AVAYA

CONVERSANT® System

Version 8.0

Speech Development, Processing, and Recognition

585-313-218
Issue 3
Comcode 108850256
December 2001

Copyright and Legal Notices

Copyright

Acknowledgment

This document was prepared by Avaya. Offices are located in Denver CO, Columbus OH, Middletown NJ, and Basking Ridge NJ, USA.

Trademarks

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

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

- Santa Cruz Operation, Inc. — Registered trademarks: UnixWare.
- UNIX System Laboratories, Inc. — Registered trademarks: UNIX.
- Veritas Software Corporation — Trademarks: VERITAS.
- Xerox Corporation — Trademarks: Ethernet.

Limited Warranty

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

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

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

Product Safety Standards

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

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

United States FCC Compliance Information

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

Electromagnetic Compatibility (EMC) Standards

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

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

Canadian Interference Information

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

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

European Union Declaration of Conformity

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

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

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

Telecom New Zealand Ltd Warning Notices

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

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

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

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

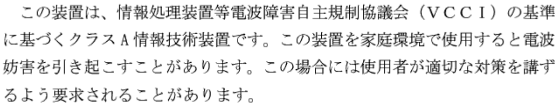
AUTOMATIC CALLS TO DIFFERENT NUMBERS: Some parameters required for compliance with Telecom’s Telepermit requirements are dependent on the equipment (PC) associated with this device. In order to operate within the limits for compliance with Telecom specifications, the associated equipment shall be set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

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

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

Japan

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



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

Preventing Toll Fraud

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

Avaya Fraud Intervention:

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

Providing Telecommunications Security

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

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

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

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

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

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

Your Responsibility for Your Company's Telecommunications Security

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

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

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

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

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

Documentation Ordering Information

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

Write, Call, or Fax

Avaya Publications Center

Voice	1 800 457-1235	International Voice	410-568-3680
FAX	1 800 457-1764	International FAX	410-891-0207

Write: Globalware Solutions
200 Ward Hill Avenue
Haverhill, MA 01835 USA

email: totalware@gwsmail.com

World Wide Web

Use a web browser to reach one of the following sites.

- Documentation:

  http://support.avaya.com/cgi-bin/gx.cgi/AppLogic+Elmo

- Updates and release notes for this documentation:

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

Standing Orders

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

Contents

Copyright and Legal Notices iii

About This Book xv

Overview xv
Intended Audience xv
How This Book Is Organized xv
How to Use This Book xvii
Conventions Used in This Book xvii
Safety and Security Alert Labels xxi
Getting Help xxi
Technical Assistance xxii
Related Resources xxii
Using the CD-ROM Documentation xxiii
How To Make Comments About This Book xxv

1 Overview of Speech 1

Overview 1
The Speech Filesystem 1
Speech Development Tools and Features 4
- Application Development Tools 5
- Text-to-Speech Features 5
- Speech Recognition Features 8

The SSP Circuit Card 13

2 Developing Speech 15

Overview 15
Determining the Transaction 16
Planning the Voice Script 16
Writing the Voice Script 18
Selecting a Speech Development Method 20
Recording Speech 25
Encoding Speech Phrases 28
Installing Speech 29

3 Editing Speech 31

Overview 31
Using Voice@Work for Editing Speech 31
Using Script Builder for Editing Speech 31

4 Recognizing WholeWord Speech Input 33

Overview 33
How WholeWord Speech Recognition Works 34
WholeWord Hardware and Software 35
Languages Supported 36
Bilingual Capabilities 36
Standard Vocabulary 36
Standard WholeWord Speech Recognition Types 37
Keyword Recognition Types 38
Connected-Digit Recognition Types 39
Recognition DIPs 41
Word Spotting 41
Phrase Screening 41
Recognition Confirmation 42
Barge-In 42
Dual Tone Multifrequency (DTMF) Support 44
Custom Grammars 45
Custom Vocabulary 45
Custom DIPs 45
WholeWord Speech Recognition Accuracy 46

5 Recognizing FlexWord Speech Input 51

Overview 51
Description 51
Languages Supported 52
FlexWord Capacity. 56
Designing a FlexWord Speech Recognition Application 57
Special FlexWord Recognition Accents 60
FlexWord Recognition Vocabulary Administration 61
FlexWord Speech Recognition Accuracy. 63

6 Using Natural Language Speech Recognition 67

Overview 67
Purpose 67
What Is Natural Language Speech Recognition? 67
The NLSR Offer 68

NLSR Offer Architecture 69
Minimum Configuration Model 69
Single-Client/Multiple-Server Model 71
Multiple-Client/Multiple-Server Model. 73
System Requirements 75
Software Requirements. 75
Hardware and LAN Requirements 76
Tasks Required to Use the NLSR Offer. 76
Configuring the Hardware and Software 77
Creating and Testing the Application 78
Deploying the Application. 79

7 Recognizing Dial Pulse Input 81

Overview 81
Hardware. 81
Simultaneous Dial Pulse Capabilities 81
Dial Pulse Recognition Types 82
Grammars 83
Recognition Confirmation 83
Barge-In. 83
How Dial Pulse Recognition Works. 84
Recognition Accuracy 84
DPR Troubleshooting 85

8 Putting It Together 87

Overview 87
Using NLSR with Other Features 87
Using WholeWord and Dial Pulse Recognition Together 88
Getting the Most Out of FlexWord Recognition. 89
Getting the Most out of NLSR 90
Getting the Most out of Dial Pulse Recognition. 91
Getting the Most out of Text-to-Speech. 91
Using FlexWord Recognition and Text-to-Speech Together 92
Using Text-to-Speech and Prerecorded Speech Together 92
Using WholeWord, FlexWord, and Natural Language Speech Recognition Together 93
WholeWord Speech Recognition Vs. FlexWord Recognition Accuracy 94
Comparison of Recognition Types 94

A Enhanced Basic Speech Formats 97

Overview 97
Languages Available 98
Background Information. 99
EBS Format Tables 102
Australian English EBS Formats 102
Brazilian Portuguese EBS Formats 104
Canadian French EBS Formats. 105
Cantonese Chinese Enhanced Basic Speech Formats. 108
Castilian Spanish Enhanced Basic Speech Formats 109
Czech Enhanced Basic Speech Formats 110
Dutch EBS Formats. 113
French EBS Formats 114
German EBS Formats. 116
Hindi EBS Formats 117
Hungarian EBS Formats 119
Indonesian Enhanced Basic Speech Formats 121
Italian Enhanced Basic Speech Formats 122
Japanese Enhanced Basic Speech Formats 123
Korean Enhanced Basic Speech Formats 124
Latin-American Spanish EBS Formats 125
Malay Enhanced Basic Speech Formats 127
Mandarin Chinese Enhanced Basic Speech Formats 128
Polish EBS Formats. 129
Slovak EBS Formats 132
Thai EBS Formats. 134
UK English EBS Formats 135
US English EBS Formats 137

B Speech File Formats 139

Overview 139
Speech File Formats 139
Speech Files 146

C Text-to-Speech Formats 149

Overview 149
US English TTS Formats. 150

D Advanced Text-to-Speech Features 153

Overview 153
Escape Sequences 153
Inserting Silent Delays and Pauses. 154
Changing the Gender of the Speaking Voice. 157
Changing the Rate of Speech 158
Setting Text Classes and Detectors. 160
Enabling Barge-In for Risky Mode. 166

E Recognition Post-Processing 167

Overview 167
Further Processing 167
Return Top 4 (RT4) 169
Prefix List Check (LSC) 170
Luhn Check (LNC). 170
Expected Value (EXV) 171
Close Match (CLM) 171
Further Processing Prefixes and Combinations 172

Glossary 173

Index 213

About This Book

Overview

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

This book offers detailed information about the features available for developing speech.

Intended Audience

The primary audience for this book includes the following:

- End customer developers — Responsible for creating and maintaining applications in the CONVERSANT system environment.
- Custom application developers — Responsible for creating applications to be used in the system environment for end-user customers. This segment includes Avaya custom application developers.
- Application distributors — Distribute and implement applications for end-users. This segment includes independent software vendors (ISVs) and voice processing co-marketers (VPCs).

How This Book Is Organized

This book is contains the following sections:

- Chapter 1, Overview of Speech — Provides an overview of speech on the CONVERSANT system, information about the speech file system and a discussion of the speech-related features available for voice response applications.
- Chapter 2, Developing Speech — Describes the procedures involved in producing speech, from determining the transaction and planning the script to recording and encoding the speech.
- Chapter 3, Editing Speech — Discusses features of the Voice@Work and Script Builder application development tools available for editing speech.

- Chapter 4, Recognizing WholeWord Speech Input — Explains the concepts behind WholeWord speech recognition and includes sections on creating WholeWord vocabularies and improving the accuracy of speech recognition.
- Chapter 5, Recognizing FlexWord Speech Input — Explains the concepts behind FlexWord speech recognition and its features.
- Chapter 6, Using Natural Language Speech Recognition — Explains the concepts behind the Natural Language Speech Recognition (NLSR) offer on CONVERSANT systems
- Chapter 7, Recognizing Dial Pulse Input — Discusses the recognition of dial pulses on rotary or push button telephones.
- Chapter 8, Putting It Together — Discusses how to make the various speech-related features and work well together in a voice response application.
- Appendix A, Enhanced Basic Speech Formats — Contains complete lists of Enhanced Basic Speech formats for all available languages.
- Appendix B, Speech File Formats — Includes information about speech files and speech file formats, including pulse code modulation (PCM) and adaptive differential pulse code modulation (ADPCM).
- Appendix C, Text-to-Speech Formats — Contains a complete list of the Text-to-Speech (TTS) formats (available for US English only).
- Appendix D, Advanced Text-to-Speech Features — Discusses how to customize synthesized speech by using escape sequences to add silence delays, change the speaking rate, select the speaking voice gender, and mark text as belonging to a specific text category.
- Appendix E, Recognition Post-Processing — Describes the data interface process (DIP) associated with WholeWord and FlexWord speech recognition and discusses five processing features that increase the system accuracy of recognition: Return Top 4, Prefix List Check, Luhn Check, Expected Value and Close Match.
- Glossary —Defines the terms, abbreviations, and acronyms used in CONVERSANT system documentation.
- Index — Alphabetically lists the principal subjects covered in the book.

How to Use This Book

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

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

This book is organized in a procedural manner in which you learn about speech on the CONVERSANT system (Chapter 1, Overview of Speech), followed by how to create speech (Chapter 2, Developing Speech) and edit speech (Chapter 3, Editing Speech).

Next, the focus turns to recognizing caller input and using speech recognition features (Chapter 4, Recognizing WholeWord Speech Input, Chapter 5, Recognizing FlexWord Speech Input, Chapter 6, Using Natural Language Speech Recognition, and Chapter 7, Recognizing Dial Pulse Input).

Finally, you learn how to effectively use speech-related features together (Chapter 8, Putting It Together).

Conventions Used in This Book

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

Terminology

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

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

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

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

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

Figure 1. Example of CONVERSANT Menu

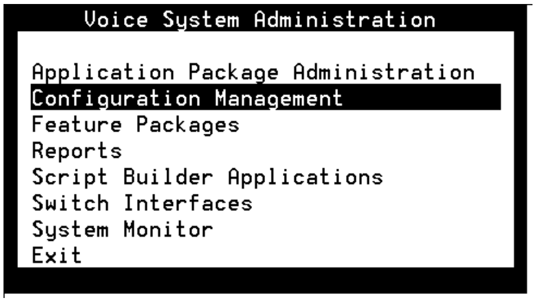
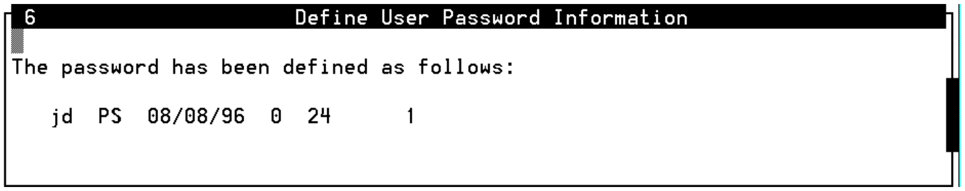

Voice System Administration
Application Package Administration
Configuration Management
Feature Packages
Reports
Script Builder Applications
Switch Interfaces
System Monitor
Exit

Figure 2. Example of CONVERSANT Window Showing Information

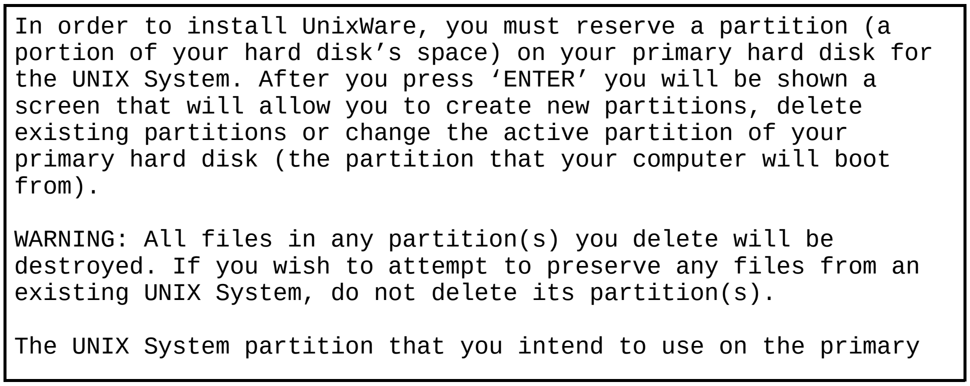


```
6                    Define User Password Information

The password has been defined as follows:

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

Figure 3. Example of CONVERSANT Screen Showing Information

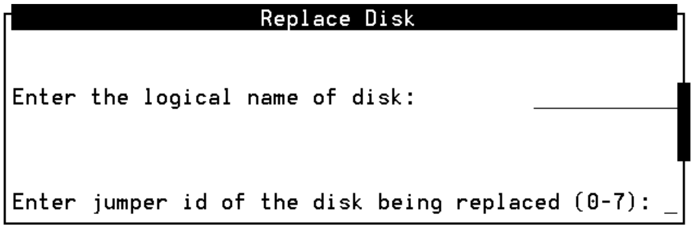


```
In order to install UnixWare, you must reserve a partition (a
portion of your hard disk’s space) on your primary hard disk for
the UNIX System. After you press ‘ENTER’ you will be shown a
screen that will allow you to create new partitions, delete
existing partitions or change the active partition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

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

The UNIX System partition that you intend to use on the primary
```

Figure 4. Example of CONVERSANT Window Requesting Information



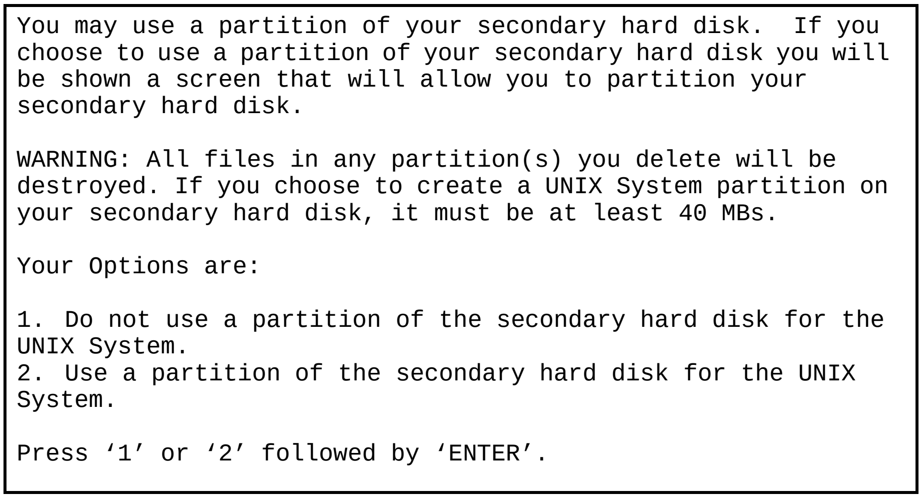
```
                    Replace Disk


Enter the logical name of disk:          __________



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

Figure 5. Example of CONVERSANT Screen Requesting Information



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

WARNING: All files in any partition(s) you delete will be
destroyed. If you choose to create a UNIX System partition on
your secondary hard disk, it must be at least 40 MBs.

Your Options are:

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

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

Keyboard and Telephone Keypad Representations

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

  Press **ENTER**.

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

  Press **ALT + D**.

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

  Press **F3** (Choices).

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

  Press **1** to record a message.

Cross References and Hypertext

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

Screen Displays

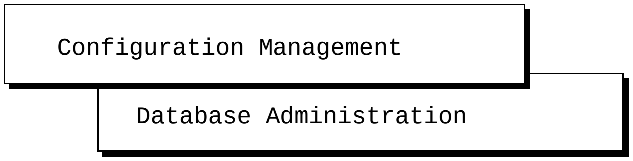
- Values, system messages, field names, prompts that appear on the screen, as well as simulated screen displays appear in typewriter-style `constant width` type, as shown in the following examples:

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

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

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

  Start at the CONVERSANT main menu and select:



```
Configuration Management
    Database Administration
```

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

Other Typography

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

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

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

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

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

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

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

Safety and Security Alert Labels

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

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

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

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

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

Getting Help

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

To use the online help:

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

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

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

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

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

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

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

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

Technical Assistance

Web Site

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

http://support.avaya.com

Contact Numbers

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

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

Related Resources

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

Training

To obtain training on the CONVERSANT product, contact the GLS Education and Training Center at one of the following numbers:

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

You can also view information on CONVERSANT training at the following web site: http://learning2.avaya.com

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

- For technicians doing repairs on CONVERSANT V8.0 systems
  - ~ BTE502H, CONVERSANT Installation and Maintenance
  - ~ BTE501W, CONVERSANT Administration for Technicians
- For technicians and administrators
  - ~ BTC344M, CONVERSANT V8 Administration Overview (CD-ROM)
- For application developers

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

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

ELMO Documentation

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

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

Additional Suggested Documentation

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

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

Obtaining Printed Versions of the Documentation

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

Using the CD-ROM Documentation

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

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

Setting the Default Magnification

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

Adjusting the Window Size

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

Hiding and Displaying Bookmarks

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

Using the Button Bar

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

Using Hypertext Links

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

Navigating with Double Arrow Keys

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

Searching for Topics

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

If you 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, do the following:

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

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

Comment Form

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

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

1 Overview of Speech

Overview

This chapter provides:

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

Topics covered include:

- The Speech Filesystem
- Speech Development Tools and Features
- The SSP Circuit Card

The Speech Filesystem

All speech to be played as part of an application resides as UnixWare files in a mounted UnixWare filesystem. By default, speech filesystems reside in **/voice1**, which is linked to **/home2**. These two filesystems can be used interchangeably. With the CONVERSANT 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 KB of space. Depending on the encoding rate, an 8-KB block holds different amounts of speech. See Table 1 to estimate how much speech is contained in each speech phrase block.

Table 1. Encoding Rates and Speech Phrase Blocks

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

For detailed information about coding rates, see Appendix B, Speech File Formats. For information about the encoding methods and capacities, see “Speech Storage Capacities” and “Speech Administration Capacities” in Chapter 4, “Features,” of *CONVERSANT System Version 8.0 System Description*, 585-313-219.

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-KB blocks, which allows for quick and efficient retrieval of speech files.

Specifying a Speech Directory

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

Procedure

1. Stop the voice system.

   See “Common System Procedures” in *CONVERSANT System Reference,* 585-313-205, for the procedure.

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

3. Add the following entry, where *directory* is the full path of the new directory where you want to store speech files:

   SPEECHDIR=*directory*

   The SPEECHDIR variable specifies the new directory.

4. Restart the voice system.

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

Saving and Restoring Speech Files

Speech files are backed up when a **mkimage** system backup is performed. The **spsav** (save) and **spres** (restore) commands are also available for backing up speech. See *CONVERSANT System Version 8.0 Administration*, 585-313-510, 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 disks during a recovery process with the **spres** command rather than from a **mkimage** tape, which takes longer and may not restore all speech.

