabulary, choose a FlexWord speech recognition.

**Note:** Both recognizers cannot be used in the same prompt.

The example in [Figure 14 on page 218](#) shows a model of a retail application that combines FlexWord wordlists and the WholeWord “yes and no” speech recognition feature. In the example, the word “blue” must appear on both the “INTERIOR” wordlist and the “EXTERIOR” wordlist. Since the word “blue” appears on two wordlists, it counts as two words. If the word “blue” is not on the “INTERIOR” wordlist, this feature will not look to the “EXTERIOR” wordlist to find it. The feature returns the word in the wordlist that it thinks is the most suitable match. Since the feature can only look at one wordlist prompt, each wordlist must contain *all* possible choices for a single prompt.

Figure 14. Illustration of the Use of FlexWord and WholeWord Speech Recognition in a Single Application Script



## WholeWord Speech Recognition Versus FlexWord Speech Recognition Accuracy

For most applications, the accuracy of Whole Word recognition is higher than for FlexWord recognition. The following list shows the accuracy rates, arranged in order from highest to lowest:

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

## Comparison of Recognition Types

[Table 16](#) summarizes the similarities and differences between WholeWord speech recognition and FlexWord speech recognition when assigned to an LSPS II circuit card.

**Table 16. Comparison of Speech Recognition Types on an LSPS II Circuit Card**

<b>WholeWord Recognition</b>	<b>FlexWord Recognition</b>
Word-based	Phoneme-based
Requires data collection in model building for vocabulary words other than digits “zero” through “nine,” common synonyms for those words, and the words “yes” and “no”	No data collection required in model building
Connected digits	Single word or phrase
Standard and custom grammars	Customized wordlists
Barge-in supported	Barge-in supported
Phrase screening not supported	Phrase screening not supported
Limited vocabulary	Approximately 10,000 but can vary
Word spotting supported	Word spotting supported

# A Speech File Formats

## Overview

This appendix describes the encoding formats for speech files.

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 UCS 1000 R4.6 system supports the following encoding formats for speech phrases:

- PCM64 (pulse code modulation at 64 kbps in Mu-law encoding format)
- ADPCM32 (adaptive differential pulse code modulation at 32 kbps)
- ADPCM16 (adaptive differential pulse code modulation at 16 kbps — not commonly used in applications)
- CELP16 (code excited linear prediction at 16 kbps)
- OKI\_ADPCM32 (adaptive differential pulse code modulation at 32 kbps and used by the LSPS II circuit card)

### PCM Encoding Formats

In a PCM speech file, speech is sampled at 8000 times a second. Each sample is digitally coded into an 8-bit pattern (allowing 256 levels), resulting in 64,000 bits per second of speech. The header, which is used only by SSP and Quad circuit cards and has the values 0xAA 0xFF 0x34 0x00, is repeated at least 10 times for every second of speech (every 800 bytes). If the value 0xAA 0xFF occurs anywhere within the speech portion of the file, it is changed to 0xAA 0xFE. The speech data consists of contiguous 8-bit patterns representing sampled speech. One type of PCM format is available, Mu-law. In the Mu-law PCM format, PCM coding uses predefined quasi-logarithmic steps for speech levels, encoding more steps when the speech level is low.

**Note:** Speech that is recorded and/or played on an SSP or Quad circuit card is not transferable to an LSPS II circuit card, and vice versa.

### ADPCM Encoding Formats

ADPCM speech encoding is based on the principle that it is possible to reduce the amount of information needed to transmit speech between the sender and the receiver by using appropriate mathematical algorithms. The following types of ADPCM speech file formats are available, both of which follow similar compression algorithms:

- ADPCM32
- ADPCM16

Given that natural speech follows specific patterns (as opposed to random noise), the encoder (or the sender) predicts the present speech level by using a predefined algorithm and considering the past speech history. This predicted value is compared to the actual present speech level and the difference is encoded into digital format. The difference between the actual and the predicted values can be made as small as possible by using suitably defined predictor algorithms.

Thus, it is possible to encode the difference between the predicted and the actual speech into digital samples of either 4 bits (16 levels) or 2 bits (4 levels). At the decoder (receiver) end, the process is reversed. The decoder uses an equivalent algorithm to predict the present speech level from the past history, and makes the correction based on the received information from the sender to get the actual speech level. The accuracy and performance of the speech encoding and decoding depends on the type of algorithm used as well as the number of bits used for digital encoding of the difference between predicted and actual speech.

- In the ADPCM32 speech file, the header, which has the values `0xAA 0xFF 0x32 0x00`, is repeated at least five times for every second of speech data.
- In the ADPCM16 speech file, the header, which has the values `0xAA 0xFF 0x30 0x00`, is repeated at least five times for every second of speech data.

If the value `0xAA 0xFF` occurs anywhere within the speech portion of the file, it is changed to `0xAA 0xFE`. This makes a small but imperceptible change to the speech.

Processing of speech data for an application can be accomplished concurrently with application development. The only requirement is that the digitized speech must be loaded on the system before the application can be assembled.

## Speech Files

Speech files are comprised of two parts:

- A header section, used by the SSP or Quad circuit card, of the format shown in [Figure 15](#) that is present at the beginning of the file and repeated at periodic intervals within the body of the file
- Encoded digital data representing speech

**Figure 15. Speech File Header Section**

0xAA	0xFF	N	L
------	------	---	---

The following rules describe the header section that is used only by the SSP and Quad circuit cards:

- `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 17 on page 227](#).
- `L` (mandatory field) represents the length of a control field that follows the header bytes. The control field is optional. If it is absent, `L` is set to `0x00`. For the PCM and ADPCM type speech files, it is set to `0x00` (no control field is specified).

The following rules apply to the data section:

- Use an even number of bytes.
- Use no more than 500 bytes between headers.
- Use only CELP 16 for the pattern `0xAA 0xFF`.

The length field `L` identifies the number of words (2 bytes each) that follow that include user defined information.

**Note:** Coding and playback of CELP16 works properly only on the hardware the SSP and Quad circuit cards. If coding or playback is performed on an unsupported circuit card, the result is silence or noise.

- When CELP16 is played on a circuit card that does not support CELP16, the result is silence or noise, depending on the phrase.
- When a circuit card that does not accept CELP16 is asked to code CELP16, it either runs a default mode or stays in the current mode and the coding circuit card fails to return a phrase.

**Table 17. Identifier Codes in Speech Encoding**

<b>N</b>	<b>Value</b>	<b>Channel Capacity (Playback/Coding) Per SSP or Quad Circuit Card</b>
PCM Mu-law	0 x 34	120/120
ADPCM 32	0 x 32	120/120
ADPCM 16	0 x 30	120/120
CELP 16	0 x 40	120/60
		<b>Channel Capacity (Playback/coding) Per LSPS II Circuit Card</b>
PCM Mu-Law	N/A	64/64
OKI_ADPCM32	N/A	64/64

**Note:** Header bytes used by the SSP and Quad circuit cards 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.

# B Advanced Text-to-Speech Features

## Overview

For some Text-to-Speech (TTS) applications, you may need to customize the way TTS speaks to improve understandability or comprehension. Methods of customization include using escape sequences to do the following:

- Insert silent pauses or delays at the beginning, middle, or end of phrases
- Change the gender (male or female) of the synthesized voice
- Change the speaking rate of the synthesized voice
- Mark text as a member of a text class and specify a detector type that determines how certain words are spoken

**Note:** Standard TTS is available for US English only. However, TTS is available for other languages using the Proxy Text-to-Speech (PTTS) feature package, which allows the TTS to be processed by an external (proxy) server. For more details, see the *Proxy Text-to-Speech (PTTS) User Guide*, 585-350-115.

## Escape Sequences Used by the SSP Circuit Card

Customization of TTS involves using strings of characters called escape sequences that reset the TTS default parameters. The exact method of changing the defaults depends on the parameter and the application development tool. See *UCS 1000 R4.6 Application Development with Advanced Methods*, 585-313-225.