Adding a Second Speech Disk

If you require speech-intensive applications, or if your system has 72 or more channels (telephone network connections), it is recommended that you add a second disk specifically for storing speech. This second disk in the system is used to store speech filesystems that reside in **/home3**.

For instructions to do so, see ”Adding a Hard Disk Drive” in “Replacing a Hard Disk Drive” in the CONVERSANT maintenance book for your platform.

Recovering from a Corrupted Speech Disk

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

Defining Phrases

A *phrase* is a unit of speech, such as a 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. Every phrase in a speech application script is identified by a phrase tag or phrase number. The application speaks a phrase to callers by referencing either the phrase tag or the phrase number in the application. See Defining Phrase Tags (on page 3) and Defining Phrase Numbers (on page 4) for more information.

Defining Phrase Tags

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

The two types of phrase tags are as follows:

- Enhanced Basic Speech (EBS):
  - Currency (such as dollars and cents)
  - Time — hours, minutes, and seconds on the 12-hour (a.m./p.m.) or 24-hour clock
  - Days of the week
  - Months
  - Numbers
  - Characters

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

Voice@Work and Script Builder use predefined EBS phrase tags for spoken output, such as digits and letters in various inflections. For more information about Voice@Work, see the EBS-related topics in the Voice@Work online help or Chapter 14, "Working with Variables" in *Using Voice@Work*, 585-313-207. For more information about Script Builder, see Chapter 9, “Speech Administration,” in *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217.

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

- Custom speech

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

Defining Phrase Numbers

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

Defining Talkfiles

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

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

Defining Speech Files

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

Defining the Speech Filesystem

A *speech filesystem* is a mounted UnixWare filesystem where speech resides and is defined in the **irAPI.rc** file. Only one speech filesystem can be active at a given time. For more information about CONVERSANT filesystems, see The Speech Filesystem (on page 1).

Speech Development Tools and Features

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

- Application Development Tools
  - ~ Voice@Work
  - ~ Script Builder
- Text-to-Speech Features
  - ~ Text-to-Speech (Standard)
  - ~ Proxy Text-to-Speech
- Speech Recognition Features
  - ~ WholeWord Speech Recognition
  - ~ FlexWord™ Speech Recognition
  - ~ Natural Language Speech Recognition (NLSR)
  - ~ Dial Pulse Recognition

Application Development Tools

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

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

For information about producing speech with Voice@Work or Script Builder, see Chapter 2, Developing Speech.

Voice@Work

Voice@Work applications are developed on a Windows-based PC and then installed on the CONVERSANT system.

For Voice@Work hardware requirements and instructions to install and use the feature software, see the Voice@Work online help or *Using Voice@Work*, 585-313-207.

Script Builder

Script Builder applications are developed through an interface accessed through the CONVERSANT main menu.

For instructions to install the Script Builder feature software, see “Installing the Script Builder Package” in “Installing the Optional Feature Software” in the CONVERSANT maintenance book for your platform. For detailed information on using Script Builder, see *CONVERSANT Version 8.0 Application Development with Script Builder*, 585-313-217.

Text-to-Speech Features

The CONVERSANT 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 (Text-to-Speech and Proxy Text-to-Speech) and a variety of speech recognition features that allow callers to speak in response to prompts (WholeWord, FlexWord, and Natural Language Speech Recognition).

Text-to-Speech (Standard)

The standard Text-to-Speech (TTS) feature allows you to include speech in an application by providing text as input and allowing the speech and signal processing (SSP) circuit card to produce synthesized speech. Currently TTS is available only for US English.

Note: Proxy Text-to-Speech (PTTS) provides similar capabilities both for US English and for other languages. For more information, see cref.

The default gender for the speaking voice is male, although a female voice can be specified.

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

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 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 nonalphabetic 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 to sound more natural 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 (for more information, see Appendix D, Advanced Text-to-Speech Features).

The 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. The 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 (standard) TTS feature requires:

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

Reference

For more information about using TTS, see the following:

- To use TTS in Voice@Work applications, see information in the appropriate online help topics or in *Using Voice@Work*, 585-313-207.
- To use TTS in Script Builder applications, see “Using Text-to-Speech” in Chapter 8, “Using Optional Features,” of *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217.
- To use TTS with TAS applications, see Appendix B, “Summary of TAS Script Instructions,” in *CONVERSANT System Version 8.0 Application Development with Advanced Methods*, 585-313-216.
- To install the TTS feature software, see “Installing the Text-to-Speech Package” in “Installing the Optional Feature Software” in the CONVERSANT maintenance book for your platform.
- For information on SSP circuit card functions, see The SSP Circuit Card (on page 13).

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 circuit card, text-to-speech processing must be done using one or more auxiliary computers connected to the CONVERSANT system in a client/server configuration. This capability is called Proxy Text-to-Speech (PTTS).

The current release of the PTTS feature supports two basic classes of languages:

- Japanese
- 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 CONVERSANT client system and on one or more customer-provided servers running the Windows NT 4.0 operating system. These computers communicate through socket interface connections, called PTTS connections, over an Ethernet Local Area Network (LAN). Multiple CONVERSANT clients and their associated PTTS servers can be placed on a single LAN.

The CONVERSANT 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 CONVERSANT 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. This means that if you already have TTS licenses, you can also use them for PTTS.

There are five external functions used to access PTTS functionality in Voice@Work or Script Builder applications.

Note: PTTS does not support the formats available for TTS on the speech and signal processor (SSP) circuit card for playback in Announce or Prompt and Collect nodes (Voice@Work) or action steps (Script Builder). These formats are listed in Appendix D, "Text-to-Speech Formats."

Requirements

The PTTS feature requires:

- The Proxy Text-to-Speech feature software package
- 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 CONVERSANT system feature that allows the system to recognize and respond to spoken voice responses from the caller. The system offers WholeWord, FlexWord, and Natural Language speech recognition.

Previous system releases offered speech technologies on a per-system basis. Releases starting with Version 6.0 offer speech technologies on a per-channel basis. The SSP circuit cards include basic playback and coding functions for WholeWord speech recognition and FlexWord speech recognition. All these types of speech recognition are licensed on a per-channel basis.

WholeWord Speech Recognition

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

WholeWord speech recognition is available in the following languages:

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

Standard Versus Custom Vocabulary

Avaya 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 or create 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 packages
- An SSP circuit card with WholeWord functionality assigned

Reference

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

- To install the WholeWord feature software, see “Installing the WholeWord Recognition Packages” in “Installing Optional Feature System Software” in the CONVERSANT maintenance book for your platform.
- For information on:
  - ~ Using the WholeWord feature in Voice@Work applications, see the WholeWord speech recognition topics in the online help or Chapter 21, "The Application Resources Tool" in *Using Voice@Work*, 585-313-207
  - ~ Using the WholeWord feature in Script Builder applications, see “Using WholeWord Speech Recognition” in Chapter 8, “Using Optional Features,” in *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217
  - ~ SSP circuit card functions, see The SSP Circuit Card (on page 13)

FlexWord Speech Recognition

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

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

Because FlexWord recognition provides a cost-effective way of designing large, customized vocabularies and menu options, it is ideal for word-intensive or phrase-intensive applications. For example, a name dialer, 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:

- The FlexWord speech recognition base package and one language package
- An SSP circuit card with FlexWord functionality assigned

Reference

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 the CONVERSANT maintenance book for your platform.
- To use the FlexWord feature in Voice@Work applications, see the FlexWord topics in the Voice@Work online help or Chapter 15 "Working with FlexWord Speech Recognition" in *Using Voice@Work*, 585-313-207.
- To use the FlexWord feature in Script Builder applications, see “Using FlexWord Speech Recognition” in Chapter 8, “Using Optional Features,” in *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217.
- To configure the video controller circuit card, see “Setting Up the Monitor” in “Installing Base System Software” in the CONVERSANT maintenance book for your platform.
- For information on SSP circuit card functions, see The SSP Circuit Card (on page 13).

Natural Language Speech Recognition

While WholeWord and FlexWord speech recognition can recognize specific words or phrases, even when extraneous words or phrases are thrown in by the caller, they have no ability to recognize what part the recognized speech plays in the overall statement. In other words, these technologies are all designed to *recognize* specific words or phrases.

Natural Language Speech Recognition (NLSR) takes the speech recognition process further by providing a more natural conversational interface with IVR systems. NLSR can be used to recognize particular words and phrases and to interpret and assign meaning to the speech it recognizes.

For example, under the more basic forms of speech recognition, a caller can respond only to specific prompts, such as “Say ‘one’ if you want information about...” or “Say ‘yes’ if this is correct.” NLSR enables you to write applications that ask the caller more open-ended questions, such as a banking application that presents the caller with a list of options and then asks “What would you like to do?” Then, when the caller responds “I’d like to know the balance of my checking account, please,” the system can recognize what kind of information the caller is asking for (the *balance* in a *checking account*) and can automatically direct the call to a new prompt that asks for the caller’s checking account number. This new technology provides a more natural way of interacting with callers.

It is worth noting that NLSR is also able to take into account grammatical structures. This allows it, for instance, to recognize and deal appropriately with differences in statements like the following caller responses:

> “I would like to fly from Chicago to LAX.”
>
> “I need to get from LAX to Chicago.”

NLSR is also capable of understanding natural numbers (“seventy-six” instead of “seven six”), natural dates (“July 26th” instead of “zero seven two six”) and natural currency (“25 dollars” instead of “two five zero zero”).

Because of the relatively complex nature of NLSR, it requires the use of larger vocabularies and grammars. For this reason, it works best when a “proxy server” is used to do the speech recognition. Speech recognizers running on external speech recognition servers have at their disposal more CPU power and more memory than specialized voice processing circuit cards such as the SSP circuit card on the CONVERSANT system. They also offer more flexibility by allowing you to use multiple speech recognition servers, thus increasing the available CPU power and memory even more.

The NLSR offer assumes you are using one or more CONVERSANT systems in conjunction with one or more NLSR computers in a client/server configuration. This setup assumes that the CONVERSANT is the client and the NLSR computer is the server. Beyond that, the exact NLSR system architecture is dependent on what other components you are using in conjunction with this offer.

All computers in the system communicate using an Ethernet PCI local area network (LAN) connection.

Requirements

NLSR requires:

- The NLSR Proxy/Client base software package
- Voice@Work (Release 3.2 or higher) or Script Builder application development software
- The appropriate LAN and telephony circuit cards and connections
- An SSP circuit card
- One or more external NLSR servers with appropriate speech recognition engine software installed

Reference

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

- To install and use NLSR, see *Natural Language Speech Recognition with the Intuity™ CONVERSANT® System*, 585-310-774.
- To use NLSR in Voice@Work applications, see the NLSR-related topics in the Voice@Work online help or in *Using Voice@Work*, 585-313-207.
- For information on SSP circuit card functions, see The SSP Circuit Card (on page 13).

Dial Pulse Recognition

Dial pulse recognition (DPR) allows users with rotary telephones or push-button telephones that generate dial pulses to respond to the CONVERSANT system. DPR converts the “pops” and “clicks” on the line to dial pulses. DPR supports the digits 0 to 9 on analog and digital interfaces. For more information on DPR, see Chapter 7, Recognizing Dial Pulse Input.

Requirements

DPR requires:

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

Reference

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 the CONVERSANT maintenance book for your platform.
- To use the DPR feature in Voice@Work applications, see the DPR-related topics in the Voice@Work online help or in *Using Voice@Work*, 585-313-207.
- For information on SSP circuit card functions, see The SSP Circuit Card (on page 13).

The SSP Circuit Card

Description

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

Capacity

Table 2 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 “Speech and Signal Processor Card” in Chapter 2, “Hardware,” in *CONVERSANT System Version 8.0 System Description*, 585-313-219.

Table 2. SSP Circuit Card Channel Capacities

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

Reference

- To install an SSP circuit card, see “Installing or Replacing Circuit Cards” in the CONVERSANT maintenance book for your platform.
- To assign SSP functionality to an SSP circuit card, see “Assign SP/SSP Functions” in Chapter 3, “Voice System Administration” in *CONVERSANT System Version 8.0 Administration*, 585-313-510.

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 signal processor (SSP) circuit card. The circuit card actually plays the speech.

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

- Determining the Transaction (on page 16)
- Planning the Voice Script (on page 16)
- Writing the Voice Script (on page 18)
- Selecting a Speech Development Method (on page 20)
- Recording Speech (on page 25)
- Encoding Speech Phrases (on page 28)
- Installing Speech (on page 29)

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.

Reference

For information on:

- Planning a voice response application, see *CONVERSANT System Application Design Guidelines*, 585-310-670 in the language translation of your choice.
- Developing an application using Voice@Work, see the topics for "Creating an Application" in the Voice@Work online help, or in Chapter 3, "Creating an Application in Voice@Work" in *Using Voice@Work*, 585-313-207.
- Developing an application using Script Builder, see *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217.
- Developing an application using Response Application Programming Interface (IRAPI), the Transaction State Machine (TSM) process script level language, or C language, see *CONVERSANT System Version 8.0 Application Development with Advanced Methods,* 585-313-216.

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. (Note that this is done for you automatically in Voice@Work applications.)
- 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 speaker 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. Also, 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 of your users.
- Use the following types of phrases in your voice script:
  - ~ Long phrases that stand alone, for example,

    “Welcome to the *XYZ* 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* your account number.” (phrase at the beginning of the sentence)

  “You must *enter* your account number.” (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 18).
- 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 to be recorded by 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 are examples from 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.

> To learn more about our investment opportunities, press the “star key.”
>
> “This amount represents” the total balance.
>
> “Please enter” two oh one.
>
> You have “a balance of” two hundred dollars.
>
> You can deposit “up to” five hundred dollars.

Analyzing Speech Inflections

Three types of inflection exist with speech phrases:

- Rising inflection

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

- Medial inflection

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

- Falling inflection

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

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

Placing Frame Words

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

Installing “and” verifying
Cutting “and” pasting

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

Note: The word “and” is part of the Avaya Enhanced Basic Speech package.

Selecting Speech Sounds for Framing

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

December “eighth”

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

August “eighth”

Including Voiceless Speech Sounds

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

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

Selecting a Speech Development Method

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

The following options require you to record speech:

- Hiring a Professional Speaker (on page 20)
- Purchasing an Avaya Custom Speech Package (on page 21)
- Producing Self-Recorded Custom Speech (on page 21) with either the Voice@Work or the Script Builder development tool

The following options do not require you to record speech:

- Purchasing the EBS (Enhanced Basic Speech) Package (on page 22)
- Using TTS (Text-to-Speech) (on page 22)
- Sharing Speech (on page 23) already recorded in another application
- Importing Speech (on page 24) from another application

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 16) and Writing the Voice Script (on page 18) 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 the appropriate inflections.
- Use the same speaker for all speech associated with a specific application.

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

See Recording Speech (on page 25) for information on setting up a recording session with a professional speaker.

Purchasing an Avaya Custom Speech Package

You can purchase a professionally recorded custom speech package from Avaya. You write out the script and Avaya 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 Avaya,” is a custom speech phrase.

An advantage of purchasing a custom speech package from Avaya 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: For more information about purchasing a custom speech package from Avaya, see Recording Speech (on page 25).

Producing Self-Recorded Custom Speech

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

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

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

Reference

For detailed information on recording speech with:

- Voice@Work, see the "Phrase Editor" and "Audio Recording Tips" topics in the Voice@Work online help or in Chapter 18, "Working with Phrases" in *Using Voice@Work*, 585-313-207.
- Script Builder, see “The Speech Administration Window” in Chapter 9, “Speech Administration,” of *CONVERSANT System 7.0 Application Development with Script Builder*, 585-313-217

Purchasing the EBS (Enhanced Basic Speech) Package

You can purchase the professionally recorded EBS (Enhanced Basic Speech) package from Avaya. The EBS package provides the following:

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

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

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 *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, and so on).
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 help make the outcome sound more natural. TTS also filters the text, which is critical because it:

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

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

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

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

For information on using TTS in a:

- Voice@Work application, see the TTS and related topics in the Voice@Work online help or in *Using Voice@Work,* 585-313-507.
- Script Builder application, see “Using Text-to-Speech” in Chapter 8, “Using Optional Features,” in *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217.
- TAS application, see Appendix B, “Summary of TAS Script Instructions,” in *CONVERSANT System Version 8.0 Application Development with Advanced Methods*, 585-313-216.

Sharing Speech

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

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

For information sharing speech in a:

- Voice@Work application, see the topics for sharing application resources in the Voice@Work online help or in *Using Voice@Work*, 585-313-207
- Script Builder application, see Chapter 9, “Speech Administration,” in *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217

Importing Speech

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

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

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

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

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

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

> “Thank you, please call Bank America again.”

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

For information on:

- Importing speech into one Voice@Work application from another, see the chapter on phrases in *Using Voice@Work*, 585-313-207
- Importing speech into one Voice@Work application from another, see "The Phrase Editor" topic in the Voice@Work online help or in Chapter 18, "Working with Phrases", in *Using Voice@Work*, 585-313-207.
- Importing speech into one Script Builder application from another, see Chapter 9, “Speech Administration,” in *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217.
- Editing speech, see Chapter 3, Editing Speech, in this book.

Recording Speech

Several methods are available to obtain recorded speech for an application. These methods include:

- Working with a Professional Speaker (on page 25)
- Working with Avaya to Develop a Custom Speech Package (on page 27)
- Recording Speech Independently (on page 28) with the Voice@Work or Script Builder application development tool or the GSE

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 25)
- Equipment Specifications (on page 25)
- Recording Natural Sounding Speech (on page 26)
- Recording Sets of Related Words (on page 26)
- Recording Alphabetic and Numeric Characters with Inflections (on page 27)

Environmental Conditions

A studio specifically designed for recording sessions is necessary when using a professional speaker. Arrange for the recording environment to be noise-free and acoustically “neutral.” The room should have walls that are covered with drapes or carpeting, and carpeted floors and be environmentally controlled.

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 magnetic 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). For tips to record alphabetic and numeric characters, see Recording Alphabetic and Numeric Characters with Inflections (on page 27).

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 four 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 framing sentence in a typical context. A framing sentence for the days of the week might be as follows:

> The movie for "[*day of the week*]" is ____.

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

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

In addition to the speaker, the studio manager and a coordinator are usually present at the recording session. The customer for whom the speech is recorded might 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 framing words that separate instances of initial, medial, and rising inflections for each letter and number.

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

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

Working with Avaya to Develop a Custom Speech Package

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

- A list of phrase tags that need to be recorded
  - ~ For information on **.pl** files in Voice@Work, you can print the contents of the **.pl** (phrase list) file to get a list of phrase tags. See "The Code Generation Tool" topic in the Voice@Work online help or Chapter 23, "The Code Generation Tool" in *Using Voice@Work,* 585-313-207.
  - ~ In Script Builder, you can use the List function to print all the phrase tags in an application. Unrecorded phrases are marked with an asterisk. See “Displaying Phrase Tags” in Chapter 9, “Speech Administration,” of *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217.
- A list of the complete text of all the phrases that need to be recorded

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

  - ~ In Voice@Work, you can use the Print Reports function to print the text of all the phrases in an application. For information on selecting phrases to print, see "The File Menu" topic in the Voice@Work online help or Chapter 5, "The File Menu" in *Using Voice@Work*, 585-313-207.
  - ~ In Script Builder, you cannot print the phrase text, only the phrase tags.
- A clear indication of the phrase tags that corresponds to each phrase
- Specifications of what type of speech encoding should be used

Note: The choice of formats depends in part on the amount of speech necessary for the applications on the CONVERSANT system. Some coding rates require more storage space than others. For a comparison of the storage requirements for various coding rates, see Table 1 on page 1 in Chapter 1, Overview of Speech.

- 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 CONVERSANT system at the following telephone number: 614-860-2260

Recording Speech Independently

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

Reference

For information on recording speech using:

- The default audio application installed on the Voice@Work PC, see the "Phrase Editor" and "Audio Recording Tips" topics in the Voice@Work online help or in Chapter 18, "Working with Phrases" in *Using Voice@Work*, 585-313-207.
- Script Builder on the CONVERSANT system, see Chapter 9, “Speech Administration,” in *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217.

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.

The following are ways to digitize speech:

- Use the Voice@Work application development tool.

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

- Use the Script Builder application development tool.

  When you record speech using Script Builder, it is digitized automatically in ADPCM32 format. See Chapter 9, “Speech Administration,” in *CONVERSANT System Version 8.0 Application Development with Script Builder,* 585-313-217 for more information about recording speech with Script Builder.

- Use Avaya.

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

Installing Speech

After the speech is digitized, it must be installed onto the CONVERSANT system.

Installing Speech in a Voice@Work Application

Voice@Work applications must be transferred from the Voice@Work PC to the CONVERSANT system before they can be installed and assigned to channels. All these operations can be done from within Voice@Work.

- To transfer a Voice@Work application, see "The Application Transfer Tool" topics in the Voice@Work online help or Chapter 24, "The Application Transfer Tool" in *Using Voice@Work,* 585-313-207.
- To install and assign a Voice@Work application, see "Using the SCI Tools" in the Voice@Work online help or "Installing the Application" in Appendix A, "Using the SCI Tools" in *Using Voice@Work,* 585-313-207.

Installing Speech in a Script Builder or TSM Application

To install a Script Builder or TSM application on the CONVERSANT system, see “Verifying and Installing the Application” in Chapter 10 “Application Administration,” of *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217. See also “Restoring Speech” in Chapter 9, “Speech Administration,” in the same book.

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 *CONVERSANT System Version 8.0 Application Development with Advanced Methods*, 585-313-216.

3 Editing Speech

Overview

This chapter describes how to use tools to edit speech files. Topics include:

- Using Voice@Work for Editing Speech (on page 31)
- Using Script Builder for Editing Speech (on page 31)

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

Using Voice@Work for Editing Speech

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

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

Using Script Builder for Editing Speech

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

For detailed information about editing speech files with Script Builder, see “Editing Speech” in Chapter 9, “Speech Administration,” of *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217.

4 Recognizing WholeWord Speech Input

Overview

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 include:

- How WholeWord Speech Recognition Works (on page 34)
- WholeWord Hardware and Software (on page 35)
- Languages Supported (on page 36)
- Bilingual Capabilities (on page 36)
- Standard Vocabulary (on page 36)
- Standard WholeWord Speech Recognition Types (on page 37)
- Keyword Recognition Types (on page 38)
- Connected-Digit Recognition Types (on page 39)
- Recognition DIPs (on page 41)
- Word Spotting (on page 41)
- Phrase Screening (on page 41)
- Recognition Confirmation (on page 42)
- Barge-In (on page 42)
- Dual Tone Multifrequency (DTMF) Support (on page 44)
- Custom Grammars (on page 45)
- Custom Vocabulary (on page 45)
- Custom DIPs (on page 45)
- WholeWord Speech Recognition Accuracy (on page 46)

How WholeWord Speech Recognition Works

Overview

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

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

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

Caller Response

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

The Recognizer

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

- For Voice@Work applications in the `Save Response In` field in the Prompt and Collect, Automenu, and Menu nodes (see the chapter on node descriptions in *Using Voice@Work*, 585-313-207, for more information)
- For Script Builder applications in the `Caller Input Field` on page 2 of the Define Prompt & Collect screen (for more information, see Chapter 8, “Using Optional Features,” in *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217)

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

Computational Processing

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

The SSP circuit card uses only 19 watts per circuit card. For more information on the SSP channel capacities, see Capacity (on page 13) in Chapter 1, Overview of Speech.

Further Processing

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

WholeWord Hardware and Software

- For a list of the required software and hardware, see WholeWord Speech Recognition (on page 9) in Chapter 1, Overview of Speech.
- For the procedure to install the WholeWord software, see “Installing the WholeWord Recognition Package” in ”Installing Optional Feature System Software” in the CONVERSANT maintenance book for your platform.

Languages Supported

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

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

For additional information about global support, contact your Avaya 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 3 on page 37 lists the standard vocabulary supported for the various languages in WholeWord speech recognition.

Table 3. Standard Vocabulary for WholeWord Speech Recognition

Caller Input Field	Australian, UK, and US, English	Canadian French and French	Castilian and Latin-American Spanish	Japanese	German	Dutch	Brazilian Portuguese	Italian
0	zero, oh	zéro	cero	zero	null	nul	zero	zero
1	one	un, une	uno	ichi	eins	een	um	uno
2	two	deux	dos	ni	zwei, zwo	twee	dois	duo
3	three	trois	tres	san	drei	drie	três	tre
4	four	quatre	cuatro	shi, yon	vier	vier	quatro	quattro
5	five	cinq	cinco	go	fünf	vijf	cinco	cinque
6	six	six	seis	roku	sechs	zes	meia, seis	sei
7	seven	sept	siete	shichi, nana	sieben	zeven	sete	sette
8	eight	huit	ocho	hachi	acht	acht	oito	otto
9	nine	neuf	nueve	ku, kyu	neun	negen	nove	novo
yes	yes	oui	sí	hai	ja	ja	sim	sì

Standard WholeWord Speech Recognition Types

Description

A recognition type allows 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 38) for more information)
- A grammar (see Grammars (on page 38) for more information)
- Specifications of maximum and minimum numbers of digits (see Connected-Digit Recognition (on page 40) and Fixed-Length Versus Variable-Length Connected Digits (on page 40) for more information)
- A recognition data interface process (DIP) — optional (see Recognition DIPs (on page 41) for more information)

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

- In Voice@Work applications, through the Response tab of the Prompt and Collect, Announce, Menu, Automenu and Voice Capture nodes (for more information, see the chapters on working with nodes in *Using Voice@Work*, 585-313-207).
- In Script Builder applications in the `Recog:` field on Page 2 of the Define Prompt & Collect screen (for more information, see "Using Dial Pulse Recognition or Speech Recognition" in Chapter 8, "Using Optional Features," of *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217).

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

Table 4. Standard WholeWord Speech Recognition Types

Language	"Yes" or "No"	One digit 1_3	One digit 1_3 or "no"	One digit 1_5	One digit 1_5 or "no"	One digit 0_9 or "oh"
Australian English	AU_YN	AU_1_3	AU_1_3N	AU_1_5	AU_1_5N	AU_DIG
Brazilian Portuguese	BP_YN	BP_1_3	BP_1_3N	BP_1_5	BP_1_5N	BP_DIG
Canadian French	CF_YN	CF_1_3	CF_1_3N	CF_1_5	CF_1_5N	CF_DIG
Castilian Spanish	CS_YN	CS_1_3	CS_1_3N	CS_1_5	CS_1_5N	CS_DIG
Dutch	NL_YN	NL_1_3	NL_1_3N	NL_1_5	NL_1_5N	NL_DIG
French	FR_YN	FR_1_3	FR_1_3N	FR_1_5	FR_1_5N	FR_DIG
German	GR_YN	GR_1_3	GR_1_3N	GR_1_5	GR_1_5N	GR_DIG
Italian	IT_YN	IT_1_3	IT_1_3N	IT_1_5	IT_1_5N	IT_DIG

1 of 2

Table 4. Standard WholeWord Speech Recognition Types

Language	"Yes" or "No"	One digit 1_3	One digit 1_3 or "no"	One digit 1_5	One digit 1_5 or "no"	One digit 0_9 or "oh"
Japanese	JN_YN	JN_1_3	JN_1_3N	JN_1_5	JN_1_5N	JN_DIG
Latin-American Spanish	MS_YN	MS_1_3	MS_1_3N	MS_1_5	MS_1_5N	MS_DIG
UK English	UK_YN	UK_1_3	UK_1_3N	UK_1_5	UK_1_5N	UK_DIG
US English	US_YN	US_1_3	US_1_3N	US_1_5	US_1_5N	US_DIG

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 10 digits, or it can be of a variable length. Grammars are provided for the 1-digit to 10-digit fixed-length and the 1-digit to 24-digit variable-length strings.Table 5 lists the connected-digit recognition types.

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

Table 5. WholeWord Speech Connected-Digit Recognition Types

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

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

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

- In Voice@Work applications, through the `Minimum Length` and `Maximum Length` fields in the Response tab of the Menu, Prompt and Collect, or Automenu node (for more information, see the appropriate node topics in the Voice@Work online help or Chapter 31, "Node Descriptions" in *Using Voice@Work*, 585-313-207).
- In Script Builder applications in the `Min Number of Digits:` and `Max Number of Digits:` fields on Page 2 of the Define Prompt & Collect screen (see Chapter 7, “Defining the Transaction,” in *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217, for more information).

Connected-Digit Recognition

Connected digits are strings of naturally spoken digits, which may or may not include a pause. All packages recognize spoken digit strings. Connected-digit recognition packages allow you to collect input by specifying a fixed-length of 1 to 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 accuracy 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 5 on page 39). 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 45) for more information). Using the provided variable-length grammar as an alternative results in decreased accuracy. If your application must accept a variable-length digit string, you may be able to increase recognition accuracy by using a two-step entry process. First prompt the caller for the number of digits, and then prompt for the digits. For example:

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

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

Recognition DIPs

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

For some recognition types, there may be tradeoffs that determine how much of the structure of the input is built into the grammar and how much should be left for the DIP. For more information about custom DIPs, see "Writing the DIP" in Chapter 4, "Data Interface Process," in *CONVERSANT System Version 8.0 Application Development with Advanced Methods*, 585-313-216. Also see Custom DIPs (on page 45). For more information on post-processing, see Appendix E, Recognition Post-Processing.

Word Spotting

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

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

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

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.

For more information regarding the recognition confirmation:

- In Voice@Work applications, see information about the `Confirmation Input Mode` field in the Confirmation tab of the Menu, Prompt and Collect, Automenu, or Voice Capture node in the Voice@Work online help or Chapter 31, “Node Descriptions” in *Using Voice@Work*, 585-313-207.
- In Script Builder applications, see “Specifying the Confirm Action” in Chapter 7, “Defining the Transaction,” in *CONVERSANT System Version 8.0 Application Development with Script Builder,* 585-313-217.

Barge-In

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 44).

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

Note: The system is able to detect touchtones immediately. However, it does take a few seconds longer to detect valid speech, as opposed to a cough, sneeze, and so on. 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 must place an **SR_Prompt** instruction to enable barge-in. If a barge-in resource is available, it is reserved for this call. Otherwise the return code to **SR_Prompt** is negative, and barge-in is not available. For information about using **SR_Prompt** in:

- *Voice@Work* applications, see the “Standard External Functions” topic in the Voice@Work online help or Appendix C, “Standard External Functions” in *Using Voice@Work*, 585-313-207.
- Script Builder applications, see “Defining the SR_Prompt External Action” in Chapter 8, “Using Optional Features,” of *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217.

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

- Voice@Work applications is done by setting the `Allow Interrupt` field to “true” in the Response tab of the Menu node, the Prompt and Collect node, or the Automenu node (for more information, see the appropriate node topic in the Voice@Work online help or Chapter 31, “Node Descriptions” in *Using Voice@Work*, 585-313-207)
- Script Builder applications is done by setting the `Speak with Interrupt` field set “yes” on page 1 of the Define Prompt & Collect screen (see Chapter 8, “Using Optional Features,” in *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217, for more information)

If your application does allow barge-in, be sure your prompts are worded so that callers know they cannot speak until the prompt is finished. For example, look at the wording and the <pause> length of the prompts in Discouraging Barge-In (on page 43) below. There are no large gaps of silence to tempt callers to speak barge in.

Discouraging Barge-In

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

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

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

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

If there is silence at the end of a prompt, the caller might speak but the system might 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 examples 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 43), so callers are encouraged to respond after hearing the desired option and the word “now.”

> “For sales say, ‘one’ now.” (pause of 1 to 1.5 seconds)
> “For service, say, ‘two’ now.” (pause of 1 to 1.5 second)
> “For a representative, say ‘three’ now.” (pause of 1 to 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 touchtones 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 might be present simultaneously. 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 touchtone 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 42) above for information comparing and contrasting talkoff and barge-in.

Custom Grammars

Recognition types are the choices associated with:

- The Input ASR Mode field of the Response tab on Prompt and Collect, Menu, and Automenu nodes in Voice@Work
- The `Recog:` field on the Define Prompt & Collect screen in Script Builder

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 Avaya representative.

Custom Vocabulary

Avaya 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 Avaya, 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 5, Recognizing FlexWord Speech Input.

For more information, contact your Avaya 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. For more information on improving accuracy of an application, see Return Top 4 (RT4) (on page 169) in Appendix E, Recognition Post-Processing.

Reprompt

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

  Placing the action required at the end of the prompt helps eliminate the possibility that the caller might forget what is required while listening to 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 number again, say ‘yes’. Otherwise, say ‘no’.”

  To encourage the use of barge-in, add a pause of 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 that is 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 that are 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.

Note: The Natural Number Speech Recognition package, available from Avaya, does allows you to accept and use more natural caller responses, such as "December 15th" instead of "1-2-1-5" or "twenty-two dollars and thirty-seven cents" instead of "2-2-3-7". You can also use the Natural Language Speech Recognition offer to get this same kind of flexibility (for more information, see Chapter 6, Using Natural Language Speech Recognition).

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 for longer strings.
- Fixed-length strings are more accurate than variable length strings 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 Avaya 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 touchtone entries, which require fewer hardware and software resources
- The number of simultaneous speech recognition calls expected
- The use of barge-in with WholeWord speech recognition, which increases the hardware and software resources required to process each transaction

5 Recognizing FlexWord Speech Input

Overview

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

- Description (on page 51)
- Languages Supported (on page 52)
- FlexWord Capacity (on page 56)
- Vocabularies (on page 56)
- Designing a FlexWord Speech Recognition Application (on page 57)
- Special FlexWord Recognition Accents (on page 60)
- FlexWord Recognition Vocabulary Administration (on page 61)
- FlexWord Speech Recognition Accuracy (on page 63)

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.

Languages Supported

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.

Illustrated in Figure 6 on page 53 through Figure 11 on page 55 are FlexWord speech recognition phonemes for the supported languages. A word accompanies each phoneme to illustrate its sound.

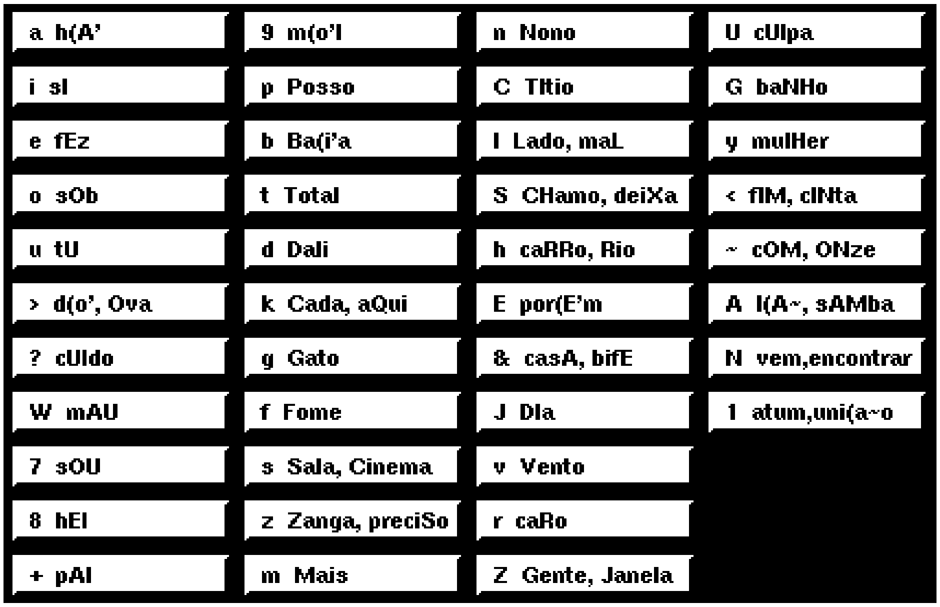
FlexWord speech recognition supports the following languages.

- Brazilian Portuguese (Figure 6 on page 53)
- French (Figure 7 on page 53)
- German (Figure 8 on page 54)
- Japanese (Figure 9 on page 54)
- Latin-American Spanish (Figure 10 on page 55)
- US English (Figure 11 on page 55

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

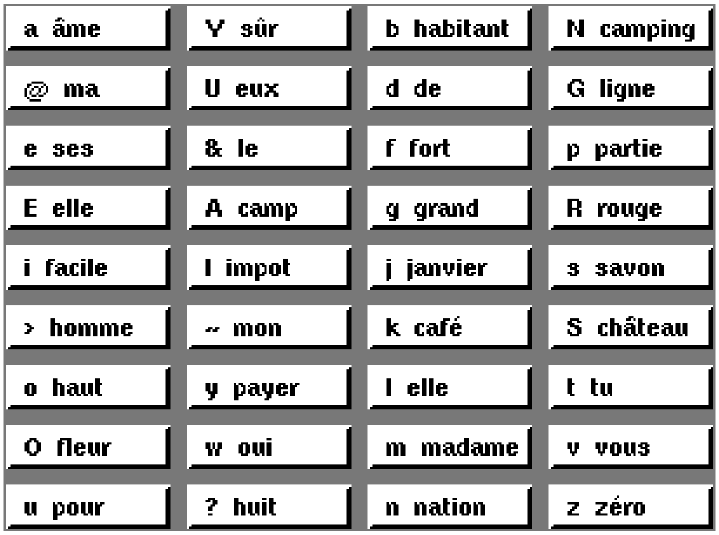
For CONVERSANT system Version 8.0, the speech and signal processor (SSP) circuit card is required for FlexWord speech recognition for all languages.