### Inserting Silent Delays and Pauses

Inclusion of silent delays or pauses in an application helps make the speech more natural sounding and may help increase listener comprehension. [Table 18 on page 230](#) lists the escape sequences used to insert TTS silences into synthesized speech.

Table 18. TTS Escape Sequence for Silence on the SSP

Sequence	Description
<code>\!sin</code> (initial)	<ul style="list-style-type: none"> <li>Specifies <math>n</math> centiseconds of silence at the beginning of the current text phrase, where <math>1 &lt; n &lt; 100</math></li> <li>Must be placed at the beginning of a text phrase and therefore cannot be used in the same phrase as a final silence (<code>\!sfn</code>)</li> </ul>
<code>\!sfn</code> (final)	<ul style="list-style-type: none"> <li>Specifies <math>n</math> centiseconds of silence at the beginning of the current text phrase, where <math>1 &lt; n &lt; 100</math></li> <li>Must be placed at the beginning of a text phrase and therefore cannot be used in the same phrase as a final silence (<code>\!sfn</code>)</li> </ul>
<code>\!(*[n])</code>	<p>Specifies <math>n</math> centiseconds of silence within the current phrase, where <math>1 &lt; n &lt; 60</math>. Can be placed anywhere within a text phrase and can be used in the same phrase with either an initial silence (<code>\!sin</code>) or a final silence (<code>\!sfn</code>).</p> <p><b>Note:</b> When typed in text or a variable, these sequences must be followed by one blank space.</p>

 **CAUTION:**

Using values outside the limits could cause problems in your application.

**Examples for  
Silence— Using  
Text**

The following examples illustrate use of escape sequences in text to add silent pauses and delays in an application.

- In the following example of a TAS script, silence of 0.5 seconds occurs before “Hello” is spoken. The delay helps secure the listeners’ attention.

```
Say ("Welcome")  
Say ("\\!si50 Hello!")  
Say ("Welcome to your Home Shopping Line!")
```

- In the following example of a TAS script, silence of 0.7 seconds is added after the phrase “Welcome to the Home Shopping Line!”. There is no silence between the word “Hello!” and the text that follows. The silence helps emphasize the end of the announcement.

```
Say ("Welcome")  
Say ("\\!si50 Hello!")  
Say ("\\!sf70 Welcome to your Home Shopping Line!")
```

- Note:** The end punctuation (exclamation point) after the first phrase (“Hello!”) is necessary for TTS to speak the 0.7-second silence at the end of the second phrase (“Welcome to your Home Shopping Line!”). Without end punctuation in the first phrase, this escape sequence would be ignored.

- In the following example of a TAS script, text phrase contains two silences— a 0.5-second silence at the beginning of the phrase and a 0.7-second silence between the words “Hello” and “and.” The placement of these silences helps make the synthesized speech more natural sounding.

```
Say ("Welcome")
Say (\!si50 Hello \!(*[70]) and welcome to your")
Say ("Home Shopping Line!")
```

### Examples for Silence— Using a Variable

- The following example from a TAS script shows a silence delay defined as a variable called Silence\_70 and used in an Announce action step. A silence of 0.7 seconds occurs between the first and second phrase.

```
Say (ch.Silence_70,
     im.\!si70")
Say (im."Welcome to your Home Shopping Line!")
Say (im."\b")
Say (ch.Silence_70)
Say (im."\b")
Say (im."To place an order, please have your")
Say (im."account number ready.")
```

## Changing the Gender of the Speaking Voice

TTS permits speech in both male and female voices. The default is male voice. Female voice provides the same capabilities and features as male voice. Voices can be switched by use of the escape sequences used by the SSP circuit card shown in [Table 19](#).

**Table 19. TTS Escape Sequences for Speech Gender Selection**

Sequence	Description
\!uSm	Male voice (default)
\!uSf	Female voice
	<b>Note:</b> When typed in text or a variable, these sequences must be followed by one blank space.

The voice selection stays in effect until an event occurs that removes speech inhibition. At that point, speech reverts to the default (male) voice. Therefore, to create an entire application in female voice, you must reset the voice repeatedly.

See *UCS 1000 R4.6 Application Development with Advanced Methods*, 585-313-225 for examples to change voice gender.

## Changing the Rate of Speech

Slowing the rate of speech for important words can provide emphasis and increase listener comprehension in an application. [Table 20](#) shows escape sequences used to control the rate of speech for TTS.

**Note:** The escape sequences in [Table 20](#) are used for TTS on the speech and signal processor (SSP) circuit card only.

**Table 20. TTS Escape Sequences for Speaking Rate for SSP Circuit Card**

Sequence	Description
\!rfastest \!rfast \!rnormal \!rslow \!rslowest	The default rate is “normal” (\!rnormal). Each rate is effective until the end of a sentence, indicated by punctuation such as a period, question mark, or exclamation mark. In text, to change the rate before the end of a sentence, insert a different escape sequence.
	<b>Note:</b> When typed in text or a variable, these sequences must be followed by one blank space.

## Setting Text Classes and Detectors

<b>Background</b>	In an application, you can designate the format of a field to be spoken by TTS.
<b>Classes</b>	<p>To solve this sort of problem, TTS allows you to explicitly assign text to any of the following four classes:</p> <ul style="list-style-type: none"><li>• Telephone number</li><li>• Proper name</li><li>• Address</li><li>• Fraction</li></ul>
<b>Class Detector Modes</b>	<p>For TTS to speak text assigned to these classes, you must use an escape sequence of the following class detector modes:</p> <ul style="list-style-type: none"><li>• Off</li><li>• Conservative mode (default)</li><li>• Risky mode</li></ul> <p><b>Note:</b> Initially all detectors are set to conservative mode.</p> <p>The detector mode determines how TTS speaks ambiguous text.</p>

- Risky mode sets a priority for the designated class to resolve ambiguity about how the information is to be classified. The setting of risky mode will be the highest priority at the time for the field to be spoken. TTS will then know, for example, that “22587” in the problem described above is part of an address.
- Conservative mode requires more contextual information to resolve ambiguous text.

[Table 21 on page 237](#) shows the escape sequences used for class detector modes.

### Guidelines

Observe the following guidelines when using class detectors:

- As a general rule, do not use risky mode unless in tests of your application TTS fails to pronounce words or phrases correctly or the text is known to be a member of the specified text class (for example, telephone numbers) or contains ambiguous abbreviations. Otherwise, always try conservative mode first (that is, do not explicitly set a class detector).
- More than one class detector can be activated at the same time, though results may be unpredictable.

- After using risky mode, reset detectors to conservative mode if special detection is not needed.
- Do not set a detector mode to off unless conservative mode is causing an undesirable effect.

**Table 21. TTS Escape Sequences for Class Detectors for the SSP Circuit Card**

<b>Sequence</b>	<b>Description</b>
Telephone number \!np0 (off) \!npr (risky mode) \!npc (conservative mode)	In conservative mode TTS speaks telephone numbers as individual digits, except for 800 in area code position, which is spoken “eight hundred”. For example, 555 1365 is spoken “five five five (pause) one three six five,” not “five million (pause) five hundred fifty-one thousand (pause) three hundred sixty-five.” Risky mode also ensures that “vanity” telephone numbers, such as 1 800 SHOPPER, are spelled out.

1 of 3

Table 21. TTS Escape Sequences for Class Detectors for the SSP Circuit Card

Sequence	Description
Proper name \!nno (off) \!nnr (risky mode) \!nnc (conservative mode)	<p>Risky mode is needed only when the data may contain name-related abbreviations or titles. Otherwise, use conservative mode. In risky mode, titles such as “Mr.” and ambiguous abbreviations such as “Dr.” are spoken correctly. Abbreviations such as “Sr.” (as in “John Doe Sr.”) and “Jr.” expand correctly in either risky or conservative mode because there is only one possible interpretation of the abbreviations.</p> <p><b>Note:</b> TTS speaks a proper name like “John Paul III” with three I’s. For TTS to pronounce this field successfully, you can enter either “John Paul third” or “John Paul 3rd”. This can be done in either conservative or risky mode because the “3rd” and “third” cannot be misinterpreted.</p>

2 of 3

**Table 21. TTS Escape Sequences for Class Detectors for the SSP Circuit Card**

Sequence	Description
Address \!nao (off) \!nar (risky mode) \!nac (conservative mode)	Use risky mode to detect partial addresses and to speak ambiguous address abbreviations correctly, such as “drive” for “Dr.” (not “doctor”) and “street” for “St.” not “saint”). Standard post office abbreviations apply.
Fraction \!nfo (off) \!nfr (risky mode) \!nfc (conservative mode)	Use risky mode so that, for example, “1/4” is spoken as “one fourth” and not “one slash four.” (To have TTS speak “1/4” as a date, use conservative date mode.)