Figure 6. Brazilian Portuguese Phonemes



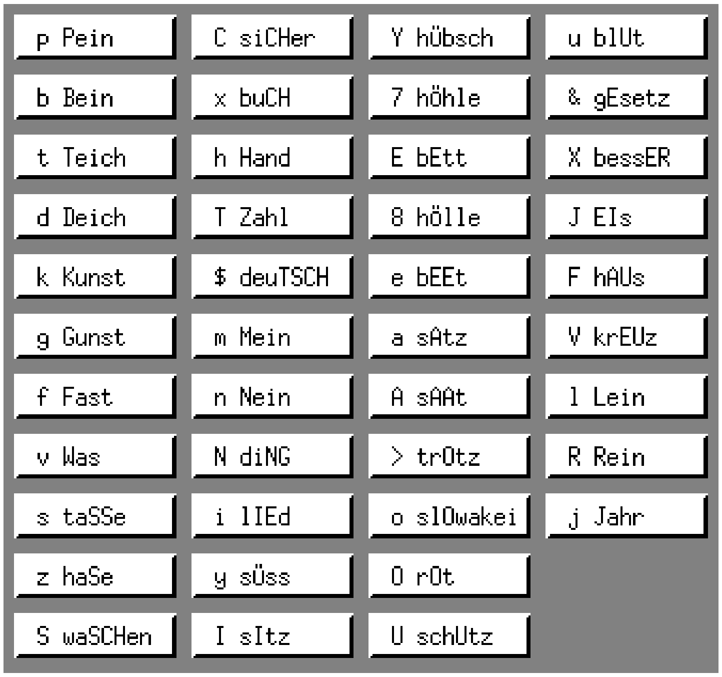
a h(A’	9 m(o’l	n Nono	U cUlpa
i sI	p Posso	C TItio	G baNHo
e fEz	b Ba(i’a	l Lado, maL	y mulHer
o sOb	t Total	S CHamo, deiXa	< fIM, cINta
u tU	d Dali	h caRRo, Rio	~ cOM, ONze
> d(o’, Ova	k Cada, aQui	E por(E’m	A l(A~, sAMba
? cUIdo	g Gato	& casA, bifE	N vem,encontrar
W mAU	f Fome	J DIa	1 atum,uni(a~o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

Figure 7. French Phonemes



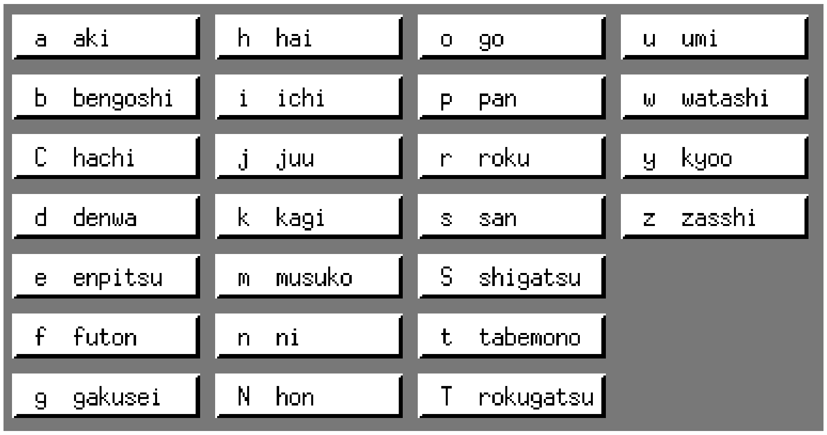
a âme	Y sûr	b habitant	N camping
@ ma	U eux	d de	G ligne
e ses	& le	f fort	p partie
E elle	A camp	g grand	R rouge
i facile	I impot	j janvier	s savon
> homme	~ mon	k café	S château
o haut	y payer	l elle	t tu
O fleur	w oui	m madame	v vous
u pour	? huit	n nation	z zéro

Figure 8. German Phonemes



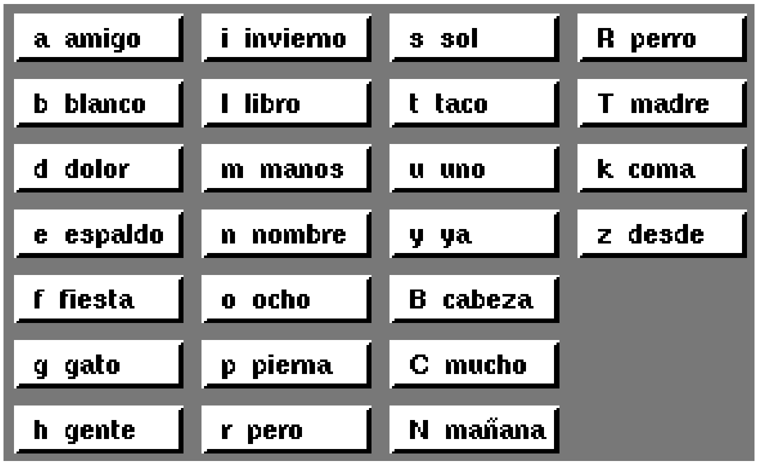
p Pein	C siCHer	Y hÜbsch	u blUt
b Bein	x buCH	7 höhle	& gEsetz
t Teich	h Hand	E bEtt	X bessER
d Deich	T Zahl	8 hölle	J EIs
k Kunst	$ deuTSCH	e bEEt	F hAUs
g Gunst	m Mein	a sAtz	V krEUz
f Fast	n Nein	A sAAt	l Lein
v Was	N diNG	> trOtz	R Rein
s taSSe	i lIEd	o slOwakei	j Jahr
z haSe	y sÜss	O rOt	
S waSCHen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

Figure 9. Japanese Phonemes



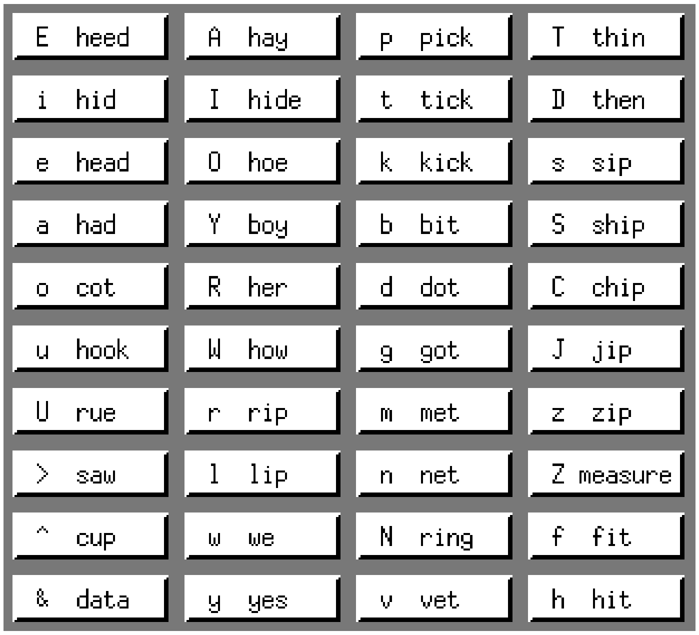
a aki	h hai	o go	u umi
b bengoshi	i ichi	p pan	w watashi
C hachi	j juu	r roku	y kyoo
d denwa	k kagi	s san	z zasshi
e enpitsu	m musuko	S shigatsu	
f futon	n ni	t tabemono	
g gakusei	N hon	T rokugatsu	

Figure 10. Latin-American Spanish Phonemes



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

Figure 11. US English Phonemes



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

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 6.) 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 6. 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 Avaya 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.

Further Processing

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

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 13 on page 59 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 13 on page 59 and Figure 12 on page 58 contain groups of possible words from which callers can choose. All of the wordlists for all of your applications constitute the vocabulary.

Using Wordlists in Scripts

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

- In Voice@Work, the wordlist is specified through the Node Inspector:
  - ~ Response tab of the Menu, Prompt and Collect, or Automenu node
  - ~ Confirmation tab of the Menu, Prompt and Collect, Automenu, or Voice Capture node

  For details, see the appropriate node description topics in the Voice@Work online help or in Chapter 31, “Node Descriptions” in *Using Voice@Work*, 585-313-207.

- In Script Builder, the wordlist is specified in the `Recog`*:* field on Page 2 of the Define Prompt and Collect screen step (the screen shows the name tags of all available wordlists). See “Defining the Prompt & Collect Action Step” in Chapter 7, “Defining the Transaction,” in *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217, for details.

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

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

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

For more information on how to create effective wordlists, see Positive Influences on FlexWord Recognition Accuracy (on page 63).

Figure 12. FlexWord Speech Application Example Using Menu Prompts

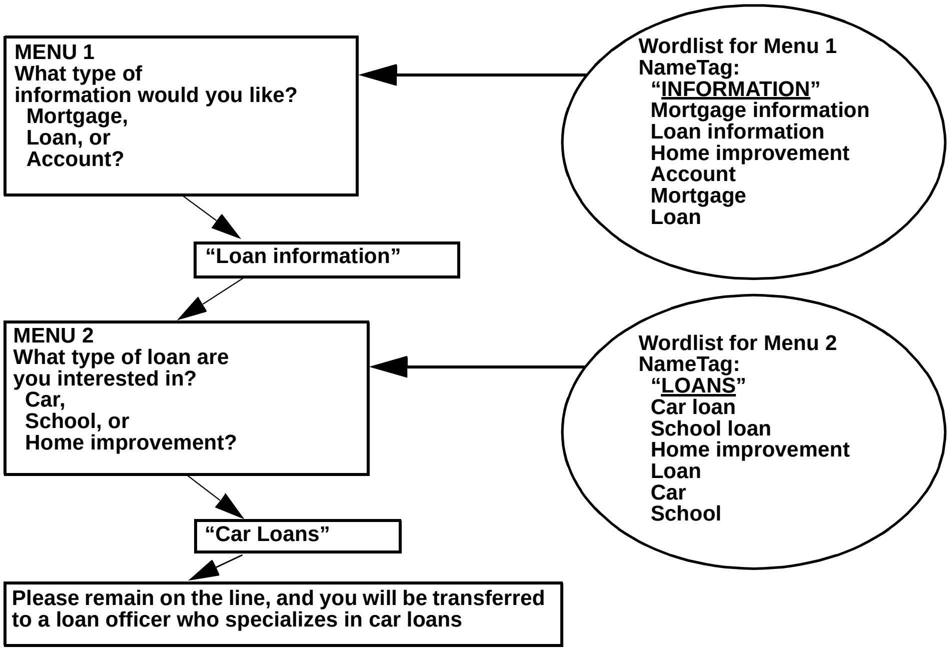
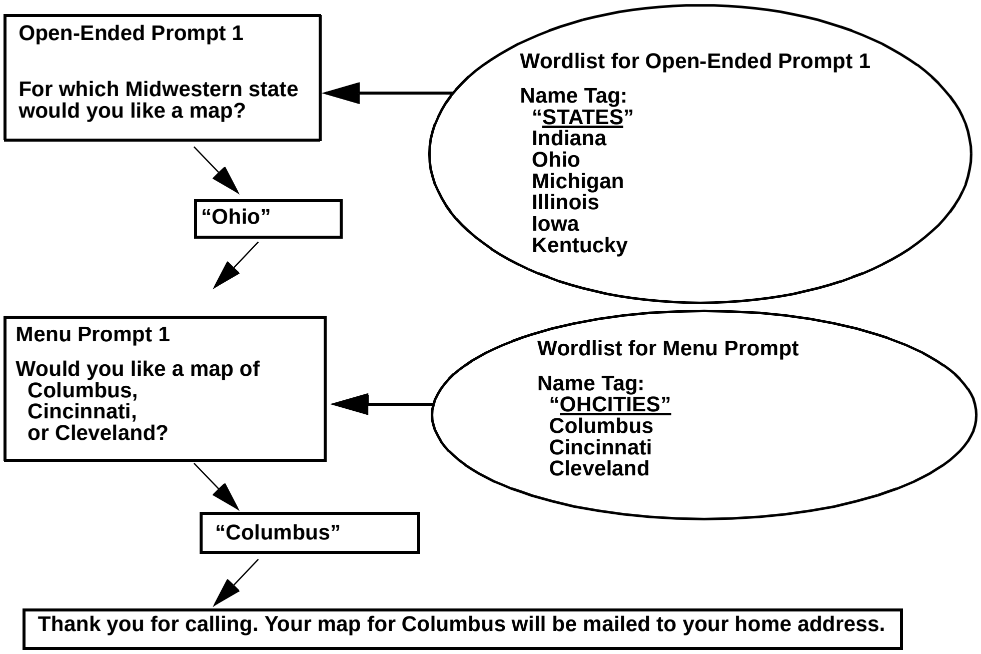


Figure 13. FlexWord Speech Application Example Using Open-Ended and Menu Prompts


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

Special FlexWord Recognition Accents

Table 7 lists the special or accented characters used in certain languages and the input strings used that represent these characters in FlexWord wordlists.

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

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

Table 7. Global Recognition Accents

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

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 CONVERSANT 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 61)) into the diskette drive when prompted.

   The system copies the wordlists to the **inactive** directory (**/att/asr/wordlists/inactive)** and then asks you if you want to activate any of the wordlists and run the **wl_gen** command.
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 61).

1 If there are no wordlists currently activated, make sure that the FlexWord function is *not* assigned to the SSP circuit card.

2 Make sure that the wordlist you want to activate is currently in the **inactive** directory (**/att/asr/wordlists/inactive**)

3 From the **inactive** directory, enter **ln *filename* /att/asr/wordlists/active** where *filename* is the name of the wordlist that you want to activate.

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

⚠ **CAUTION:**
Do not run Script Builder at the same time as the **wl_gen** command in step 4.

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

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

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

For more information about the **diagnose card** command, see *CONVERSANT System Version 8.0 Administration,* 585-313-510.

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

- ~ For a Voice@Work application, see the topics for transferring, installing and assigning applications in the Voice@Work online help or in *Using Voice@Work*, 585-313-207.
- ~ For a Script Builder application, see “Verifying and Installing the Application,” in Chapter 10, “Application Administration,” in *CONVERSANT System 7.0 Application Development with Script Builder,* 585-313-217.

7 Assign the FlexWord function to the SSP circuit card.

For the procedure, see “Assign SP/SSP Functions” in Chapter 3, “Voice System Administration” in *CONVERSANT System Version 8.0 Administration*, 585-313-510.

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 that FlexWord function is *not* assigned to the SSP circuit card.

   For more information, see “Assign SP/SSP Functions” in Chapter 3, “Voice System Administration” in *CONVERSANT System Version 8.0 Administration*, 585-313-510.

2. Make sure that 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 **ln** command when the wordlist was activated.

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

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

Removing Wordlists

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

1. Deactivate the wordlist.

   See Deactivating Wordlists (on page 63) 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 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 8, Putting It Together, for information about getting the most out of a FlexWord speech recognition application.

Effective Wordlists

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

Including Out-of-Vocabulary Terms

Note words that 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 no regular 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."

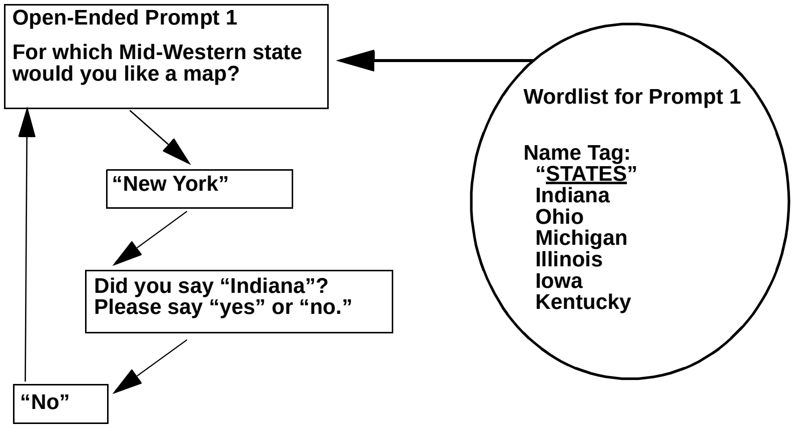
Confirmation and Reprompt Steps

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 14 provides an illustration of a confirmation reprompt.

Figure 14. Example of a Reprompt in a FlexWord Application



Informative Prompts

Lengthy prompts providing detailed response instructions may improve recognition accuracy. However, experienced users often find lengthy prompts annoying, since they must to 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 Recognition Accuracy

Low recognition accuracy rates are usually due to:

- Inexperienced callers
- Environmental factors
- Ineffective wordlists

Inexperienced Callers

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

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

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

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

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

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

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.

6 Using Natural Language Speech Recognition

Overview

This chapter provides an overview of the Natural Language Speech Recognition (NLSR) offer for CONVERSANT systems Version 7.0 and beyond.

Purpose

The purpose of this chapter is to familiarize you with the NLSR offer and software package. Topics include the following:

- What Is Natural Language Speech Recognition?
- The NLSR Offer
- NLSR Offer Architecture
- System Requirements
- Tasks Required to Use the NLSR Offer

What Is Natural Language Speech Recognition?

Natural Language Speech Recognition refers to a newer form of speech recognition technology that allows callers to interact with IVR systems in more natural and meaningful ways than earlier speech recognition technologies (WholeWord and FlexWord speech recognition). These earlier technologies share some limitations, both in the number of words or phrases that can be recognized, and also in their inability to take into account grammatical sentence structure. While these speech recognition technologies can recognize specific words or phrases, even when extraneous words or phrases are thrown in by the caller, they have no ability to recognize what part the recognized speech plays in the overall statement. In other words, these speech recognition technologies are all designed to *recognize* specific words or phrases, but not to *interpret* what they recognize.

Natural language speech recognition (NLSR) takes the speech recognition process several steps further by providing a more natural conversational interface with IVR systems. Not only can NLSR be used to recognize particular words and phrases, it can also interpret and assign meaning to the speech it recognizes.

For example, under the more basic forms of speech recognition, a caller can respond only to specific prompts, such as “Say ‘one’ if you want information about...” or “Say ‘yes’ if this is correct.” NLSR enables you to write applications that ask the caller more open-ended questions, such as a banking application that presents the caller with a list of options and then asks “What would you like to do?” Then, when the caller responds “I’d like to know the balance of my checking account, please,” the system can recognize what kind of information the caller is asking for (the *balance* in a *checking account*) and can automatically direct the call to a new prompt that would ask for the caller’s checking account number. This new technology provides a more natural way of interacting with callers.

It is worth noting that NLSR is also able to take into account grammatical structures. This allows it, for instance, to recognize and deal appropriately with differences in statements like the following caller responses:

> “I would like to fly from Chicago to LAX.”
>
> “I need to get from LAX to Chicago.”

NLSR is also capable of understanding natural numbers (“seventy-six” instead of “seven six”), natural dates (“July 26th” instead of “zero seven two six”) and natural currency (“25 dollars” instead of “two five zero zero”).

Because of the relatively complex nature of NLSR, it requires the use of larger vocabularies and grammars. For this reason, it often works best when a “proxy server” is used to do the speech recognition. Speech recognizers running on external speech recognition servers have at their disposal more CPU power and more memory than specialized voice processing circuit cards such as the Speech and Signal Processing (SSP) circuit card on the CONVERSANT system. They also offer more flexibility by allowing you to use multiple speech recognition servers, thus increasing the available CPU power and memory even more. That is the thinking behind this Avaya NLSR offer.

The NLSR Offer

The NLSR offer provides you with the ability to use the CONVERSANT voice response system with NLSR components from external vendors in a client/server configuration. (For a more complete description of this process, see Tasks Required to Use the NLSR Offer.)

The NLSR offer consists of the following components:

- **Natural Language Speech Recognition - Proxy/Client** software package
- **Natural Language Speech Recognition with the Intuity™ CONVERSANT® System** software installation instructions on paper
- **Natural Language Speech Recognition with the Intuity™ CONVERSANT® System** “electronic book” in the form of a .pdf file available from the GLS Documentation Internet site

The **Natural Language Speech Recognition - Proxy/Client** floppy diskette contains software that provides an interface to connect the CONVERSANT with an external NLSR server using a dedicated LAN connection. This proxy software, after verifying the availability of a recognizer resource on the NLSR server, passes the caller's utterance, in the form of digitized speech, to the recognizer. It then passes the results—that is, what the recognizer heard—back to the CONVERSANT. Thus, this software package provides a way for non-resident NLSR resources to appear as resident resources to CONVERSANT applications.

The **Natural Language Speech Recognition with the Intuity™ CONVERSANT® System** software installation documentation is a short instruction sheet that tells you how to install the NLSR software on the CONVERSANT system. It also has the instructions for downloading the full documentation for this offer (in the form of a .pdf file) from the Internet.

NLSR Offer Architecture

The NLSR offer assumes you are using one or more CONVERSANT systems in conjunction with one or more NLSR computers in a client/server configuration. This setup assumes that the CONVERSANT is the client and the NLSR computer is the server. Beyond that, the exact NLSR system architecture is dependent on what other components you are using in conjunction with this offer.

All computers in the system communicate using an Ethernet PCI local area network (LAN) connection.

The following scenarios depict various configurations that might be used to deploy NLSR applications. They include:

- A minimum configuration model, showing the most basic type of setup you might use
- A multiple-client/multiple-server configuration model

Minimum Configuration Model

Figure 15 on page 70 shows a prototypical model for the NLSR offer using a minimum- component configuration. It is followed by a description of how the system works (Table 8 on page 70). Note that this is only a sample of *one* way an NLSR system might be set up and used.