**3 of 3**

## Escape Sequences Used by the LSPS II Circuit Card

The LSPS II circuit card has several types of embedded escape sequences such as:

- [Changing Text Class Detection Mode on page 240](#)
- [Switching to Special Translation Mode on page 242](#)
- [Changing Text Expansion Modes on page 243](#)
- [Operating on User Dictionaries on page 244](#)
- [Using Other Escape Sequences on page 244](#)

**Note:** The information in this section provides listings for some of the escape sequences that can be used by the LSPS II circuit card. For more information on escape sequencing, see the LSPS Software Development Kit resident on the system at [`/usr/lsp/doc/pdf/sdk.pdf`](#)

### Changing Text Class Detection Mode

**\incm** is the general escape sequence format where **c** is the text class detector and **m** is the modes to which the detector is set. [Table 22 on page 241](#) lists the function for the command variables.

Table 22. Text Class Detection Mode Variable Definitions

<b>Text Class Detector (c)</b>	<b>Function</b>
a	used for address class
d	used for date class
f	used for fraction class
i	used for measurement class
m	used for money class
n	used for proper names class
o	used for ordinal numbers class
p	used for telephone numbers class
t	used for time class
<b>Class Detector Setting (m)</b>	<b>Function</b>
o	turn the class detector off
c	set class detector to conservative mode (default)
r	set class detector to risky mode

**Note:** For example, to switch off the telephone numbers detector, enter **\!npo**

### Switching to Special Translation Mode

[Table 23](#) lists the escape sequences available to switch to special translation mode.

**Table 23. Switch to Special Translation Mode Escape Sequences**

Escape Sequence	Function
\!sb	spell mode begin
\!se	spell mode end
\!mb	math mode begin
\!me	math mode end
\!rb	raw mode begin
\!re	raw mode end
\!pb	proofread mode begin
\!pe	proofread mode end

**Changing Text  
Expansion Modes**

[Table 24](#) lists the escape sequences available to change text expansion modes.

**Table 24. Change Text Expansion Modes Escape Sequences**

<b>Escape Sequence</b>	<b>Function</b>
<code>\!ab</code>	begin acronym spell mode (default)
<code>\!ae</code>	end acronym spell mode
<code>\!eb</code>	begin abbreviation expansion mode (default)
<code>\!ee</code>	end abbreviation expansion mode
<code>\!hnv</code>	set numerical hyphen verbose mode (default)
<code>\!hnt</code>	set numerical hyphen terse mode
<code>\!hav</code>	set alphabetic hyphen verbose mode
<code>\!hate</code>	set alphabetic hyphen terse mode (default)

**Operating on User Dictionaries**

The cache file, residing at `/usr/lspstts/data` contains entries that are read in when an LSPS II circuit card is booted with a board type of TTS. The related sequences are the following:

- `\[ai/abbrev/expansion/]` - add **abbrev** to the abbreviation dictionary
- `\[ad/abbrev/]` - delete abbrev from the dictionary

**Using Other Escape Sequences**

[Table 25](#) lists remaining escape sequences and their functions.

**Table 25. Other Escape Sequences**

Escape Sequence	Function
<code>\!br</code>	force end of sentence regardless of punctuation
<code>\!cb</code>	switch to case significant mode (default)
<code>\!ce</code>	switch to case insignificant mode
<code>\!de</code>	set all text class detection modes and text expansion modes to default
<code>\!pn</code>	following word is marked as proper noun (may alter pronunciation)

1 of 2

Table 25. Other Escape Sequences

Escape Sequence	Function
<code>\!n</code>	set index marker <i>n</i> (0-32000) to signal application of when the marked text is actually spoken by the board
<code>\!t</code>	load external text class detection rules from the default file <i>ftspec.b</i>
<code>\!tfile</code>	load external text class detection rules from <i>file</i>

2 of 2

Illustrated in [Table 26 on page 246](#) are TTS recognition phonemes for the LSPS II circuit card in the US English language. A word accompanies each phoneme to illustrate its sound.

Table 26. US English Phonemes for the LSPS II Circuit Card

Phoneme	Word Example	Phoneme	Word Example
a	had	l	lid or hall*
@	father	m	moo or ham*
A	hay	n	nip or span*
^	cup	N	hang
W	how	o	hot
?	ohoh	O	hoe
Y	boy	p	pick or spit*
b	bit	r	rat or tar*
C	chip	s	sick
d	dip	S	ship
D	this	t	tip or stick*
E	heed	T	thick
R	her	U	rue
e	head	u	hook

1 of 2

Table 26. US English Phonemes for the LSPS II Circuit Card

Phoneme	Word Example	Phoneme	Word Example
f	fool	v	voice
g	got	w	we
h	hit	y	yes
l	hide	z	zap
i	hid	Z	measure
J	judge	~	little
k	kick or skip*	&	data
>	saw		

2 of 2

\* How used depends on the transcription.

## 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.

### **47B+D**

47 bearer (communication) and 1 data (signaling) channel on two T1 PRI circuit cards.

### **4ESS<sup>®</sup>**

A large Lucent central office switch used to route calls through the telephone network.

**5ESS®**

A Lucent electronic switching machine used to route calls through the telephone network or private branch exchange.

**A****AC**

alternating current

**ACD**

[automatic call distributor](#)

**AD**

application dispatch

**AD-API**

application dispatch application programming interface

**adaptive differential pulse code modulation**

A means of encoding analog voice signals into digital signals by adaptively predicting future encoded voice signals. This adaptive modulation method reduces the number of bits required to encode voice. See also [PTTS](#).

**adjunct products**

Products (for example, the Adjunct/Switch Application Interface) that the system administers via cut-through access to the inherent management capabilities of the product itself; this is in opposition to the ability of the system to administer the switch directly.

**ADPCM**

[adaptive differential pulse code modulation](#)

**ADU**

[asynchronous data unit](#)

**advanced speech recognition**

A speech recognition ability that allows the system to understand WholeWord and FlexWord™ inputs from callers.

**affiliate**

A business organization that Lucent controls or with which Lucent is in partnership.

**AGL**

application generation language

**ALERT**

System alerter process

**alerter**

A system process that responds to patterns of events logged by the “logdaemon” process.

**American National Standards Institute**

(ANSI) A private non-profit organization that develops and publishes voluntary standards for various industries in the United States.

**American Standard Code for Information Interchange**

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

**analog**

An analog signal, such as voice or music, that varies in a continuous manner. An analog signal may be contrasted with a digital signal, which represents only discrete states.

**ANI**

[automatic number identification](#)

**ANSI**

[American National Standards Institute](#)

**announcement**

A message the system plays to the caller to provide information. The caller is not asked to give a response. Compare to [prompt](#).

**API**

Application programming interface

**application**

The automated transaction (interactions) among the caller, the voice response system, and any databases or host computers required for your business.

**application administration**

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

**application verification**

A process in which the system verifies that all the components needed by an application are complete.

**ASCII**

[American Standard Code for Information Interchange](#)

**ASI**

analog switch integration

**ASR**

[advanced speech recognition](#)

**asynchronous communication**

A method of data transmission in which bits or characters are sent at irregular intervals and spaced by start and stop bits rather than by time. Compare to [synchronous communication](#).

**asynchronous data unit**

An electronic communications device that allows computer systems to communicate over asynchronous lines more than 50 feet (15 m) in length.