Figure 15. Minimum Configuration Model for NLSR Offer

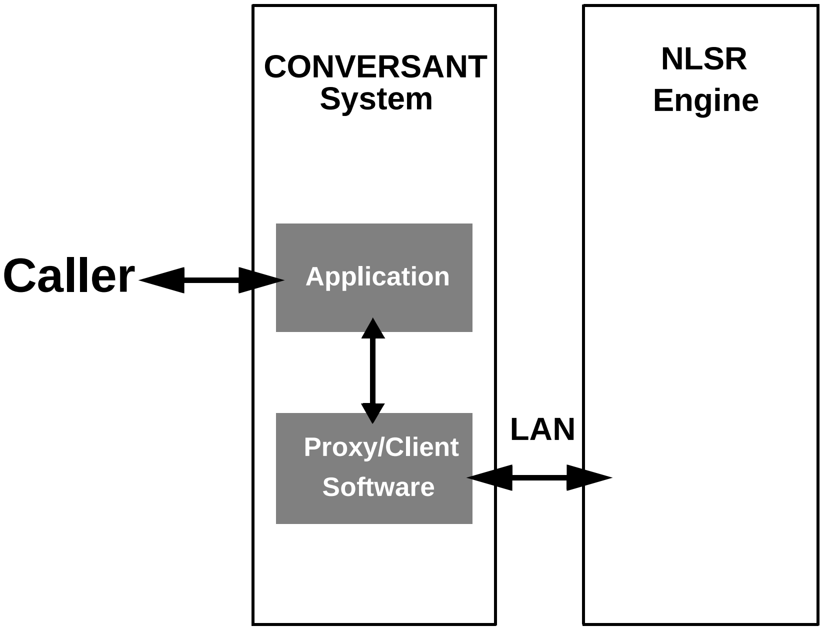


Table 8. NLSR Minimum Configuration Model Description

Action	Relevant Object(s) in Figure 1
1 The call is received on the CONVERSANT, and the NLSR call flow application begins running.	Caller, Application
2 The application answers the call and allocates a remote NLSR resource.	Application, Proxy/Client Software
3 (Optional) The application sends recognizer-specific commands to the remote NLSR resource to set it up for the caller's response.	Application, Proxy/Client Software, NLSR Engine
4 The application performs a Prompt and Collect action to obtain a response from the caller.	Application, Caller
5 The caller's spoken response is converted into digitized speech data and sent to the remote NLSR recognizer.	Proxy/Client Software, NLSR Engine
6 The remote NLSR recognizer processes the incoming digitized speech data.	NLSR Engine

1 of 2

Table 8. NLSR Minimum Configuration Model Description

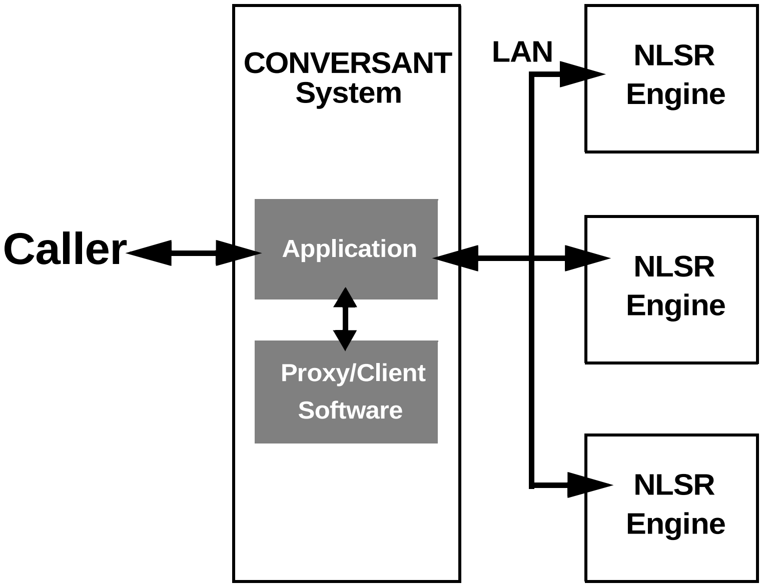
Action		Relevant Object(s) in Figure 1
7	The remote NLSR recognizer returns a recognition string to the application.	NLSR Engine, Application, Proxy/Client Software
8	(Optional) The application may query the remote NLSR resource one or more times to obtain or send vendor-specific data and information.	Application, Proxy/Client Software, NLSR Engine
9	The application takes whatever steps are necessary to complete the caller's request. This may involve multiple Prompt and Collects and several data exchanges between the CONVERSANT and the NLSR server.	Caller, Application, Proxy/Client Software, NLSR Engine
10	The application releases the remote NLSR resource.	Application, Proxy/Client Software
11	The call is terminated.	Application

2 of 2

Single-Client/Multiple-Server Model

Figure 16 on page 72 shows a model for the NLSR offer using a single CONVERSANT with multiple NLSR servers. It is followed by a description of how the system works (Table 9 on page 72). Note that this is only a sample of *one* way an NLSR system might be set up and used.

Figure 16. Single-Client/Multiple-Server Model



Note: Each NLSR engine in this setup *must* be administered identically to the other NLSR engines in the system. This means, among other things, that you can have only one type of external vendor NLSR speech engine per system.

Table 9. NLSR Single-Client/Multiple-Server Model Description

Action	Relevant Object(s) in Figure 1
1 The call is received on the CONVERSANT, and the NLSR call flow application begins running.	Caller, Application
2 The application answers the call and allocates a remote NLSR resource.	Application, Proxy/Client Software
3 The application performs a Prompt and Collect action to obtain a response from the caller.	Application, Caller
4 The caller’s spoken response is converted into digitized speech data and sent to the remote NLSR recognizer.	Proxy/Client Software, NLSR Engine
5 The remote NLSR recognizer processes the incoming speech data.	NLSR Engine
6 The remote NLSR recognizer returns a recognition string to the application.	NLSR Engine, Application, Proxy/Client Software

1 of 2

Table 9. NLSR Single-Client/Multiple-Server Model Description

Action	Relevant Object(s) in Figure 1
7 The application uses the Natural Language Interpreter to get “tokens” and values of “variables” from the recognition string.	Application
8 The application takes the steps necessary to complete the caller’s request.	Application
9 (Optional) Steps 3 through 8 are repeated any number of times, as required by the call flow.	
10 The application releases the remote NLSR resource.	Application, Proxy/Client Software
11 The call is terminated.	Application

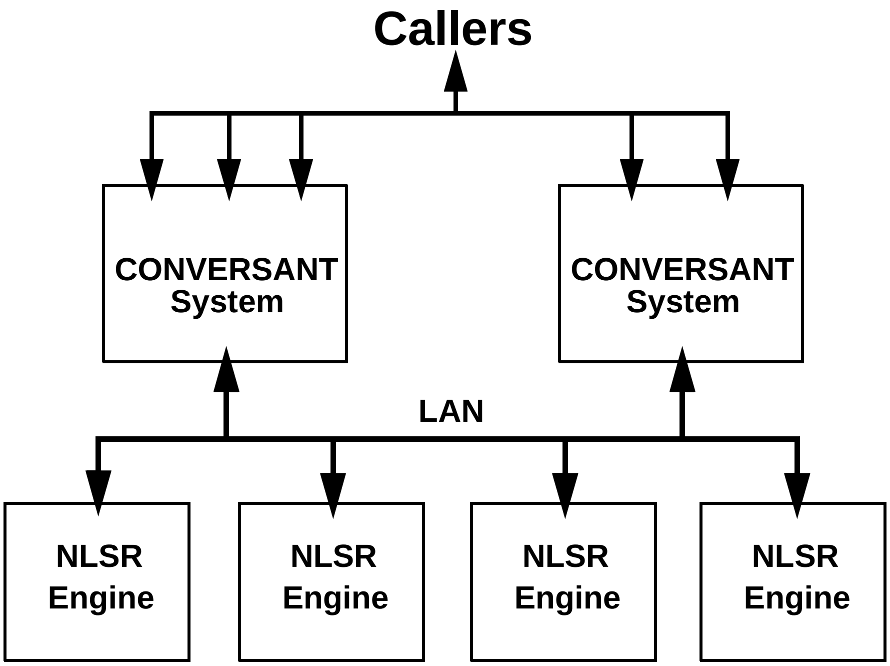
2 of 2

Multiple-Client/Multiple-Server Model

Figure 17 on page 74 shows a model for the NLSR offer using a multiple-client (CONVERSANT)/multiple-server configuration. Note that this is only a sample of *one* way such an NLSR system might be set up and used.

Note: When setting up a multiple-client/multiple-server configuration, keep in mind that each CONVERSANT system acts independently and cannot dynamically share resources with other CONVERSANT systems. In other words, you must make sure you assign your NLSR servers carefully. While it is possible to do otherwise, Avaya strongly recommends that you assign each NLSR server to a single CONVERSANT client, even though all machines in the system may share a single LAN. In the example below, for instance, you could assign the first two NLSR servers to the first CONVERSANT system and the third and fourth NLSR servers to the second CONVERSANT system. You must also carefully plan your incoming telephone lines for each CONVERSANT system.

Figure 17. Multiple-Client/Multiple-Server Model



Note: Each NLSR engine assigned to a single CONVERSANT client in this setup *must* be administered identically to the other NLSR engines in the system assigned to that same CONVERSANT client. This means, among other things, that you can have only one type of external vendor NLSR speech engine per CONVERSANT client.

System Requirements

This section provides information about the software and hardware requirements for the NLSR offer.

Software Requirements

The NLSR offer requires a number of software packages. Table 10 lists the required software packages and whether each package is available as part of the NLSR offer package.

Table 10. NLSR Required Software

Package Name	Available as part of the NLSR offer package?
Intuity CONVERSANT base system software (V7.0 or beyond)	**N**
rfu+b patches (V7.0 systems only): asp+bp1 and us+bp1	**N**
Appropriate circuit card drivers (for example, SSP, T1, and LAN)	**N**
INTUITY ASP driver package	**N**
Natural Language Speech Recognition - Proxy/Client software package	**Y**
Voice@Work or Script Builder with INTUITY Service Creation Integration Package	**N**
NLSR engine software (provided by the NLSR engine vendor and installed on remote NLSR server)	**N**

Hardware and LAN Requirements

Table 11 summarizes the hardware and LAN requirements for the NLSR offer. In addition to these requirements, you must also have one or more remote NLSR servers, depending on the NLSR engine requirements. Hardware requirements for these NLSR servers are provided by the NLSR engine vendor.

Table 11. Hardware and LAN Requirements

CONVERSANT System	LAN
MAP/40P or MAP/100P with: **Note:** This offer is not valid for MAP/5P systems, because they do not support the SSP circuit card. • At least one of the following telephony cards: ~ Tip/Ring (analog) ~ E1/T1 (digital) • Avaya-certified Ethernet PCI (Peripheral Component Interconnect) LAN circuit card • SSP circuit card **Note:** You cannot use SP circuit cards for the NLSR offer or for speech coding applications in general. • 128 Mb of RAM	**Note:** LAN installation is the customer's responsibility. Transmission Control Protocol/ Internet Protocol (TCP/IP) PCI Ethernet network operating at a speed of at least one of the following: • 10 Mbps (supports up to 28 channels) • 100 Mbps (supports up to 96 channels) **Note:** The speed required is influenced by the number of LAN- intensive features, such as ASAI or Proxy Text-to-Speech, running on the CONVERSANT. LAN segment dedicated for all CONVERSANT traffic

Tasks Required to Use the NLSR Offer

There are three basic sets of tasks you must accomplish in order to use the NLSR offer:

- First, install and configure the appropriate software and hardware.
- Next, design, develop, and test the NLSR application.
- Finally, deploy the NLSR application.

This section provides the basic steps for accomplishing these tasks.

Configuring the Hardware and Software

Table 12 lists the basic steps to install and administer the NLSR offer. Detailed procedures for each step are found in the section cross references listed in the second column.

Note: The following steps are for the configuration of CONVERSANT systems only; for the procedures to configure the NLSR server(s), see the documentation for your NLSR system.

Table 12. Steps for NLSR Initial Installation and Administration

Action/Step	Procedure Reference
1 Verify that you have all the required software and hardware for the installation set up properly.	System Requirements in this chapter
2 Verify that the Ethernet PCI LAN circuit card is installed and administered on the CONVERSANT system.	The *Intuity™ CONVERSANT® System Maintenance* book for your system, Chapter 2, “Installing or Replacing Circuit Cards, PCI Ethernet LAN Circuit Cards”
3 Install the **Natural Language Speech Recognition - Proxy/Client** software package on the CONVERSANT system.	*Natural Language Speech Recognition with the Intuity™ CONVERSANT® System*, 585-310-774, Chapter 2, “Software Installation and Removal”
4 Perform all the initial administration tasks prescribed for your system, taking special note of procedures that only apply to one configuration or another (that is, procedures specific to applications involving one or more external vendors).	*Natural Language Speech Recognition with the Intuity™ CONVERSANT® System*, 585-310-774, Chapter 3, “Administration”
5 Verify that connections between the CONVERSANT and any NLSR servers are established.	*Natural Language Speech Recognition with the Intuity™ CONVERSANT® System*, 585-310-774, Chapter 3, “Administration”

While there are other options you can use to enhance the NLSR offer, once you have completed these basic steps, you are ready to use the NLSR offer. For more information about these other options, see *Natural Language Speech Recognition with the Intuity™ CONVERSANT® System*, 585-310-774.

Creating and Testing the Application

Table 13 lists the basic steps required to create an NLSR application. Depending on your system configuration and what you want your NLSR application to do, you may need to add steps to or skip over steps in this basic model. This example only provides guidelines.

Table 13. Steps for Creating an NLSR Application

Action/Step	Procedure Reference(s)
1 Design the call flow architecture. You must analyze what you want your application to do and then create a call flow to move the caller through the transaction, paying special attention to the prompts used to elicit caller responses.	Select from among the following: • *Using Voice@Work*, 585-313-207 • *Intuity™ CONVERSANT® System Application Design with Script Builder* for your system • (Optional, if using an external vendor application design tool) Documentation for the external vendor tool In addition, you may find the following book helpful: *Intuity™ CONVERSANT® System Version 6.0 Application Design Guidelines*, 585-310-670
2 Analyze your call flow design to determine what kinds of responses you are likely to get from prompts, perhaps using customer interaction testing.	
3 Use the results of your analysis to develop your natural language vocabulary grammars.	• The documentation that came with your NLSR grammar development tool
4 Create a stub grammar.	*Natural Language Speech Recognition with the Intuity™ CONVERSANT® System*, 585-310-774
5 Develop the actual CONVERSANT call flow application using Voice@Work or Script Builder.	Select from among the following: • *Using Voice@Work*, 585-313-207 • *Intuity™ CONVERSANT® System Application Design with Script Builder* for your system
6 Verify that the application runs; debug as necessary.	*Intuity™ CONVERSANT® System Version 6.0 Application Design Guidelines*, 585-310-670

Note: Many of these steps are iterative and may need to be repeated several times before you are ready to deploy the application.

Deploying the Application

Table 14 lists the basic steps required to deploy an NLSR application. Depending on your system configuration and what you want your NLSR application to do, you may need to add steps to or skip over steps in this basic model. This example only provides guidelines.

Table 14. Steps for Deploying an NLSR Application

Action/Step	Procedure Reference(s)
1 Create, test, and debug your application.	Creating and Testing the Application, above
2 Transfer the call flow application(s) to the CONVERSANT system.	Select from among the following: • *Using Voice@Work*, 585-313-207 • *Intuity™ CONVERSANT® System Application Design with Script Builder* for your system
3 Install the call flow application(s) on the CONVERSANT system.	Select from among the following: • *Using Voice@Work*, 585-313-207 • *Intuity™ CONVERSANT® System Application Design with Script Builder* for your system
4 Install the appropriate software on the NLSR server.	Select from among the following: • Documentation for the NLSR engine
5 Place a test call to the system and verify that it responds as expected.	

Note: Many of these steps are iterative and may need to be repeated several times before you are ready to place the application in active service.

7 Recognizing Dial Pulse Input

Overview

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

Note: DPR is supported in both Voice@Work and Script Builder applications.

Topics covered include:

- Hardware (on page 81)
- Simultaneous Dial Pulse Capabilities (on page 81)
- Dial Pulse Recognition Types (on page 82)
- Grammars (on page 83)
- Recognition Confirmation (on page 83)
- Barge-In (on page 83)
- How Dial Pulse Recognition Works (on page 84)
- Recognition Accuracy (on page 84)
- DPR Troubleshooting (on page 85)

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

Recognition types are the choices associated with:

- The `Input ASR Mode` of the Response tab of the Menu, Prompt and Collect, and Automenu nodes in Voice@Work
- The `Input ASR Mode` of the Confirmation tab of the Menu, Prompt and Collect, Automenu, and Voice Capture nodes in Voice@Work
- The `Recog:` field on the Script Builder Define Prompt & Collect screen

The system uses the recognition type along with other fields on these tabs or screens to select a recognition grammar to be used for that prompt.

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

Table 15 shows the standard dial pulse recognition types.

Table 15. Standard Dial Pulse Recognition Types

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

Grammars

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

Grammars for DPR are modeled after grammars for speech recognition. DPR currently uses grammars only to do input restriction such as “ignore digit one” and pulse-to-digit mapping. Grammars provided for DPR in Script Builder applications are described in “Using Dial Pulse Recognition” in Chapter 8, “Using Optional Features,” of *CONVERSANT System Version 8.0 Application Development with Script Builder,* 585-313-217*.*

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 120 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 *CONVERSANT System Version 8.0 Administration*, 585-313-510.

The Recognizer

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

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

Recognition Accuracy

False recognition can occur if the caller is in a noisy environment or on a noisy telephone line. For example, an electrical disturbance that causes a “pop” sound on the telephone line can be recognized as a dial pulse digit. 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 can 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 is 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 CONVERSANT 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 16 to troubleshoot problems with DPR.

Table 16. DPR Troubleshooting

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

Common Failure Modes

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

Noise on the Line

Recognition failure due to noise on the line 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 recognizes incorrectly, 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.

8 Putting It Together

Overview

This chapter describes the considerations for using speech-related features with optimal effectiveness in applications, such as why certain features work well together and when you can use certain features together.

Topics include:

- Using NLSR with Other Features (on page 87)
- Using WholeWord and Dial Pulse Recognition Together (on page 88)
- Getting the Most Out of FlexWord Recognition (on page 89)
- Getting the Most out of NLSR (on page 90)
- Getting the Most out of Dial Pulse Recognition (on page 91)
- Getting the Most out of Text-to-Speech (on page 91)
- Using FlexWord Recognition and Text-to-Speech Together (on page 92)
- Using Text-to-Speech and Prerecorded Speech Together (on page 92)
- Using WholeWord, FlexWord, and Natural Language Speech Recognition Together (on page 93)
- WholeWord Speech Recognition Vs. FlexWord Recognition Accuracy (on page 94)
- Comparison of Recognition Types (on page 94)

Using NLSR with Other Features

Natural Language Speech Recognition (NLSR) allows you to create a more natural, conversational style of interacting with the CONVERSANT system. Depending on the external vendor’s speech recognition engine and its capabilities, you may not need any other speech recognition technologies in your system.

There may, however, be times when it is more advantageous to use a variety of speech recognition technologies, depending on what kind of speech you need to recognize at each prompt. NLSR can be used in combination with other speech recognition technologies such as WholeWord, FlexWord, and dial pulse recognition (DPR).

If, for example, you are looking for a simple "yes" or "no" response from a caller, it may be more efficient to use WholeWord speech recognition rather than tying up your NLSR resources. Or, if you want to provide dial pulse capabilities for your non-touchtone customers, you could use the DPR feature or NLSR for those prompts.

In general, you probably do not want to use NLSR in conjunction with FlexWord speech recognition, because, in general, NLSR will prove to be more flexible and reliable. (FlexWord technology was developed before NLSR technology became available.)

In the end, however, a great deal depends on your NLSR engine and its capabilities, when trying to answer the question of whether to use other speech recognition technologies in conjunction with NLSR.

For more information about using NLSR, see *Natural Language Speech Recognition with the Intuity™ CONVERSANT® System*, 585-310-774.

For more ideas on how to use NLSR with other speech recognition technologies, see Using WholeWord, FlexWord, and Natural Language Speech Recognition Together (on page 93).

Using WholeWord 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 touchtone application, see *CONVERSANT System Application Design Guidelines*, 585-310-670.

Getting the Most Out of FlexWord 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 inappropriate 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.