**automatic call distributor**

That part of a telephone system that recognizes and answers incoming calls and completes these calls based on a set of instructions contained in a database. The ACD can send the call to an operator or group of operators as soon as the operator has completed a previous call or after the system has played a message to the caller.

**automatic number identification**

A method of identifying the calling party by automatically receiving a string of digits that identifies the calling station of a particular customer.

**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 and FlexWord speech recognition and Dial Pulse Recognition (DPR) that allows callers to speak or enter their responses during the prompt and have those responses recognized (similar to the Speak with Interrupt capability). See also [echo cancellation](#).

**batch file**

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

**BB**

bulletin board

**blind transfer protocol**

A protocol in which a call is completed as soon as the extension is dialed, without having to wait to see if the telephone is busy or if the caller answered.

**bps**

bits per second

**BRDG**

call bridging process

**bridging**

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

A process that enables application designers to use information available within the system to classify the disposition of originated and transferred calls. CCA is an optional feature package.

**call data event**

A parameter that specifies a list of variables that are appended to a call data record at the end of each call.

**call data handler process**

A software process that accumulates generic call statistics and application events.

**called party number**

The number dialed by the person making a telephone call. Telephone switching equipment can use this number to selectively route an incoming call to a particular department or agent.

**caller**

The party who calls for a service, gets connected to the system, and interacts with it. As the system can also make outbound calls for service, the caller can also be the person who responds to those outbound calls.

**call flow**

See [transaction](#).

**call progress tones**

Standard telephony sounds that indicate the status of the call. These sounds include busy, fast busy, ringback, reorder, etc.

**card cage**

An area within a hardware platform that contains and secures all of the standard and optional circuit cards used in the system.

**cartridge tape drive**

A high-capacity data storage/retrieval device that can be used to transfer large amounts of information onto high-density magnetic cartridge tape based on a predetermined format. This tape can be removed from the system and stored as a backup, or used on another system.

**CAS**

channel associated signalling

**caution**

An admonishment or advisory statement used in the system documentation to alert the user to the possibility of a service interruption or a loss of data.

**CCA**

[call classification analysis](#)

**CDH**

[call data handler process](#)

**CELP**

[code excited linear prediction](#)

**central office**

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

**central processing unit**

See [processor](#).

**CGEN**

Voice system general message class

**channel**

See [port](#).

**channel associated signaling**

A type of signaling that can be used on E1 circuit cards. It occurs on channel 16.

**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.

**CO**

[central office](#)

**code excited linear prediction**

A means of encoding analog voice signals into digital signals that provides excellent quality with use of minimum disk space.

**command**

An instruction or request the user issues to the system software to make the system perform a particular function. An entire command consists of the command name and options.

**configuration**

The arrangement of the software and hardware of a computer system or network. The system configuration includes either a standard or custom processor, peripheral equipment (for example, printers and modems), and software applications. Configuration also refers to the way 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.

**controller circuit card**

A circuit card used on a computer system that controls its basic functionality and makes the system operational. These circuit cards are used to control magnetic peripherals, video monitors, and basic system communications.

**copying an application**

A utility in which information from a source application is directed into the destination application.

**coresidency**

The ability of two products or services to operate and interact with each other on a single hardware platform.

**CPE**

customer provided equipment or customer premise equipment

**CPN**

[called party number](#)

**CPT**

[call progress tones](#)

**CPU**

[central processing unit](#)

**CPU Complex**

The processor for the UCS 1000 R4.6 consisting of a single-board computing circuit card and an I/O companion board (SBC/IOB). The CPU complex is also used in other compactPCI platforms.

**crash**

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

**CSU**

channel service unit

**custom speech**

Unique words or phrases to be used in 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.

**CVS**

converse vector step

**D****danger**

An admonishment or advisory statement used in system documentation to alert the user to the possibility of personal injury or death.

**data interface process**

A software process that communicates with Script Builder applications.

**database**

A structured set of files, records, or tables.

**database field**

A field used to extract values from a local database and form the structure upon which a database is built.

**database record**

The information in a database for a person, product, event, etc. The database record is made up of individual fields for each information item.

**database table**

A structure, made up of columns and rows, that holds information in a database. Database tables provide a means of storing information that changes too often to “hard-code,” or store permanently, in the transaction outline.

**dB**

decibel

**DB**

database

**DBC**

database checking process

**DBMS**

database management system

**DC**

direct current

**DCE**

data communications equipment

**DCP**

digital communications protocol

**debug**

The process of locating and correcting errors in computer programs; also referred to as [troubleshooting](#).

**default**

The way a computer performs a task in the absence of other instructions.

**default owner**

The owner of a channel when no process takes ownership of that channel. The default owner holds all idle, in-service channels. In terms of the IRAPI, this is typically the Application Dispatch process.

**diagnose**

The process of performing diagnostics on a bus or on Tip/Ring, E1/T1, or SSP circuit cards.

**dial ahead**

The ability to collect and process touch-tone inputs in sequence, even when they are received before the prompts.

**dial pulse recognition**

A method of recognizing caller pulse inputs from a rotary telephone.

**dialed number identification service**

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

**dial through**

A capability provided by touch-tone and dial pulse recognition that allows callers to enter their responses during the prompt and have those responses recognized (similar to the Speak with Interrupt capability). See also [barge-in](#) and [echo cancellation](#).

**DIO**

disk input and output process

**DIP**

[data interface process](#)

**directory**

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

**DMA**

direct memory address

**DNIS**

[dialed number identification service](#)

**DPR**

[dial pulse recognition](#)

**DSP**

digital signal processor

**DTE**

data terminal equipment

**DTMF**

[dual tone multi-frequency](#)

**DTR**

data terminal ready

**dual tone multi-frequency**

A touch-tone sound that is an audio signal including two different frequencies. *DTMF feedback* is the process of the “switch” providing this information to the system. *DTMF muting* is the process of ignoring these tones (which might be simulated by human speech) when they are not needed for the application.

**dump space**

An area of the disk that is fixed in size and should equal the amount of RAM on the system. The operating system “dumps” an image of core memory when the system crashes. The dump can be fetched after rebooting to help in analyzing the cause of the crash.

**E****E&M**

[Ear and Mouth](#)

**E1 / T1**

Digital telephony interfaces, commonly called *trunks*. E1 is an international standard at 2.048 Mbps. T1 is a North American standard at 1.544 Mbps.

**Ear and Mouth**

A common T1 trunking protocol for connection between two “switches.”

**EBCDIC**

Extended Binary Coded Decimal Interexchange Code

**echo cancellation**

The process of making the channel quiet enough so that the system can hear and recognize WholeWord, FlexWord, 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.

**EIA**

Electronic Industries Association

**EISA**

Extended Industry Standard Architecture

**EMI**

electromagnetic interference

**Enhanced Basic Speech**

Pre-recorded speech available from Lucent Technologies in several languages. Sometimes called [standard speech](#).

**error message**

A message on the screen indicating that something is wrong with a possible suggestion of how to correct it.

**ESD**

electrostatic discharge

**ESS**

electronic switching system

**EST**

Enhanced Software Technologies, Inc.

**ET**

error tracker

**ETSI**

[European Telecommunications Standards Institute](#)

**Ethernet**

A name for a local area network that follows IEEE standard 802.3. Supported implementations are 10BaseT and/or 100BaseT.

## European Telecommunications Standards Institute

(ETSI) The European counterpart to ANSI ([American National Standards Institute](#)). Develops and publishes telecommunications standards.

### event

The notification given to an application when some condition occurs that is generally not encountered in normal operation.

### EXTA

external alarms feature message class

### external actions

Specific predefined system tasks that Script Builder can call or *invoke* to interact with other products or services. When an external action is invoked, the systems displays a form that provides choices in each field for the application developer to select. Examples are Call\_Bridge, Make\_Call, SP\_Allocate, SR\_Prompt, etc. In Voice@Work, external actions are treated as [external functions](#).

### external functions

Specific predefined (or customer-created) system tasks that Voice@Work or Script Builder can call or *invoke* to interact with other products or services. The function allows the application developer to enter the argument(s) for the function to act on. Examples are concat, getarg, length, substring, etc. See also [external actions](#).

**F****FCC**

Federal Communications Commission

**FDD**

floppy disk drive

**feature**

A function or capability of a product or an application within the system.

**feature package**

An optional package that may contain both hardware and software resources to provide additional functionality to a standard system.

**feature\_tst script package**

A standard system software program that allows a user to perform self-tests of critical hardware and software functionality.

**FEP**

front end processor

**field**

See [database field](#).

**FIFO**

first-in-first-out processing order

**file**

A collection of data treated as a basic unit of storage.

**file transfer**

An option that allows you to transfer files interactively or directly to and from UNIX using the file transfer system (FTS).

**filename**

Alphabetic characters used to identify a particular file.

**FlexWord speech recognition**

A type of speech recognition based on subword technology that recognizes phonemes or parts of words in a specific language. See also [subword technology](#).

**foos**

facility out-of-service state

**FTS**

file transfer process message class

**function key**

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

**G****GEN**

PRISM logger and alerter general message class

**grammar**

The inputs that a recognizer can match (identify) from a caller.

**GUI**

graphical user interface

**H****hard disk drive**

A high-capacity data storage/retrieval device that is located inside a computer platform. A hard disk drive stores data on nonremovable high-density magnetic media based on a predetermined format for retrieval by the system at a later date.

**hardware**

The physical components of a computer system. The central processing unit, disks, tape, and floppy drives, etc., are all hardware.

**hardware upgrade**

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

**HDD**

[hard disk drive](#)

**hwoos**

hardware out-of-service state

**Hz**

Hertz

**IBM**

International Business Machines

**iCk or ICK**

The system integrity checking process.

**ID**

identification

**IDE**

integrated disk electronics

**idle channel**

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

**IE**

information element

**IEEE**

Institute of Electrical and Electronic Engineers

**IND\$FILE**

The standard SNA file transfer utility that runs as an application under CICS, TSO, and CMS. IND\$FILE is independent of link-level protocols such as BISYNC and SDLC.

**independent software vendor**

A company that has an agreement with Lucent Technologies to develop software to work with the system to provide additional features required by customers.

**indexed table**

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

**industry standard architecture**

A PC bus standard that allows processors and other circuit cards to communicate with each other.

**INIT**

voice system initialization message class

**initialize**

To start up the system for the first time.

**inserv**

in-service state

**Integrated Services Digital Network**

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

**intelligent CCA**

Monitoring the line after dialing is complete to determine whether a busy, reorder (fast busy), or other failure has been encountered. 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 system, the interface is designed to provide you with easy access to the software capabilities.

**interrupt**

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

**Intuity Response Application Programming Interface**

A library of commands that provide a standard development interface for voice-telephony applications.

**IOB**

I/O companion card to the [SBC](#). This is part of the [CPU Complex](#).

**IPC**

interprocess communication

**IPC**

intelligent ports card (IPC-900)

**IRAPI**

[Intuity Response Application Programming Interface](#)

**IRQ**

interrupt request

**ISA**

[industry standard architecture](#)

**ISDN**

[Integrated Services Digital Network](#)

**ISV**

[independent software vendor](#)

**ITAC**

International Technical Assistance Center

**K****Kbps**

kilobytes per second

**Kbyte**

kilobyte

**keyboard mapping**

In emulation mode, this feature enables the keyboard to send 3270 keyboard codes to the host according to a configuration table set up during installation.

**keyword spotting**

A capability provided by WholeWord speech recognition that allows the system to recognize a single word in the middle of an entire phrase spoken by a caller in response to a prompt.

**L****LAN**

[local area network](#)

**LDB**

[local database](#)

**LED**

light-emitting diode

**library states**

The state information about channel activities maintained by the IRAPI.

**LIFO**

last-in-first-out processing order

**line side E1**

A digital method of interfacing a system to a PBX or “switch” using E1-related hardware and software.

**line side T1**

A digital method of interfacing a system to a PBX or “switch” using T1-related hardware and software.

**listfile**

An ASCII catalog that lists the contents of one or more talkfiles. Each application script is typically associated with a separate listfile. The listfile maps speech phrase strings used by application scripts into speech phrase numbers.

**local area network**

A data communications network in a limited geographical area. The LAN provides communications between computers and peripherals.

**local database**

A database residing on the system.

**LOG**

System logger process message class

**logical unit**

A type of SNA Network Addressable Unit.

**logdaemon**

A UNIX system information and error logging process.

**logger**

See [logdaemon](#).

**logging on/off**

Entering or exiting the system software.

**LSE1**

[line side E1](#)

**LSPS II**

[Lucent speech processing solutions II circuit card \(6UB5\)](#)

**LST1**

[line side T1](#)

**LU**

[logical unit](#)

**Lucent speech processing solutions II circuit card (6UB5)**

A high-performance speech processing circuit card capable of simultaneous support for various speech technologies. In addition to the basic speech-processing features, The LSPS II circuit card provides enhanced Text-to-Speech capabilities and subword recognition for large vocabularies.

**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 system screen from which you are able to enter either the System Administration or Voice System Administration menu.

**maintenance process**

A software process that runs temporary diagnostics and maintains the state of circuit cards and channels.

**manoo**

manually out-of-service state

**masked event**

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

**master**

A circuit card that provides clock information to the TDM bus.

**Mbps**

megabits per second

**MByte**

[megabyte](#)

**megabyte**

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

**menu**

Options presented to a user on a computer screen or with voice prompts.

**MF**

[multifrequency](#)

**MHz**

megahertz

**ms**

millisecond

**msec**

millisecond

**MS-DOS**

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

**MTC**

[maintenance process](#)

**multifrequency**

Dual tone digit signalling (similar to DTMF), used for trunk addressing between network switches or by network operators.

**multithreaded application**

A single process/application that controls several channels. Each thread of the application is managed explicitly. Typically this means state information for each thread is maintained and the state of the application on each channel is tracked.

**N****NCP**

Network Control Program

**NEBS**

Network Equipment Building Standards

**NEMA**

National Electrical Manufacturers Association

**netoos**

network out-of-service state

non-facility associated signalling

**NFS**

network file sharing

**NM-API**

Network Management - Application Programming Interface

**NMVT**

network management vector transport

**nonex**

nonexistent state

**nonindexed table**

A table that can be searched only in a sequential manner and not via a field name.

**nonmasked event**

An event that must be sent to the application. Generally, an event is nonmaskable if the application 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****OEM**

original equipment manufacturer

**on-line help**

Messages or information that appear on the user's screen when a function key (usually F1) is pressed.

**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****PBX**

[private branch exchange](#)

**PC**

personal computer

**PCB**

printed circuit board

**PCI**

[peripheral component interconnect](#)

**PCI Mezzanine Card**

A PCI module, such as a LAN or RAID controller, that connects to the [CPU Complex IOB](#) companion card.

**PCM**

[PTTS](#)

**PEC**

price element code

**peripheral (device)**

Equipment such as printers or terminals that is in addition to the basic processor.

**peripheral component interconnect**

A newer, higher speed PC bus that is gradually displacing ISA for many components.

**permanent process**

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

**phoneme**

A single basic sound of a particular spoken language. For example, the English language contains 40 phonemes that represent all basic sounds used with the language. The English word “one” can be represented with three phonemes, “w” - “uh” - “n.” Phonemes vary between languages because of guttural and nasal inflections and syllable constructs.

**phrase filtering (screening)**

The rejection of unrecognized speech. The WholeWord and FlexWord speech recognition packages can be programmed to reprompt the caller if the 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.

**PMC**

[PCI Mezzanine Card](#)

**poll**

A message sent from a central controller to an individual station on a multipoint network inviting that station to send if it has any traffic.

**polling**

A network arrangement whereby a central computer asks each remote location whether it wants to send information. This arrangement enables each user or remote data terminal to transmit and receive information on shared facilities.

**port**

A connection or link between two devices that allows information to travel to a desired location. See [telephone network connection](#).