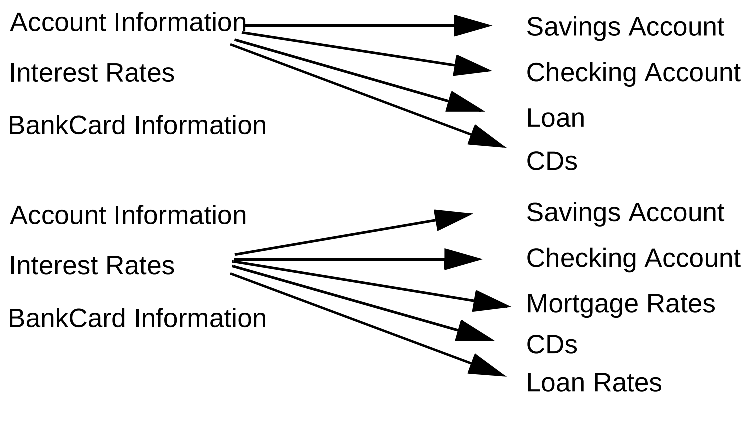
Applications Requiring Large Wordlists

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 18 shows an example of a menu-based application.

Figure 18. Example of Menu-Based Application



Getting the Most out of NLSR

For complete information about how to use and get the most out of NLSR, see *Natural Language Speech Recognition with the Intuity™ CONVERSANT® System*, 585-310-774.

Getting the Most out of Dial Pulse Recognition

As discussed in Chapter 7, 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 and grammars available.

  Grammars provided for DPR are described in “Using Dial Pulse Recognition”, in Chapter 8, “Using Optional Features,” of *CONVERSANT System Version 8.0 Application Development with Script Builder,* 585-313-217*.*
- 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 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 $CI_VALUE (the recognized word on the wordlist) easily. Also, if the wordlist changes, TTS still works— possibly without changes to your application. If you use prerecorded speech to speak the wordlist, you will have to record each time the wordlist changes.

Note: 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 Voice@Work or Script Builder application can speak prompts and announcements in prerecorded speech only, TTS only, 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 reinstall it on the CONVERSANT. Obviously, this would involved 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, FlexWord, and Natural Language Speech Recognition Together

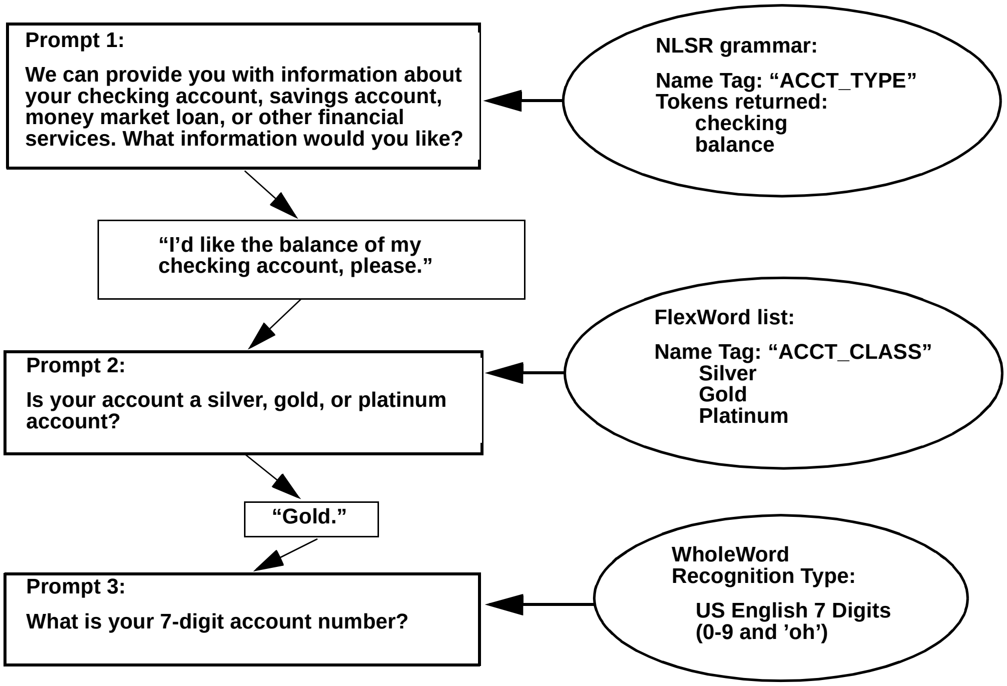
As stated in Using NLSR with Other Features (on page 87), WholeWord, FlexWord, and NLSR speech recognition can be used in a single application script. This allows flexibility in what you can ask callers to say. In Voice@Work and Script Builder applications, you can specify that a Prompt and Collect step use any of these three speech recognition technologies, depending on what you want to prompt callers to say. In general:

- If you want callers to say “yes” or “no,” a series of digits, or a single digit, choose WholeWord speech recognition.
- If you want callers to speak a word or phrase from a custom vocabulary, choose FlexWord speech recognition or NLSR.
- If you want to allow callers more freedom to speak their responses with a system capable of recognizing and interpreting what they say, use NLSR.

Note: Only one recognizer can be used in each single prompt.

The example in Figure 19 shows a model of a banking application that combines NLSR, FlexWord, and WholeWord speech recognition features.

Figure 19. Illustration of the Use of NLSR, FlexWord, and WholeWord Speech Recognition in a Single Application Script


Prompt 1:
We can provide you with information about your checking account, savings account, money market loan, or other financial services. What information would you like?
NLSR grammar:
Name Tag: “ACCT_TYPE”
Tokens returned:
checking
balance
“I’d like the balance of my checking account, please.”
Prompt 2:
Is your account a silver, gold, or platinum account?
FlexWord list:
Name Tag: “ACCT_CLASS”
Silver
Gold
Platinum
“Gold.”
Prompt 3:
What is your 7-digit account number?
WholeWord Recognition Type:
US English 7 Digits
(0-9 and ’oh’)

WholeWord Speech Recognition Vs. FlexWord 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 17 summarizes the similarities and differences between WholeWord speech recognition, FlexWord speech recognition, and DPR.

Table 17. Comparison of Speech Recognition Types

WholeWord Recognition	FlexWord Recognition	Natural Language Speech Recognition	Dial Pulse Recognition
Word-based	Phoneme-based	Depends on external vendor NLSR engine	Pulse-based
Requires data collection in model building for vocabulary words other than digits 0 through 9, “oh”, and the words “yes” and “no”	No data collection required in model building	Depends on external vendor NLSR engine	No data collection required in model building
Single or connected digits	Single word or phrase	Single or connected digits, single word or phrase	Single or connected digits
Standard and custom grammars	Customized wordlists	Custom or standard external vendor NLSR grammars	Standard grammars
Barge-in supported	Barge-in not supported	Depends on external vendor NLSR engine	Barge-in not supported

1 of 2

Table 17. Comparison of Speech Recognition Types

WholeWord Recognition	FlexWord Recognition	Natural Language Speech Recognition	Dial Pulse Recognition
Phrase screening supported	Phrase screening supported	Phrase screening supported	Phrase screening supported
Limited vocabulary	2000-word vocabulary	Vocabulary limited by external vendor NLSR engine	Limited vocabulary
Word spotting supported	Word spotting supported	Depends on external vendor NLSR engine, but word spotting usually supported	Word spotting not supported

2 of 2

A Enhanced Basic Speech Formats

Overview

Enhanced basic speech (EBS) formats specify algorithms that convert the values in fields (Script Builder) and variables (Voice@Work) to prerecorded phrases. EBS formats are available for application development in numerous languages.

This appendix contains tables that list the formats in the EBS packages for all supported languages (Table 20 on page 102 through Table 42 on page 137). The tables are designed to be used as reference when you create speech applications using EBS. Background information is also provided for the EBS formats.

For information on using EBS formats:

- In Script Builder applications, see "Defining Announce" and "Defining Prompt & Collect" in Chapter 7, "Defining the Transaction," of *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217.
- In Voice@Work applications, see the chapter on variables in *Using Voice@Work*, 585-313-207.

Languages Available

Table 18 lists the languages for which EBS is available and, for each language, shows the gender of the speaking voice (female or male), the abbreviation, and the phrase pool name.

Table 18. EBS Languages

EBS Version 6.0	Speaking Voice	Abbreviation	Phrase Pool Name	Reference
Australian English	female	ebsAE	AU_English	Table 20 on page 102
Brazilian Portuguese	female	ebsBP	BR_Portug	Table 21 on page 104
Canadian French	female	ebsCF	Can_French	Table 22 on page 105
Cantonese Chinese	female	ebsCT	Cantonese	Table 23 on page 108
Castilian Spanish	female	ebsCS	Cast_Span	Table 24 on page 109
Czech	female	ebsCZ	Czech	Table 25 on page 110
Dutch	female	ebsDT	Dutch	Table 26 on page 113
French	female	ebsFR	French	Table 27 on page 114
German	female	ebsGR	German	Table 28 on page 116
Hindi	female	ebsHD	Hindi	Table 29 on page 117
Hungarian	female	ebsHU	Hungarian	Table 30 on page 119
Indonesian	Female	ebsIN	Indonesian	Table 31 on page 121
Italian	Female	ebsIT	Italian	Table 32 on page 122
Japanese	Female	ebsJP	Japanese	Table 33 on page 123
Korean	Female	ebsKO	Korean	Table 34 on page 124
Latin-American Spanish	Female	ebsLS	LA_Spanish	Table 35 on page 125
Malay	Female	ebsMA	Malay	Table 36 on page 127
Mandarin Chinese	Female	ebsMD	Mandarin	Table 37 on page 128
Polish	Female	ebsPL	Polish	Table 38 on page 129
Slovak	Female	ebsSQ	Slovak	Table 39 on page 132
Thai	Female	ebsTH	Thai	Table 40 on page 134
United Kingdom (UK) English	Female	ebsUK	UK_English	Table 41 on page 135
United States (US) English	Female	ebsUS	US_English	Table 42 on page 137
	Male	ebsUSM		

Background Information

Common Formats

The EBS package for each language contains the following common formats:

- D (date)
- T (time)
- N (numbers)
- N$ (currency)
- C (characters)

These formats, while common to all languages, may be defined differently from language to language, as appropriate to the conventions of usage in each language. For example, T (time) is spoken as THMAM in some languages, but as TH24M in others, depending on the language specification (see Variations on the Common Formats (on page 99)). The definition reflects the most normal or customary manner of speaking the phrase in the language.

Variations on the Common Formats

The common EBS formats may be enhanced to include different ways to be spoken, depending on the language specification. The most frequently occurring variations on the common formats are listed below. However, some languages also contain other variations. (For information on inflections used in the formats, see Inflections (on page 100).)

- Date formats always start with D, and these modifiers may follow:
  - ~ YY — year spoken with four digits
  - ~ Y — year spoken with two digits
  - ~ M — month spoken as a number (rather than as a word)
  - ~ SP — month spoken as a word (rather than as a number)
  - ~ D — day (number)
  - ~ W— weekday (name)
  - ~ f — falling inflection
- Time formats always start with T, and these modifiers may follow:
  - ~ H — hour
  - ~ 24 — 24-hour clock
  - ~ M — minutes
  - ~ AM — 12-hour clock (AM/PM)
  - ~ f — falling inflection
- Number formats always start with N, and these modifiers may follow:
  - ~ M — male form
  - ~ F — female form
  - ~ N — neutral form
  - ~ m (or mmm) — medial inflection
  - ~ f (or mmf) — falling inflection
  - ~ r (or rmm)— rising inflection
  - ~ rmf — total inflection

- Decimal number formats always start with ND, and the number that follows indicates the number of decimal places after the decimal point.

  The input string is assumed not to have a decimal point in it. For example, if 12345678 is input with format ND4, the number 1234.5678 is spoken.

- Currency formats always start with N$, and these modifiers may follow:
  - ~ Dx — where x indicates the number of decimal places
  - ~ f — falling inflection
- Character formats always start with C, and these modifiers may follow:
  - ~ m (or mmm) — medial inflection
  - ~ f (or mmf) — falling inflection
  - ~ r (or rmm) — total inflection
  - ~ rmf — total inflection

Note: When you select a variable Type in the Spoken As column in the Voice@Work Variables Manager, by default Voice@Work automatically assigns the variable a common format (D, C, T, N, or N$). You can either accept the default or change it to one of the variations available for that common format. For example, when you select the character Type, the character variable defaults to the C format. You can then select another character format, if one is available in the language you have selected. For US English, (see Table 42 on page 137), you could select Crmf, Crmm, Cmmf or Cmmm. (C is spoken as Cmmm, however.) See the chapter on variables in *Using Voice@Work*, 585-313-207, for more information on defaults for formats in the Voice@Work interface.

Inflections

Inflection is a change in voice intonation appropriate to the context in which a speech phrase occurs. For example, typically in natural speech, rising inflection occurs in questions, medial inflection (no inflection) occurs in the middle of words and phrases, and falling inflection occurs at the end of words and phrases. The EBS package includes formats for inflections.

The EBS format tables use the following terminology to describe inflections:

- **Rising inflection** — The first character or number is spoken with rising inflection and all other others with medial inflection.
- **Falling inflection** — The last character or number is spoken with falling inflection and all others with medial inflection.
- **Total inflection** — The first character or number is spoken with rising inflection, the last with falling inflection, and all others with medial inflection.

Note: Unless otherwise specified, formats use medial inflection.

Grammatical Gender

Words in some languages are spoken differently depending on whether they are grammatically male, female, or neutral. The EBS format tables use the terminology *male form*, *female form*, and *neutral form* to describe gender-specific formats.

Converting Applications to Other Languages

An application containing EBS formats can have the formats converted to another language by changing the language option. However if the original application uses formats not available for the new language, then the converted application will not be identical. Therefore, advance planning is advisable for applications that may be converted to other languages. A recommended strategy is to create the original application using only the common formats available in all languages, create duplicate applications for the target languages, and adjust the formats on a per-application basis, if needed. (See Common Formats (on page 99) for more information.)

Maximum Values for Numbers and Currency

A variable defined as a number or as currency on the CONVERSANT system has 4 bytes and can support a numeric value up to 2,147,483,647 (2^{31} - 1). To support larger numbers, for some languages the number and currency formats also accept inputs as characters. These character variables can handle values up to 15 digits (999,999,999,999,999) plus a decimal point and can be preceded by a minus sign (-) to indicate a negative number.

Table 19 distinguishes the languages that receive only number variable support from EBS for number and currency formats from the languages that also receive character variable support.

Table 19. Support for Number and Currency Formats

Number Variable Support Only	Character Variable Support
• Australian English	• Brazilian Portuguese
• Cantonese Chinese	• Canadian French
• Castilian Spanish	• Czech
• Dutch	• French
• German	• Hungarian
• Hindi	• Italian
• Indonesian	• Japanese
• Korean	• Latin-American Spanish
• Malay	• Polish
• Mandarin Chinese	• Slovak
• Thai	• UK English
• US English	

Speaking Phrase Numbers and Packed Talkfile Numbers

The NX format is a CONVERSANT system feature used to speak:

- A phrase with a specified phrase tag number
- Packed talkfile numbers and phrase numbers

This format is included in the EBS tables for all supported languages.

See The Speech Filesystem (on page 1) in Chapter 1, Overview of Speech, for information on speech phrases and talkfiles. See Chapter 11, "Using Advanced Features," in *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217, for information on the NX format, including use of the **pack_phrNX** and **unpack_phrNX** external functions. For information on using external functions in Voice@Work, see the chapter on the External Functions Manager in *Using Voice@Work*, 585-313-207.

EBS Format Tables

The following tables (Table 20 through Table 42 on page 137) provide a complete list of EBS formats by language.

Note: For a list of all languages for which EBS is available, see Table 18 on page 98.

Australian English EBS Formats

Table 20. Australian English EBS Formats

Format	Description
D	Date — same as DMSPDY
DMDYY	Date — 2-digit month, day, 4-digit year
DMDY	Date — 2-digit month, day, year
DMD	Date — 2-digit month, day, no year
DMSPDYY	Date — spoken month, 2-digit day, 4-digit year
DMSPDY	Date — spoken month, 2-digit day, 2-digit year
DMSPD	Date — spoken month, 2-digit day, no year
T	Time — same as THMAM
TH24M	Time — hour, minute, 24-hour clock
THMAM	Time — hour, minute, AM/PM
N	Number — same as Nmmm
Nrmf	Number — total inflection
Nmmf	Number — falling inflection
Nrmm	Number — rising inflection
Nmmm	Number — medial inflection
	1 of 2

Table 20. Australian English EBS Formats

Format	Description
ND	Decimal number — no decimal places This format may not be valid for Voice@Work. See the chapter on variables in *Using Voice@Work*, 585-313-207, for specific information.
ND0	Decimal number — no decimal places This format may not be valid for Voice@Work. See the chapter on variables in *Using Voice@Work*, 585-313-207, for specific information.
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
N$	Currency — same as N$D0
N$D0	Currency — whole dollars
N$D2	Currency — dollars and cents
N$D1	Currency — dollars with 1 decimal place
N$D3	Currency — dollars with 3 decimal places
N$D4	Currency — dollars with 4 decimal places
N$D5	Currency — dollars with 5 decimal places
N$D6	Currency — dollars with 6 decimal places
N$D7	Currency — dollars with 7 decimal places
N$D8	Currency — dollars with 8 decimal places
N$D9	Currency — dollars with 9 decimal places
C	Characters — same as Cmmm
Crmf	Characters — spoken individually, total inflection
Crmm	Characters — spoken individually, rising inflection
Cmmf	Characters — spoken individually, falling inflection
Cmmm	Characters — spoken individually, medial inflection
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).