**PRA**

[Primary Rate Access](#)

**Primary Rate Access**

The [ETSI](#) equivalent of ISDN PRI ([Primary Rate Interface](#)).

**PRI**[Primary Rate Interface](#)**Primary Rate Interface**

An ISDN term for connections over E1 or T1 facilities that are usually treated as trunks.

**private branch exchange**

A private switching system, either manual or automatic, usually serving an organization, such as a business or government agency, and usually located on the customer's premises.

**processor**

In system documentation, the computer on which UnixWare and 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 [ANSI](#).

**Proxy Text-to-Speech**

A form of Text-to-Speech (TTS) technology in which the task of converting the designated text to synthesized speech is handled by an external (proxy) computer, rather than on the SSP or Quad circuit card as standard TTS is. See also [Text-to-Speech](#).

**pseudo driver**

A driver that does not control any hardware.

**PSTN**

public switch telephone network

**PTTS**

[Proxy Text-to-Speech](#)

**pulse code modulation**

A digital modulation method of encoding voice signals into digital signals. See also [adaptive differential pulse code modulation](#).

**Q****Quad circuit card (CWB10, CWB11, or CWB12)**

A series of three digital network-interface circuit cards, each of which can be configured to support up to four E1 or T1 connections (or [spans](#)):

- Quad T1 (CWB10, with CYD10 rear I/O transition card, RJ48C connectors)
- Quad E1 (CWB11, with CYD10 rear I/O transition card, RJ48C connectors)
- Quad E1 (CWB12, with CYD12 rear I/O transition card, coaxial connectors)

**R****RAID**

redundant array of independent disks

**RAID Array**

An assembly of disk drives configured to provide some level of RAID functionality

**RAM**

random access memory

**RDMBS**

ORACLE relational database management system

**RECOG**

speech recognition feature message class

**recognition type**

The type of input the recognizer can understand. Available types include touch-tone, dial pulse, and Advanced Speech Recognition (ASR), which includes WholeWord and FlexWord speech recognition.

**recognizer**

The part of the system that compares caller input to a grammar in order to correctly match (identify) the caller input.

**record**

See [database record](#).

**recovery**

The process of using copies of the system software to reconstruct files that have been lost or damaged. See also [restore](#).

**remote database**

Information stored on a system other than your current system that can be accessed by your current system.

**REN**

ringer equivalence number

**reports administration**

The component of a system that provides access to system reports, including call classification, call data detail, call data summary, message log, and traffic reports.

**restore**

The process of recovering lost or damaged files by retrieving them from available back-up tapes or from another disk device. See also “recovery.”

**restore application**

A utility that replaces a damaged application or restores an older version of an application.

**reuse**

The concept of using a component from a source system in a target system after a software upgrade or platform migration.

**RFS**

remote file sharing

**RM**

resource manager

**roll back**

To cancel changes to a database since the point at which changes were last committed.

**rollback segment**

A portion of the database that records actions that should be undone under certain circumstances. Rollback segments are used to provide transaction rollback, read consistency, and recovery.

**RTS**

request to send

**S****SCA**

single connector architecture

**SBC**

A single-board computing circuit card used in the UCS 1000 R4.6. It is part of the CPU complex.

**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 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 system (see also [Voice@Work](#)).

**SCSI**

[small computer system interface](#)

**SDN**

software defined network

**shared database table**

A database table that is used in more than one application.

**shared speech**

Speech that is a part of more than one application.

**shared speech pools**

A parameter that allows the user of a voice application to share speech components with other applications.

**SID**

station identification

**single-threaded application**

An application that runs on a single voice channel.

**slave**

A circuit card that depends on the TDM bus for clock information.

**SLIP**

serial line interface protocol

**small computer system interface**

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

**SNA**

systems network architecture

**SNMP**

simple network management protocol

**software**

The set or sets of programs that instruct the computer hardware to perform a task or series of tasks — for example, UnixWare software and the 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).

**span**

A network-interface connection on an E1/T1 (CWB2), Quad T1(CWB10) or Quad E1 (CWB11 and CWB12) circuit card. Each Quad circuit card has from one to four spans (0, 1, 2, 3). An E1/T1 circuit card has one span (0).

**speech and signal processor circuit card (CWB1)**

A high-performance signal processing circuit card capable of simultaneous support for various speech technologies.

**speech energy**

The amount of energy in an audio signal. Literally translated, it is the output level of the sound in every phonetic utterance.

**speech envelope**

The linear representation of voltage on a line. It reflects the sound wave amplitude at different intervals of time. This envelope can be plotted on a graph to represent the oscillation of an audio signal between the positive and negative extremes.

**speech file**

A file containing an encoded speech phrase.

**speech filesystem**

A collection of several talkfiles. The filesystem is organized into 16-Kbyte blocks for efficient management and retrieval of talkfiles.

**speech modeling**

The process of creating WholeWord speech recognition algorithms by collecting thousands of different speech samples of a single word and comparing them all to obtain a statistical average of the word. This average is then used by a WholeWord speech recognition program to recognize a single spoken word.

**speech space**

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

**speech phrase**

A continuous speech segment encoded into a digital string.

**speech recognition**

The ability of the system to understand input from callers.

**SPIP**

signal processor interface process

**SPPLIB**

speech processing library

**SQL**

[structured query language](#)

**SR**

[speech recognition](#)

**SSP**

[speech and signal processor circuit card \(CWB1\)](#)

**standard speech**

The speech package available in several languages containing simple words and phrases produced by Lucent Technologies for use with the system. This package includes digits, numbers, days of the week, and months, each spoken with initial, medial, and falling inflection. The speech is in digitized files stored on the hard disk to be used in voice prompts and messages to the caller. This feature is also called Enhanced Basic Speech.

**standard vocabulary**

A standard package of simple word speech models provided by Lucent Technologies and used for WholeWord speech recognition. These phrases include the digits “zero” through “nine,” “yes,” “no,” and “oh,” or the equivalent words in a specific local language.

**string**

A contiguous sequence of characters treated as a unit. Strings are normally bounded by white spaces, tabs, or a character designated as a separator. A string value is a specified group of characters symbolized by a variable.

**structured query language**

A standard data programming language used with data storage and data query applications.

**subword technology**

A method of speech recognition used in FlexWord recognition that recognizes phonemes or parts of words. Compare to [WholeWord speech recognition](#).

**switch**

A software and hardware device that controls and directs voice and data traffic. A customer-based switch is known as a [private branch exchange](#).

**switch hook**

The device at the top of most telephones that is depressed when the handset is resting in the cradle (in other words, is *on hook*). The device is raised when the handset is picked up (in other words, when the telephone is *off hook*).

**switch hook flash**

A signaling technique in which the signal is originated by momentarily depressing the “switch hook.”

**switch interface administration**

The component of the system that enables you to define the interaction between the system and switches by allowing you to establish and modify switch interface parameters and protocol options for both analog and digital interfaces.

**switch network**

Two or more interconnected telephone switching systems.

**synchronous communication**

A method of data transmission in which bits or characters are sent at regular time intervals, rather than being spaced by start and stop bits. Compare to [asynchronous communication](#).

**SYS**

UNIX system calls message class

**sysgen**

system generation

**system administrator**

The person assigned the responsibility of monitoring all system software processing, performing daily system operations and preventive maintenance, and troubleshooting errors as required.

**system architecture**

The manner in which the system software is structured.

**system message**

An event or alarm generated by either the system or end-user process.

**system monitor**

A component of the 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.

**TAM**

[telecom alarm module](#)

**target system**

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

**TAS**

[transaction assembler script](#)

**TCP/IP**

transmission control protocol/internet protocol

**TDM**

time division multiplexing

**telecom alarm module**

An intelligent alarm module that provides critical, major, and minor alarm indicators.

**telephone network connection**

The point at which a telephone network connection terminates on a system. Supported telephone connections are Tip/Ring, T1, and E1.

**Text-to-Speech**

An optional feature that allows an application to play US English speech directly from ASCII text by converting that text to synthesized speech. The text can be used for prompts or for text retrieved from a database or host, and can be spoken in an application with prerecorded speech.

**ThickNet**

A 10-mm (10BASE5) coaxial cable used to provide interLAN communications.

**ThinNet**

A 5-mm (10BASE2) coaxial cable used to provide interLAN communications.

**time-division multiplex**

A method of serving a number of simultaneous channels over a common transmission path by assigning the transmission path sequentially to the channels, with each assignment being for a discrete time interval.

**Tip/Ring**

Analog telecommunications using four-wire media.

**token ring**

A ring type of local area network that allows any station in the network to communicate with any other station.

**trace**

A command that can be used to monitor the execution of a script.

**traffic**

The flow of information or messages through a communications network for voice, data, or audio services.

**transaction**

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

**transaction assembler script**

The computer program code that controls the application operating on the voice response system. The code can be produced from Voice@Work, Script Builder, or by writing directly in TAS code.

**transaction state machine process**

A multi-channel IRAPI application that runs applications controlled by TAS script code.

**transient process**

A process that is created dynamically only when needed.

**troubleshooting**

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

**TSO**

time share operation

**TSM**

[transaction state machine process](#)

**TTS**

[Text-to-Speech](#)

**TWIP**

T1 interface process

**U****UCS**

Unified Communications Server

**UK**

United Kingdom

**US**

United States of America

**UNIX Operating System**

A multiuser, multitasking computer operating system originally developed by Lucent Technologies.

**UNIX shell**

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

**upgrade scenario**

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

**usability**

A measurement of how easy an application is for callers to use. The measurement is made by making observations and by asking questions. An application should have high usability to be successful.

**USOC**

universal service ordering code

**UVL**

unified voice library

**V****VDC**

video display controller

**vi editor**

A screen editor used to create and change electronic files.

**virtual channel**

A channel that is not associated with an interface to the telephone network (Tip/Ring, T1, LSE1/LST1, or PRI). Virtual channels are intended to run “data-only” applications which do not interact with callers but may interact with DIPs. Voice or network functions (for example, coding or playing speech, call answer, origination, or transfer) will not work on a virtual channel. Virtual channel applications can be initiated only by a “virtual seizure” request to TSM from a DIP.

**vocabulary**

A collection of words that the system is able to recognize using either WholeWord or FlexWord speech recognition.

**vocabulary activation**

The set of active vocabularies that define the words and wordlists known to the FlexWord recognizer.

**vocabulary loading**

The process of copying the vocabulary from the system where it was developed and adding it to the target system.

**Voice@Work**

An optional software package that provides a graphical interface to assist in development of voice response applications on the system (see also [Script Builder](#)).

**voice channel**

A channel that is associated with an interface to the telephone network (T1, E1, or PRI). Any system application can run on a voice channel. Voice channel applications can be initiated by being assigned to particular voice channels or dialed numbers to handle incoming calls or by a “soft seizure” request to TSM from a DIP or the **soft\_szr** command.

**voice processing co-marketer**

A company licensed to purchase voice processing equipment to market and sell based on their own marketing strategies.

**voice response output process**

A software process that transfers digitized speech between system hardware (for example, Tip/Ring and SSP circuit cards) and data storage devices (for example, hard disk, etc.)

**voice response unit**

A computer connected to a telephone network that can play messages to callers, recognize caller inputs, access and update a databases, and transfer and monitor calls.

**voice system administration**

The means by which you are able to administer both voice-related aspects of the system.

**VPC**

[voice processing co-marketer](#)

**VROP**

voice response output process

**VRU**

[voice response unit](#)

**W****warning**

An admonishment or advisory statement used in system documentation to alert the user to the possibility of equipment damage.

**watchdog timer**

An timer that activates a [TAM](#) alarm when CPU activity is not received within the 30-second threshold.

**WholeWord speech recognition**

An optional feature package based on whole-word technology that can recognize the numbers one through zero, “yes”, and “no” (the key words). This feature is reliable, regardless of the individual speaker. This feature can identify the key words when spoken in phrases with other words. A string of key words, called *connected digits*,

can be recognized. During the prompt announcement, the caller can speak or use touch tones (or dial pulses, if available). See also [whole-word technology](#).

**whole-word technology**

The ability to recognize an entire word, rather than just the phoneme or a part of a word. Compare to “subword technology.”

**wink signal**

An interruption of current to a busy lamp indicating that there is a line on hold.

**word**

A unique utterance understood by the recognizer.

**wordlist**

A set of words available for FlexWord recognition by an application during a Prompt & Collect action step.

**word spotting**

The ability to search through extraneous speech during a recognition.

## A

### accuracy

DPR recognition [103](#)

FlexWord speech recognition [90–95](#)

LSPS WholeWord speech recognition  
[141–145](#)

WholeWord speech recognition [65–69](#)

### Acrobat Reader

adjusting the window size [xliv](#)

hiding and displaying bookmarks [xliv](#)

navigating [xliv](#)

printing from [xliv](#)

searching [xliv](#)

setting the default magnification [xliv](#)

### adaptive differential pulse code modulation (ADPCM)

ADPCM16 [223](#)

ADPCM32 [223](#)

### add command [5](#)

ADPCM, see adaptive differential pulse code  
modulation

allow interrupt, see barge-in

### applications

#### speech

confirmation steps [93](#)

developing [26–45](#)

features [7](#)

informative prompts [94](#)

installing [44–45](#)

reprompt steps [93](#)

## B

### barge-in [60](#), [136](#)

capability for TTS [115](#), [214](#)

encouraging and discouraging [60–61](#), [137](#)

not supported with DPR [102](#)

not supported with FlexWord speech  
recognition [80](#), [156](#)

response time compared with talkoff [62](#), [138](#)

response time for WholeWord speech  
recognition [61](#), [138](#)

see also talkoff

supported with LSPS WholeWord speech recognition [135](#)  
supported with WholeWord speech recognition [59](#)

## C

call flow, see transaction

CD-ROM documentation, printing [xliv](#)

CELP, see code excited linear prediction

code excited linear prediction (CELP) [222](#), [226](#)

connected digit speech recognition [55](#), [132](#)

conservative mode, see Text-to-Speech (TTS)

copy command [5](#)

custom

    vocabulary

        LSPS FlexWord speech recognition [155](#)

        WholeWord speech recognition [64](#), [140](#)

## D

DEFINITY switches

    limitations for DPR [104](#)

dial pulse recognition (DPR)

    accuracy [103](#)

    bargain-in not supported [102](#)

    capabilities [98](#)

    circuit cards [20](#)

    comparison with WholeWord and FlexWord  
    speech recognition [121](#)

    confirmation [101](#)

    description [18](#)

    digits [98](#)

    failure modes [106](#)

    functionality [102](#)

    processing [102](#)

    recognizer [102](#)

    requirements, hardware and software [18](#)

    switch limitations [104](#)

    training [103](#)

    troubleshooting [105](#)

    types [98](#)

    using to best advantage [114](#)

    using with other recognition types [98](#)

DIPs

    custom for LSPS WholeWord speech  
    recognition [140](#)

    custom for WholeWord speech recognition  
    [65](#)

DPR, see dial pulse recognition

drives

- hard disk
  - recovering a corrupted speech disk [6](#)

DTMF, see dual-tone multifrequency

dual tone multifrequency (DTMF)

- support for WholeWord speech recognition [62](#)
- touch tones [62](#), [138](#)

**E**

enabling or disabling [60](#), [136](#)

encoding formats, see speech

erase command [5](#)

escape sequences, see Text-to-Speech (TTS)

**F**

feature\_tat script [105](#)

FlexWord speech recognition

- accuracy [90](#), [120](#)
- compared to WholeWord speech recognition and DPR [121](#)
- description [17](#), [75](#), [150](#)
- designing an application [80](#)-??
- keyword spotting [79](#)
- languages supported ??-[77](#)
- phrase screening [79](#), [155](#)
- prompting for input [111](#)
- synonyms [111](#)
- use in applications [112](#)
- using with other recognition types [98](#), [116](#), [118](#)
- vocabulary [75](#), [150](#)
- word [75](#), [150](#)
- wordlists [75](#), [150](#)
  - activating when resident on the system [87](#)
  - moving between active and inactive directories [85](#)