2 of 2

Brazilian Portuguese EBS Formats

Table 21. Brazilian Portuguese EBS Formats

Format	Description
D	Date — same as DMSPDYY
DMSPDYY	Date — spoken month, 2-digit day, 4-digit year
DMSPDY	Date — spoken month, 2-digit day, year
DMSPD	Date — spoken month, 2-digit day, no year
DMSPfYY	Date — same as DMSPDYY but with falling inflection
DMSPfY	Date — same as DMSPDY but with falling inflection
DMSPf	Date — same as DMSPD but with falling inflection
T	Time — same as THMAM
THMAM	Time — hour, minute, AM/PM
TH24M	Time — hour, minute, 24-hour clock
THMAMf	Time — hour, minute, AM/PM, falling inflection
TH24Mf	Time — hour, minute, 24-hour clock, falling inflection
N	Number — same as Nrmf
Nmf	Number — falling inflection
Nrm	Number — rising inflection
Nrmf	Number — total inflection
NM	Number — male form
NF	Number — female form
NN	Number — neutral form
ND0	Decimal number — no decimal places
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
N$	Currency — same as N$D2
N$D2	Currency — reais and centavos
C	Characters — same as Cmmm
Crmf	Characters — spoken individually, total inflection

1 of 2

Table 21. Brazilian Portuguese EBS Formats

Format	Description
Crmm	Characters — spoken individually, rising inflection
Cmmf	Characters — spoken individually, falling inflection
Cmmm	Characters — spoken individually, medial inflection
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).
	2 of 2

Canadian French EBS Formats

Table 22. Canadian French EBS Formats

Format	Description
D	Date — same as DMSPDYY
DMSPDYY	Date — spoken month, 2-digit day, 4-digit year
DMSPDY	Date — spoken month, 2-digit day and year
DMSPD	Date — spoken month, 2-digit day, no year
DMSPfYY	Date — same as DMSPDYY, but with falling inflection
DMSPfY	Date — same as DMSPDY, but with falling inflection
DMSPf	Date — same as DMSPD, but with falling inflection
T	Time — same as TH24M
THMAM	Time — hour, minute, AM/PM
THMAMf	Time — hour, minute, AM/PM, falling inflection
TH24M	Time — hour, minute, 24-hour clock
TH24Mf	Time — hour, minute, 24-hour clock, falling inflection
TPD	Time — same as TH24M
THMAMPD	Time — PDT (Pacific Daylight Saving Time), hour, minute, AM/PM
TH24MPD	Time — PDT, hour, minute, 24-hour clock
THMAMPDf	Time — PDT, hour, minute, AM/PM, falling inflection
TH24MPDf	Time — PDT, 24-hour clock, falling inflection
TPS	Time — same as TH24M
THMAMPS	Time — PST (Pacific Standard Time), hour, minute, AM/PM
TH24MPS	Time — PST, hour, minute, 24-hour clock
THMAMPSf	Time — PST, hour, minute AM/PM, falling inflection
TH24MPSf	Time — PST, hour, minute, 24-hour clock, falling inflection
TMD	Time — same as TH24M
THMAMMD	Time — MDT (Mountain Daylight Saving Time), hour, minute, AM/PM
TH24MMD	Time — MDT, hour, minute, 24-hour clock
	1 of 4

Table 22. Canadian French EBS Formats

Format	Description
THMAMMDf	Time — MDT, hour, minute, AM/PM, falling inflection
TH24MMDf	Time — MDT, hour, minute, 24-hour clock, falling inflection
TMS	Time — same as TH24M
THMAMMS	Time — MST (Mountain Standard Time), hour, minute, AM/PM
TH24MMS	Time — MST, hour, minute, 24-hour clock
THMAMMSf	Time — MST, hour, minute, AM/PM, falling inflection
TH24MMSf	Time — MST, hour, minute, 24-hour clock, falling inflection
TCD	Time — same as TH24M
THMAMCD	Time — CDT (Central Daylight Saving Time), hour, minute, AM/PM
TH24MCD	Time — CDT, hour, minute, 24-hour clock
THMAMCDf	Time — CDT, hour, minute, AM/PM, falling inflection
TH24MCDf	Time — CDT, hour, minute, 24-hour clock, falling inflection
TCS	Time — same as TH24M
THMAMCS	Time — CST (Central Standard Time), hour, minute, AM/PM
TH24MCS	Time — CST, hour, minute, 24-hour clocks
THMAMCSf	Time — CST, hour, minute, AM/PM, falling inflection
TH24MCSf	Time — CST, hour, minute, 24-hour clock, falling inflection
TAD	Time — same as TH24M
THMAMAD	Time — ADT (Atlantic Daylight Saving Time), hour, minute, AM/PM
TH24MAD	Time — ADT, hour, minute, 24-hour clock
THMAMADf	Time — ADT, hour, minute, AM/PM, falling inflection
TH24MADf	Time — ADT, hour, minute, 24-hour clock, falling inflection
TAS	Time — same as TH24M
THMAMAS	Time — AST (Atlantic Standard Time), hour, minute, AM/PM
TH24MAS	Time — AST, hour, minute, 24-hour clocks
THMAMASf	Time — AST, hour, minute, AM/PM, falling inflection
TH24MASf	Time — AST, hour, minute, 24-hour clock, falling inflection
N	Number — same as Nrmf
Nmf	Number — falling inflection
Nrm	Number — rising inflection
Nrmf	Number — total inflection
NM	Number — male form
NF	Number — female form
NN	Number — neutral form
ND0	Decimal number — no decimal places, “virgule zero”

2 of 4

Table 22. Canadian French EBS Formats

Format	Description
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
NDF0	Decimal number — no decimal places
NDF1	Decimal number — 1 decimal place, decimal part spoken as number
NDF2	Decimal number — 2 decimal places, decimal part spoken as number
NDF3	Decimal number — 3 decimal places, decimal part spoken as number
NDF4	Decimal number — 4 decimal places, decimal part spoken as number
NDF5	Decimal number — 5 decimal places, decimal part spoken as number
NDF6	Decimal number — 6 decimal places, decimal part spoken as number
NDF7	Decimal number — 7 decimal places, decimal part spoken as number
NDF8	Decimal number — 8 decimal places, decimal part spoken as number
NDF9	Decimal number — 9 decimal places, decimal part spoken as number
N$	Currency — same as N$D2
N$D2	Number — dollars and cents
N$F	Currency — whole dollars
N$F0	Currency — whole dollars, no decimal places
N$F1	Currency — dollars with 1 decimal place
N$F2	Currency — dollars and cents
N$F3	Currency — dollars with 3 decimal places
N$F4	Currency — dollars with 4 decimal places
N$F5	Currency — dollars with 5 decimal places
N$F6	Currency — dollars with 6 decimal places
N$F7	Currency — dollars with 7 decimal places
N$F8	Currency — dollars with 8 decimal places
N$F9	Currency — dollars with 9 decimal places
C	Characters — same as Cmmm
Crmf	Characters — spoken individually, total inflection
Crmm	Characters — spoken individually, rising inflection
	3 of 4

Table 22. Canadian French EBS Formats

Format	Description
Cmmf	Characters — spoken individually, falling inflection
Cmmm	Characters — spoken individually, medial inflection
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).
	4 of 4

Cantonese Chinese Enhanced Basic Speech Formats

Note: For United States currency the word “maygum” is spoken first to indicate United States currency.

Table 23. Cantonese Chinese EBS Formats

Format	Description
D	Date — same as DYYMD
DYMDD	Date — 4-digit year, 2-digit month and day —
DMD	Date — month followed by day
T	Time — same as THMAM
THMAM	Time — time, period followed by time
N	Number — integer, medial inflection
ND0	Decimal number — no decimal places
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
N$	Currency — same as N$D0
N$D0	Currency — whole dollars
N$D2	Currency — dollars and cents
N$D1	Currency — dollars with 1 decimal place
N$D3	Currency — dollars with 3 decimal places
N$D4	Currency — dollars with 4 decimal places
N$D5	Currency — dollars with 5 decimal places
N$D6	Currency — dollars with 6 decimal places
N$D7	Currency — dollars with 7 decimal places
	1 of 2

Table 23. Cantonese Chinese EBS Formats

Format	Description
N$D8	Currency — dollars with 8 decimal places
N$D9	Currency — dollars with 9 decimal places
NYD0	Currency — whole yuan
NYD1	Currency — yuan with 1 decimal place
NYD2	Currency — yuan, miao, and seen
NYD3	Currency — yuan with 3 decimal places
NYD4	Currency — yuan with 4 decimal places
NYD5	Currency — yuan with 5 decimal places
NYD6	Currency — yuan with 6 decimal places
NYD7	Currency — yuan with 7 decimal places
NYD8	Currency — yuan with 8 decimal places
NYD9	Currency — yuan with 9 decimal places
C	Characters — same as Cmmm
Cmmm	Characters — spoken individually, medial inflection
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).
	2 of 2

Castilian Spanish Enhanced Basic Speech Formats

Table 24. Castilian Spanish EBS Formats

Format	Description
D	Date — same as DMSPDYY
DMSPDYY	Date — spoken month, 2-digit day, 4-digit year
DMSPDY	Date — spoken month, 2-digit day and year
DMSPD	Date — spoken month, 2-digit day, no year
T	Time — same as THMAM
THMAM	Time — hour, minute, AM/PM
TH24M	Time — hour, minute, 24-hour clock
N	Number — same as NN
NM	Number — male form
NF	Number — female form
NN	Number — neutral form
ND0	Decimal number — no decimal places
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
	1 of 2

Table 24. Castilian Spanish EBS Formats

Format	Description
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
N$	Currency — same as N$D0
N$D0	Currency — whole pesetas
N$D2	Currency — pesetas and centimos
N$EU	Currency —same as N$D0EU
N$D0EU	Currency — whole Euros
N$D2EU	Currency — Euros and cents
C	Characters — same as Cmmm
Crmf	Characters — spoken individually, total inflection
Crmm	Characters — spoken individually, rising inflection
Cmmf	Characters — spoken individually, falling inflection
Cmmm	Characters — spoken individually, medial inflection
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).
	2 of 2

Czech Enhanced Basic Speech Formats

Table 25. Czech EBS Formats

Format	Description
D	Date — same as DDMSPYY
DDMSPYY	Date — day, spoken month, 4-digit year
DDMYY	Date — day, month, 4-digit year
DMMSP	Date — day, spoken month, no year
DDM	Date — day, month, no year
DDMSPYYf	Date — day, spoken month, 4-digit year, falling inflection
DDMYYf	Date — day, month, 4-digit year, falling inflection
DMMSPf	Date — day, spoken month, no year, falling inflection
DDMf	Date — day, month, no year, falling inflection
T	Time — same as TH24M
	1 of 3

Table 25. Czech EBS Formats

Format	Description
TH24M	Time — hour, minute, 24-hour clock
TH24MS	Time — hour, minute, second, 24-hour clock
TH24Mf	Time — hour, minute, falling inflection
TH24MSf	Time — hour, minute, second, falling inflection
N	Number — same as NMm
NMm	Number — male form, medial inflection
NMf	Number — male form, falling inflection
NFm	Number — female form, medial inflection
NFf	Number — female form, falling inflection
NNm	Number — neutral form, medial inflection
NNf	Number — neutral form, falling inflection
ND0	Decimal number — whole number spoken as "number comma 0", no decimal places
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
NDX	Decimal number — variable number of decimal places, decimal separator in argument NOTE: A number passed to a function can contain either a period (.) or a comma (,) to identify a number of decimal digits spoken for that number.
NDX1	Decimal number — 1 decimal place, spoken as a fraction
NDX2	Decimal number — 2 decimal places, spoken as a fraction
NDX3	Decimal number — 3 decimal places, spoken as a fraction
NDXS	Decimal number — up to 3 decimal places, spoken as a fraction; otherwise NDX
NDX0S	Decimal number — up to 3 decimal places, spoken as number with leading zeros; otherwise NDX
ND0f	Decimal number — whole number, spoken as "number comma 0", falling inflection
ND1f	Decimal number —1 decimal place, falling inflection
ND2f	Decimal number —2 decimal places, falling inflection
ND3f	Decimal number —3 decimal places, falling inflection
ND4f	Decimal number —4 decimal places, falling inflection
ND5f	Decimal number —5 decimal places, falling inflection
ND6f	Decimal number —6 decimal places, falling inflection
	2 of 3

Table 25. Czech EBS Formats

Format	Description
ND7f	Decimal number —7 decimal places, falling inflection
ND8f	Decimal number — 8 decimal places, falling inflection
ND9f	Decimal number — 9 decimal places, falling inflection
NDXf	Decimal number — variable number of decimal places, falling inflection, decimal separator in argument
NDX1f	Decimal number —1 decimal place, spoken as a fraction, falling inflection
NDX2f	Decimal number — 2 decimal places, falling inflection
NDX3f	Decimal number — 3 decimal places, spoken as a fraction, falling inflection
NDX0Sf	Decimal number — up to 3 decimal places, spoken as a number with leading zeros; otherwise NDX
NDXSf	Decimal number — up to 3 decimal places, spoken as a fraction, falling inflection; otherwise NDX
N$	Currency — same as N$D0
N$D0	Currency — whole crowns
N$D2	Currency — crowns and hellers
N$DX	Currency — crowns, spoken as a decimal number with a variable number of decimal places, decimal separator in argument
N$D2f	Currency — crowns and hellers, falling inflection
N$EU	Currency — same as N$D0EU
N$D0EU	Currency — whole Euros
N$D2EU	Currency — Euros and cents
N$DXEU	Currency — Euros, spoken as a decimal number with a variable number of decimal places, decimal separator in argument
N$D2fEU	Currency — Euros and cents, falling inflection
C	Characters — same as Cmm
Cspd	Characters — spoken individually, variable pace
Cmf	Characters — spoken individually, falling inflection
Cmm	Characters — spoken individually, medial inflection
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).

3 of 3

Dutch EBS Formats

Table 26. Dutch EBS Formats

Format	Description
D	Date — same as DMSPDYY
DMSPD	Date — spoken month, 2-digit day, 4-digit year
DMSPD	Date — spoken month, 2-digit day, 2-digit year
DMSPD	Date — spoken month, 2-digit day, no year
T	Time — same as TH24M
TH24M	Time — hour, minute, 24-hour clock
N	Number — same as NN
ND0	Decimal number — no decimal places
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
N$	Currency — same as N$D0
N$D0	Currency — whole gulden
N$D2	Currency — gulden and cents
N$EU	Currency — whole Euros
N$D0EU	Currency — whole Euros
N$D2EU	Currency — Euros and cents
C	Characters — same as Cmmm
Crmf	Characters — spoken individually, total inflection
Crmm	Characters — spoken individually, rising inflection
Cmmf	Characters — spoken individually, falling inflection
Cmmm	Characters — spoken individually, medial inflection
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).

French EBS Formats

Table 27. French EBS Formats

Format	Description
D	Date — same as DMSPDYY
DMSPDYY	Date — spoken month, 2-digit day, 4-digit year
DMSPDY	Date —spoken month, 2-digit day and year
DMSPD	Date — spoken month, 2-digit day, no year
DMSPfYY	Date — same as DMSPDYY but with falling inflection
DMSPfY	Date — same as DMSPDY but with falling inflection
DMSPf	Date — same as DMSPD but with falling inflection
DMSPlYY	Date — same as DMSPDYY but preceded by “le”
DMSPlY	Date — same as DMSPDY but preceded by “le”
DMSPl	Date — same as DMSPD but preceded by “le”
T	Time — same as TH24M
TH24M	Time — hour, minute, 24-hour clock
TH24Mf	Time — hour, minute, 24-hour clock, falling inflection
N	Number — same as Nrmf
Nmf	Number — falling inflection
Nrm	Number — rising inflection
Nrmf	Number — total inflection
NN	Number — neutral form
NF	Number — female form
NM	Number — male form
ND0	Decimal number — no decimal places, “virgule zero”
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
NDF0	Decimal number — no decimal place
NDF1	Decimal number — 1 decimal place, decimal part spoken as number
NDF2	Decimal number — 2 decimal places, decimal part spoken as number

1 of 3

Table 27. French EBS Formats

Format	Description
NDF3	Decimal number — 3 decimal places, decimal part spoken as number
NDF4	Decimal number — 4 decimal places, decimal part spoken as number
NDF5	Decimal number — 5 decimal places, decimal part spoken as number
NDF6	Decimal number — 6 decimal places, decimal part spoken as number
NDF7	Decimal number — 7 decimal places, decimal part spoken as number
NDF8	Decimal number — 8 decimal places, decimal part spoken as number
NDF9	Decimal number — 9 decimal places, decimal part spoken as number
N$	Currency — same as N$D2
N$D2	Currency — francs and centimes
N$F	Currency — whole francs
N$F0	Currency — francs with no decimal places
N$F1	Currency — francs with 1 decimal place
N$F2	Currency — francs with 2 decimal places
N$F3	Currency — francs with 3 decimal places
N$F4	Currency — francs with 4 decimal places
N$F5	Currency — francs with 5 decimal places
N$F6	Currency — francs with 6 decimal places
N$F7	Currency — francs with 7 decimal places
N$F8	Currency — francs with 8 decimal places
N$F9	Currency — francs with 9 decimal places
N$EU	Currency — same as N$D2EU
N$D2EU	Currency — Euros and centimes
N$FEU	Currency — whole Euros
N$F0EU	Currency — whole Euros
N$F1EU	Currency — Euros with 1 decimal place
N$F2EU	Currency — Euros with 2 decimal places
N$F3EU	Currency — Euros with 3 decimal places
N$F4EU	Currency — Euros with 4 decimal places
N$F5EU	Currency — Euros with 5 decimal places
N$F6EU	Currency — Euros with 6 decimal places
N$F7EU	Currency — Euros with 7 decimal places
N$F8EU	Currency — Euros with 8 decimal places
N$F9EU	Currency — Euros with 9 decimal places
C	Characters — same as Cmmm
Crmf	Characters — spoken individually, total inflection

2 of 3

Table 27. French EBS Formats

Format	Description
Crmm	Characters — spoken individually, rising inflection
Cmmf	Characters — spoken individually, falling inflection
Cmmm	Characters — spoken individually, medial inflection
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).
	3 of 3

German EBS Formats

Table 28. German EBS Formats

Format	Description
D	Date — same as DMSPDYY
DMSPDYY	Date — spoken month, 2-digit day, 4-digit year
DMSPDY	Date — spoken month, 2-digit day and year
DMSPD	Date — spoken month, 2-digit day
T	Time — same as TH24AM
TH24M	Time — 24-hour clock, no seconds
N	Number — same as NN
NM	Number — male form
NF	Number — female form
NN	Number — neutral form
ND0	Decimal number — no decimal places
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND3	Decimal number — 3 decimal places t
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
N$	Currency — same as N$D0
N$D0	Currency — whole marks
N$D2	Currency — marks and pfennig
N$DM	Currency — whole Deutsch marks
N$DM0	Currency — whole Deutsch marks
	1 of 2

Table 28. German EBS Formats

Format	Description
N$DM2	Currency — Deutsch marks and pfennig
N$EU	Currency — whole Euros
N$D0EU	Currency — whole Euros
N$D2EU	Currency — Euros and cents
C	Characters — same as Cmmm
Crmf	Characters — spoken individually, total inflection
Crmm	Characters — spoken individually, rising inflection
Cmmf	Characters — spoken individually, falling inflection
Cmmm	Characters — spoken individually, medial inflection
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).
	2 of 2

Hindi EBS Formats

Table 29. Hindi EBS Formats

Format	Description
D	Date — same as DDMSPYY
DDMSP	Date — 2-digit day, spoken month
DDMSPY	Date — 2-digit day, spoken month, 4-digit year
T	Time —same as THMAM
THMAM	Time — time period followed by time
N	Number — integer, medial inflection (default)
ND0	Decimal number — no decimal place
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
N$	Currency — same as N$D2
N$D0	Currency — whole dollars
N$D1	Currency — dollars with 1 decimal place
	1 of 2

Table 29. Hindi EBS Formats

Format	Description
N$D2	Currency — dollars and cents
N$D3	Currency — dollars with 3 decimal places
N$D4	Currency — dollars with 4 digits decimal places
N$D5	Currency — dollars with 5 digits decimal places
N$D6	Currency — dollars with 6 digits decimal places
N$D7	Currency — dollars with 7 digits decimal places
N$D8	Currency — dollars with 8 decimal places
N$D9	Currency — dollars with 9 with decimal places
NRD0	Currency — whole rupees
NRD1	Currency — rupees with 1 decimal place
NRD2	Currency — rupees with 2 decimal places
NRD3	Currency — rupees with 3 decimal places
NDR4	Currency — rupees with 4 decimal places
NDR5	Currency — rupees with 5 decimal places
NDR6	Currency — rupees with 6 decimal places
NDR7	Currency — rupees with 7 decimal places
NDR8	Currency — rupees with 8 decimal places
NDR9	Currency — rupees with 9 decimal places
C	Characters — same as Cmmm
Cmmm	Characters — spoken individually, medial inflection
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).
	2 of 2

Hungarian EBS Formats

Table 30. Hungarian EBS Formats

Format	Description
D	Date — same as YYMSPDD
YYMSPDD	Date — 4-digit year, spoken month and day
MSPDD	Date — no year, spoken month and day
YYMSPDDf	Date — 4-digit year, spoken month and day, falling inflection
MSPDDf	Date — no year, spoken month and day, falling inflection
T	Time — same as TH24M
TH24M	Time — hour, minute, 24-hour clock
TH24MS	Time — hour, minute, second, 24-hour clock
TH24Mf	Time — hour, minute, 24-hour clock, falling inflection
TH24MSf	Time — hour, minute, second, 24-hour clock, falling inflection
N	Number — same as NM2m
NMm	Number — medial inflection
NMf	Number — falling inflection
NM2m	Number — alternate NMm format, medial inflection
NM2f	Number — alternate NMf format, falling inflection
ND0	Decimal number — whole number spoken as “number comma 0”, no decimal places
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
NDX	Decimal number — variable number of decimal places, decimal separator in argument NOTE: A number passed to a function can contain either a period (.) or a comma (,) to identify a number of decimal digits spoken for that number.
NDX1	Decimal number — 1 decimal place, spoken as a fraction
NDX2	Decimal number — 2 decimal places, spoken as a fraction
NDX3	Decimal number — 3 decimal places, spoken as a fraction
NDX0S	Decimal number — 3 or fewer decimal places, spoken as a whole number with padding zeros
NDXS	Decimal number — 3 or fewer decimal places, spoken as a fraction; otherwise NDX
ND0f	Number — whole number, spoken as “number comma 0”, falling inflection

1 of 2

Table 30. Hungarian EBS Formats

Format	Description
ND1f	Decimal number — 1 decimal place, falling inflection
ND2f	Decimal number — 2 decimal places, falling inflection
ND3f	Decimal number — 3 decimal places, falling inflection
ND4f	Decimal number — 4 decimal places, falling inflection
ND5f	Decimal number — 5 decimal places, falling inflection
ND6f	Decimal number — 6 decimal places, falling inflection
ND7f	Decimal number — 7 decimal places, falling inflection
ND8f	Decimal number — 8 decimal places, falling inflection
ND9f	Decimal number — 9 decimal places, falling inflection
NDXf	Decimal number — variable number of decimal places, spoken as a whole number with padding zeros, falling inflection
NDX1f	Decimal number — 1 decimal place, spoken as a fraction, falling inflection
NDX2f	Decimal number — 2 decimal places, spoken as a fraction, falling inflection
NDX3f	Decimal number — 3 decimal places, spoken as a fraction, falling inflection
NDXSf	Decimal number — 3 or fewer decimal places, spoken as a fraction, falling inflection; otherwise NDX
NDX0Sf	Decimal number — 3 or fewer decimal places, spoken as a fraction, falling inflection; otherwise NDX
N$	Currency — same as N$D0
N$D0	Currency — whole forints
N$D2	Currency — forints and filers
N$DX	Currency — forints as a decimal number, variable number of decimal places
C	Characters — same as Cmm
Cspd	Characters — spoken individually, variable pace
Cmf	Characters — spoken individually, falling inflection
Cmm	Characters — spoken individually, medial inflection
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).
	2 of 2

Indonesian Enhanced Basic Speech Formats

Table 31. Indonesian EBS Formats

Format	Description
N	Number — integer, medial inflection (default)
ND0	Decimal number — no decimal place
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places, numbers after decimal point spoken digit-by-digit
ND2W	Decimal number — 2 digits after decimal point spoken as number
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
D	Date — same as DMSPDYY
DMSPDYY	Date — spoken month, 2-digit day, 4-digit year
T	Time — same as THMAM
THMAM	Time — hour, minute, AM/PM
TH24M	Time — hour, minute, 24 hour clocks
N$	Currency — same as N$R
N$R	Currency — whole Indonesian rupiah
N$D0	Currency — whole US dollars
N$D2	Currency — US dollars and cents
C	Characters — same as CIm
Cm	Characters — English, spoken individually, medial inflection
Cf	Characters — English, spoken individually, falling inflection
Cr	Characters — English, spoken individually, rising inflection
CIm	Characters — Indonesian, spoken individually, medial inflection
CIf	Characters — Indonesian, spoken individually, falling inflection
CIr	Characters — Indonesian, spoken individually, rising inflection
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).