FlexWord toolkit

- description [17](#)

## FlexWord wordlists

- copying to diskette for installation [85](#)
- deactivating [88](#)
- design guidelines [110](#)
- effective and ineffective [90–96](#)
- installing and activating from diskette [86](#)
- maximum number allowed [78](#)
- name tag characters [78](#)
- number of entries allowed [78](#)
- removing [89](#)
- size [78](#)
- using in scripts [81](#)
- word choice [91](#), [96](#), [111](#)

framing, see voice scripts

## I

## inflections

- recording speech with frame words [42](#)
- types [32](#)

## K

keyword spotting, see FlexWord speech recognition

keyword spotting, see LSPS FlexWord speech recognition

## L

## languages

## US English

- FlexWord phonemes [77](#)
- LSPS WholeWord recognition [127](#)
- WholeWord recognition [51](#)

licensing, TTS and PTTS [14](#)

list command [5](#)

In command [87](#)

## LSPS FlexWord speech recognition

- accuracy [219](#)
- compared to LSPS WholeWord speech recognition [220](#)
- designing an application [156–202](#)
- keyword spotting [155](#)
- prompting for input [211](#)
- synonyms [211](#)
- usage data files [202](#)
- use in applications [212](#)
- using with other recognition types [215](#), [217](#)

LSPS FlexWord wordlists  
  design guidelines [210](#)  
  word choice [211](#)

LSPS WholeWord speech recognition  
  accents, regional and national [146](#)  
  accuracy [141](#), [219](#)  
  barge-in [135](#), [137](#), [138](#)  
  compared to LSPS FlexWord speech  
    recognition [220](#)  
  confirmation [134](#)  
  custom  
    DIPs [141](#)  
    recognition types [139](#)  
  description [123](#)  
  dialects [146](#)  
  digits, fixed and variable length connected  
    [131–133](#)  
  DTMF support [138](#)  
  grammars [132](#)  
  languages [126](#)  
  models [129](#)  
  processing [124](#)–??  
  standard vocabulary [127](#)  
  types [128](#)  
  using with other recognition types [217](#)  
  word spotting [134](#)

LSPS, see Lucent Speech Processing Solutions  
  circuit cards

Lucent Speech Processing Solutions (LSPS)  
  circuit cards  
  description [22](#)  
  speech channel capacities [22](#)  
  TTS [23](#)

## M

Mu-law PCM format [223](#)

## N

natural language speech recognition (NLSR)  
  using with other recognition types [98](#)

## P

PCM, see pulse code modulation

phonemes  
  definition [151](#)  
  LSPS FlexWord speech recognition [151](#)

phrase screening [58](#), [79](#), [155](#)

prerecorded speech, using with TTS [117](#), [216](#)

prompts

for FlexWord speech recognition [111](#)

for LSPS FlexWord speech recognition [211](#)

informative [94](#)

structure and design [92](#)

Proxy Text-to-Speech [9](#)

Proxy Text-to-Speech (PTTS)

description [12](#)

languages supported [13](#)

licensing [14](#)

requirements, hardware and software [14](#)

see also Text-to-Speech (TTS)

PTTS, see Proxy Text-to-Speech (PTTS)

pulse code modulation (PCM) [222](#), [223](#)

## R

recognition confirmation

DPR [101](#)

WholeWord [58](#), [134](#)

restoring

speech files [5](#)

risky mode, see Text-to-Speech (TTS)

## S

## speech

- development methods [34](#)–??

- encoding formats

- overview [222](#)

- encoding phrases [43](#)

- features [7](#)

- files

- default directory [4](#), [7](#)

- formats [222](#)

- restoring [5](#)

- saving and restoring [5](#)

- storage space requirements [2](#)

- talkfiles [5](#), [6](#)

- filesystem

- structure [2](#), [7](#)

- inflections [32](#)

- installing ??–[45](#)

- phrases [25](#)

- processing [25](#)

- purchasing

- custom speech package [37](#)

- recording

- alphabetic characters [41](#)

- environmental conditions [40](#)

- equipment specifications [40](#)

- inflections [42](#)

- numbers [41](#)

- options [39](#)

- professional speakers [35](#), [39](#)

- recovering a corrupted speech disk [6](#)

- speech and signal processor (SSP) circuit cards

- description [19](#)

- DPR [19](#), [102](#)

- FlexWord speech recognition [18](#)

- speech channel capacities [19](#)

- TTS [20](#), [234](#)

- WholeWord speech recognition [50](#)

- speech recognition

- bilingual capabilities [50](#)

- comparison of types [121](#), [220](#)

- overview [14](#)

- see also DPR, FlexWord speech recognition,  
and WholeWord speech recognition

- SPEECHDIR variable [4](#)

- spreps command [5](#)

- spsav command [5](#)

- SSP, see speech and signal processor circuit  
cards

- standard vocabulary, see LSPS WholeWord  
speech recognition

- standard vocabulary, see WholeWord speech  
recognition

switches

central office limitations for DPR [104](#)

DEFINITY limitations for DPR [104](#)

synthesized speech, see Text-to-Speech (TTS)

## T

talkfiles, see speech

talkoff

DTMF support [62](#), [138](#)

response time compared with barge-in [62](#),  
[138](#)

see also barge-in

Text-to-Speech

language support for [9](#)

Text-to-Speech (TTS) [214](#)

conservative and risky mode [235](#)

customizing [38](#)

description [9](#), [37](#)

escape sequences [229](#)–??

functionality [38](#)

gender of speaking voice [233](#)

rate of speech [234](#)

requirements, hardware and software [11](#)

silent pauses [229](#)

text class detectors [235](#)

uses [11](#)

using with FlexWord speech recognition [115](#),  
[116](#), [214](#)

using with LSPS FlexWord speech  
recognition [215](#)

using with prerecorded speech [117](#), [216](#)

using with touch-tone input [214](#)

using with touchtone input [115](#)

using with WholeWord speech recognition  
[115](#)

Tip/Ring circuit cards

DPR [102](#)

transaction [27](#)

TTS, see Text-to-Speech

## U

updates to the product

<http://glsdocs.lucent.com> [xxix](#), [xlii](#)

## V

vdf command [5](#)

vocabulary

Flexword speech recognition [75](#), [150](#)

LSPS WholeWord speech recognition [123](#)

WholeWord speech recognition [15](#), [46](#), [47](#),  
[64](#), [140](#)

voice scripts

analyzing speech inflections [32](#)

description [28](#)

framing [41](#)

description [31](#)

example [31](#)

placing frame words [32](#)

selecting speech sounds [32](#)

using voiceless speech sounds [33](#)

planning [28](#)

types of phrases [29](#)

writing [30](#)

## W

WholeWord speech recognition

accents, regional and national [70](#)

accuracy [65](#), [120](#)

barge-in [59](#), [61](#)

bilingual recognition [50](#)

compared to FlexWord speech recognition  
and DPR [121](#)

confirmation [58](#)

custom

DIPs [65](#)

recognition types [63](#)

vocabulary [15](#), [64](#), [140](#)

description [15](#), [46](#), [47](#)

dialects [70](#)

digits, fixed and variable length connected  
[54-56](#)

DTMF support [62](#)

grammars [55](#)

languages [15](#), [49](#)

LSPS channel capacity [23](#)

models [52](#)

phrase screening [58](#), [134](#)

processing [47-49](#)

requirements, hardware and software [16](#)

SSP channel capacity [21](#)

standard vocabulary [15](#), [50](#)

- types [52](#)
- using with other recognition types [98](#), [118](#)
- word spotting [57](#)
- wl\_copy command [85](#)
- wl\_gen command [87](#), [89](#)
- wl\_install command [86](#)
- word spotting, see LSPS WholeWord speech recognition
- word spotting, see WholeWord speech recognition
- wordlists, see FlexWord wordlists