Italian Enhanced Basic Speech Formats

Table 32. Italian EBS Formats

Format	Description
D	Date — same as DDMSPYYf
DDMSPYY	Date — day, spoken month, 4-digit year
DDMSPY	Date — day, spoken month, 2-digit year
DDMSP	Date — spoken month, 2-digit day, no year
DDMSPYYf	Date — same as DDMSPYY but with falling inflection
DDMSPYf	Date — same as DDMSPY but with falling inflection
DDMSPf	Date — same as DDMSP but with falling inflection
T	Time — same as TH24Mf
THMAM	Time — hour, minute, AM/PM
TH24M	Time — hour, minute, 24-hour clock
THMAMf	Time — hour, minute, AM/PM, falling inflection
TH24Mf	Time — hour, minute, 24-hour clock, falling inflection
N	Number — same as Nrm
Nmf	Number — falling inflection
Nrm	Number — rising inflection
Nrmf	Number — total inflection
NN	Number — “uno”
NF	Number — “una”
NM	Number — “un”
ND0	Decimal number — no decimal places
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
N$	Currency — lire, medial inflection (default)
N$f	Currency — lire, falling inflection
N$EU	Currency — Euros and cents, medial inflection
N$fEU	Currency — Euros and cents, falling inflection

1 of 2

Table 32. Italian EBS Formats

Format	Description
C	Characters — same as Cmmm
Crmf	Characters — spoken individually, total inflection
Crmm	Characters — spoken individually, rising inflection
Cmmf	Characters — spoken individually, falling inflection
Cmmm	Characters — spoken individually, medial inflection
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).
	2 of 2

Japanese Enhanced Basic Speech Formats

Table 33. Japanese EBS Formats

Format	Description
D	Date — same as DSPYYMD
DSPYYMD	Date — spoken 4-digit year, month, and day
DSPYYMDW	Date — spoken 4-digit year, month, day, and day of the week
DSPYMD	Date — last 2 digits of year spoken, month, and day
DSPYMDW	Date — last 2 digits of year spoken, month, day, and day of the week
DSPMD	Date — spoken month and day
DSPMDW	Date — spoken month, day, and day of the week
T	Time — same as THMAM
THMAM	Time — hour, minute, AM/PM
N	Number — integer, total inflection (default)
ND0	Decimal number — no decimal places
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
N$	Currency — whole yen (default)
N$D2	Currency — yen, 2 decimal places
C	Characters — spoken individually, medial inflection
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).

Korean Enhanced Basic Speech Formats

Table 34. Korean EBS Formats

Format	Description
D	Date — same as DYYMD
DYYMD	Date — 4-digit year, month, day
DMD	Date — month followed by day
T	Time — same as TAMHM
TAMHM	Time — time period followed by time
N	Number — integer, cardinal number (default)
NT	Number — ordinal
ND0	Decimal number — no decimal places
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
N$	Currency — same as N$D0
N$D0	Currency — whole dollars
N$D2	Currency — dollars and cents
N$D1	Currency — dollars with 1 decimal place
N$D3	Currency — dollars with 3 decimal places
N$D4	Currency — dollars with 4 decimal places
N$D5	Currency — dollars with 5 decimal places
N$D6	Currency — dollars with 6 decimal places
N$D7	Currency — dollars with 7 decimal places
N$D8	Currency — dollars with 8 decimal places
N$D9	Currency — dollars with 9 decimal places
NW	Currency — whole won
C	Character — same as Cmmm
Cmmm	Characters — spoken individually, medial inflection
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).

Latin-American Spanish EBS Formats

Table 35. Latin-American Spanish EBS Format Set

Format	Description
D	Date — same as DMSPDYY
DMSPDYY	Date — spoken month, 2-digit day, 4-digit year
DMSPDY	Date — spoken month, 2-digit day, 2-digit year
DMSPD	Date — spoken month, 2-digit day
T	Time — same as THMAM
THMAM	Time — hour, minute, AM/PM
TH24M	Time —hour, minute, 24-hour clock
N	Number — same as NN
NM	Number — male form
NF	Number — female form
NN	Number — neutral form
ND0	Decimal number — no decimal places
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
C	Character — spoken individually (default)
N$	Currency — same as N$D0
N$D0	Currency — whole pesos
N$D2	Currency — pesos and centavos
NDOL$	Currency — whole dollars
NDOL$D0	Currency — whole dollars
NDOL$D2	Currency — dollars and centavos
NPN$	Currency — whole pesos nuevos
NPN$D0	Currency — whole pesos nuevos
NPN$D2	Currency — pesos nuevos and centavos
NCOL$	Currency — whole colones
NCOL$D0	Currency — whole colones

1 of 2

Table 35. Latin-American Spanish EBS Format Set

Format	Description
NCOL$D2	Currency — colones and centavos
NSUC$	Currency — whole sucres
NSUC$D0	Currency — whole sucres
NSUC$D2	Currency — sucres and centavos
NQUE$	Currency — whole quetzales
NQUE$D0	Currency — whole quetzales
NQUE$D2	Currency — quetzales and centavos
NLEM$	Currency — whole lempiras
NLEM$D0	Currency — whole lempiras
NLEM$D2	Currency — lempiras and centavos
NCOR$	Currency — whole cordobas
NCOR$D0	Currency — whole cordobas
NCOR$D2	Currency — cordobas and centavos
NGUA$	Currency — whole guaranis
NGUA$D0	Currency — whole guaranis
NGUA$D2	Currency — guaranis and centimos
NSOL$	Currency — whole soles
NSOL$D0	Currency — whole soles
NSOL$D2	Currency — soles and centavos
NBOL$	Currency — whole bolivares
NBOL$D0	Currency — whole bolivares
NBOL$D2	Currency — bolivares and centimos
NPTA$	Currency — whole pesetas
NPTA$D0	Currency — whole pesetas
NPTA$D2	Currency — pesetas and centimos
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).

2 of 2

Malay Enhanced Basic Speech Formats

Table 36. Malay EBS Formats

Format	Description
D	Date — same as DMSPDYY
DMSPDYY	Date — spoken month, 2-digit day, 4-digit year
T	Time — same as THMAM
THMAM	Time — hour, minute, AM/PM
TH24M	Time — hour, minute, 24-hour clock
N	Number — integer (default)
ND0	Decimal number — no decimal places
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND2W	Decimal number — 2 digits after decimal point spoken as number
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
N$	Currency — same as NR$
N$R	Currency — whole Malaysian ringgits
N$R2	Currency — Malaysian ringgits and cents
N$D0	Currency — whole US dollars
N$D2	Currency — US dollars and cents
C	Characters — same as CIm
Cm	Characters — English, spoken individually, medial inflection
Cf	Characters — English, spoken individually, falling inflection
Cr	Characters — English, spoken individually, rising inflection
CIm	Characters — Same as Cm
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).

Mandarin Chinese Enhanced Basic Speech Formats

Table 37. Mandarin EBS Formats

Format	Description
D	Date — same as DYYMD
DYYMD	Date — 4-digit year, month and day
DMD	Date — month followed by day
DTWYYMD	Date — 4-digit year per Taiwanese Republic calendar, month, day
T	Time — same as THMAM
THMAM	Time — hour, minute, 24-hour clock
N	Number — integer, with digit 2 spoken as "ur" (default)
N2	Number — integer, with digit 2 spoken as "lyang"
ND0	Decimal number — no decimal places
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
N$	Currency — same as NYD0
N$D0	Currency — whole dollars
N$D1	Currency — dollars with 1 decimal place
N$D2	Currency — dollars and cents
N$D3	Currency — dollars with 3 decimal places
N$D4	Currency — dollars with 4 decimal places
N$D5	Currency — dollars with 5 decimal places
N$D6	Currency — dollars with 6 decimal places
N$D7	Currency — dollars with 7 decimal places
N$D8	Currency — dollars with 8 decimal places
N$D9	Currency — dollars with 9 decimal places
NYD0	Currency — whole yuan
NYD1	Currency — yuan with 1 decimal place
NYD2	Currency — yuan, chiao, and fun
NYD3	Currency — yuan with 3 decimal places

1 of 2

Table 37. Mandarin EBS Formats

Format	Description
NYD4	Currency — yuan with 4 decimal places
NYD5	Currency — yuan with 5 decimal places
NYD6	Currency — yuan with 6 decimal places
NYD7	Currency — yuan with 7 decimal places
NYD8	Currency — yuan with 8 decimal places
NYD9	Currency — yuan with 9 decimal places
NQD0	Currency — whole quai
NQD1	Currency — quai with 1 decimal place
NQD2	Currency — quai with 2 decimal places
NQD3	Currency — quai with 3 decimal places
NQD4	Currency — quai with 4 decimal places
NQD5	Currency — quai with 5 decimal places
NQD6	Currency — quai with 6 decimal places
NQD7	Currency — quai with 7 decimal places
NQD8	Currency — quai with 8 decimal places
NQD9	Currency — quai with 9 decimal places
C	Characters — same as Cmmm
Cmmm	Characters — spoken individually
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).
	2 of 2

Polish EBS Formats

Table 38. Polish EBS Formats

Format	Description
D	Date — same as DDMSPYY
DDMSPYY	Date — day, spoken month, 4-digit year
DMMSP	Date — day, spoken month, no year
DDMSPYYf	Date — day, spoken month, 4-digit year, falling inflection
DMMSPf	Date — day and month, no year, falling inflection
T	Time — same as TH24M
TH24M	Time — hour and minute as numbers, 24-hour clock
TSPH24M	Time — spoken hour and minute, 24-hour clock
TSPH24MS	Time — spoken hour, minute, and second, 24-hour clock
TOH24M	Time — hour and minute in form "at time"
	1 of 3

Table 38. Polish EBS Formats

Format	Description
TH24Mf	Time — hour and minute as numbers, falling inflection
TSPH24Mf	Time — spoken hour and minute, falling inflection
TSPH24MSf	Time — spoken hour, minute, and second, falling inflection
TOH24Mf	Time — hour and minute in form "at time," falling inflection
N	Number —same as NMm
NMm	Number — up to 1 billion, male form, medial inflection
NMf	Number — up to 1 billion, male form, falling inflection
NFm	Number — up to 1 billion, female form, medial inflection
NFf	Number — up to 1 billion, female form, falling inflection
NNm	Number — up to 1 billion, neutral form, medial inflection
NNf	Number — up to 1 billion, neutral form, falling inflection
ND0	Decimal number — whole number spoken as "number comma 0"
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
NDX1	Decimal number — 1 decimal place, spoken as a fraction
NDX2	Decimal number — 2 decimal places, spoken as a fraction
NDX3	Decimal number — 3 decimal places, spoken as a fraction
NDX	Decimal number — variable number of decimal places, decimal separator in argument NOTE: A number passed to a function can contain either a period (.) or a comma (,) to identify a number of decimal digits spoken for that number.
NDX0S	Decimal number — 3 or fewer decimal places, spoken as a number with leading zeros; otherwise NDX
NDXS	Decimal number — 3 or fewer decimal places, spoken as a fraction; otherwise NDX
ND0f	Decimal number — whole number spoken as "number comma 0", falling inflection
ND1f	Decimal number — 1 decimal place, falling inflection
ND2f	Decimal number —- 2 decimal places, falling inflection
ND3f	Decimal number — 3 decimal places, falling inflection
ND4f	Decimal number — 4 decimal places, falling inflection
ND5f	Decimal number — 5 decimal places, falling inflection

2 of 3

Table 38. Polish EBS Formats

Format	Description
ND6f	Decimal number — 6 decimal places, falling inflection
ND7f	Decimal number — 7 decimal places, falling inflection
ND8f	Decimal number — 8 decimal places, falling inflection
ND9f	Decimal number — 9 decimal places, falling inflection
NDX1f	Decimal number — 1 decimal place, spoken as a fraction, falling inflection
NDX2f	Decimal number — 2 decimal places, spoken as a fraction, falling inflection
NDX3f	Number — 3 decimal places, spoken as a fraction, falling inflection
NDXf	Decimal number — variable number of decimal places, falling inflection, decimal separator in argument NOTE: A number passed to a function can contain either a period (.) or a comma (,) to identify a number of decimal digits spoken for that number)
NDX0Sf	Decimal number — 3 or fewer decimal places, spoken as number with leading zeros; otherwise NDX
NDXSf	Decimal number — 3 or fewer decimal places, spoken as a fraction, falling inflection; otherwise NDX
N$	Currency — same as N$D2f
N$D0	Currency — whole zlote, up to 1 billion, medial inflection)
N$D2	Currency — zlote and grosze, up to 1 billion, medial inflection
N$D2f	Currency — zlote and grosze, up to 1 billion, falling inflection
N$DX	Currency — zlote, up to 1 billion, decimal form
N$EU	Currency — Euros and cents, up to 1 billion, falling inflection
N$D0EU	Currency — Euros, up to 1 billion, medial inflection
N$D2EU	Currency — Euros and cents, up to 1 billion, medial inflection
N$D2fEU	Currency — Euros and cents, up to 1 billion, falling inflection
N$DXEU	Currency — Euros, up to 1 billion, decimal form
C	Characters — same as Cmm
Cspd	Characters — spoken individually, variable pace, medial inflection
Cmf	Characters — spoken individually, falling inflection
Cmm	Characters — spoken individually, medial inflection
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).
	3 of 3

Slovak EBS Formats

Table 39. Slovak EBS Formats

Format	Description
D	Date — same as DDMSPYY
DDMSPYY	Date — day, spoken month, 4-digit year
DDMYY	Date — day, month, 4-digit year
DMMSP	Date — day, spoken month, no year
DDM	Date — day, month, no year
DDMSPYYf	Date — day, spoken month, 4-digit year, falling inflection
DDMYYf	Date — day, month, 4-digit year, falling inflection
DMMSPf	Date — day, spoken month, falling inflection
DDMf	Date — day, month, falling inflection
T	Time — same as TH24M
TH24M	Time — hour, minute, 24-hour clock
TH24MS	Time — hour, minute, second, 24-hour clock
TH24Mf	Time — hour, minute, falling inflection
TH24MSf	Time — hour, minute, second, falling inflection
N	Number — same as NMm
NMm	Number — male form, medial inflection
NMLm	Number — male form, for animate subjects, medial inflection
NMf	Number — male form, falling inflection
NMLf	Number — male form, for animate subjects, falling inflection
NFm	Number — female form, medial inflection
NFf	Number — female form, falling inflection
NNm	Number — neutral form, medial inflection
NNf	Number — neutral form, falling inflection
ND0	Decimal number — whole number spoken as "number comma 0"
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal place
ND9	Decimal number — 9 decimal place
	1 of 2

Table 39. Slovak EBS Formats

Format	Description
NDX1	Decimal number— 1 decimal place, spoken as a fraction
NDX2	Decimal number — 2 decimal places, spoken as a fraction
NDX3	Decimal number — 3 decimal places, spoken as a fraction
NDX	Decimal number — variable number of decimal places, decimal separator in argument NOTE: A number passed to a function can contain either a period (.) or a comma (,) to identify a number of decimal digits spoken for that number.
NDXS	Decimal number — 3 or fewer decimal places spoken as fraction, falling inflection; otherwise NDX
ND0f	Decimal number — whole number spoken as "number comma 0", falling inflection
ND1f	Decimal number — 1 decimal place, falling inflection
ND2f	Decimal number — 2 decimal places, falling inflection
ND3f	Decimal number — 3 decimal places, falling inflection
ND4f	Decimal number — 4 decimal places, falling inflection
ND5f	Decimal number — 5 decimal places, falling inflection
ND6f	Decimal number — 6 decimal places, falling inflection
ND7f	Decimal number — 7 decimal places, falling inflection
ND8f	Decimal number — 8 decimal places, falling inflection
ND9f	Decimal number — 9 decimal places, falling inflection
NDX1f	Decimal number — 1 decimal place, spoken as a fraction, falling inflection
NDX2f	Decimal number — 2 decimal places, spoken as a fraction, falling inflection
NDX3f	Decimal number — 3 decimal places, spoken as a fraction, falling inflection
NDXf	Decimal number — variable number of decimal places, falling inflection, decimal separator in argument NOTE: A number passed to a function can contain either a period (.) or a comma (,) to identify a number of decimal digits spoken for that number.
NDXSf	Decimal number — 3 or fewer decimal places, spoken as a fraction, falling inflection; otherwise NDX
N$	Currency — same as N$D0
N$D0	Currency — whole crowns
N$D2	Currency — crowns and hellers
N$DX	Currency — crowns spoken as a decimal number, variable number of decimal places
N$EU	Currency — same as N$D0EU
N$D0EU	Currency — whole Euros
N$D2EU	Currency — Euros and cents
N$DXEU	Currency — Euros as decimal number with a variable number of decimal places
C	Characters — same as Cmm
Cmf	Characters — spoken individually, falling inflection
Cmm	Characters — spoken individually, medial inflection
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).
	2 of 2

Thai EBS Formats

Table 40. Thai EBS Formats

Format	Description
D	Date — same as DBY
DWY	Date — Western year, spoken as a whole number
DWYD	Date — Western year, spoken digit by digit
DWYS	Date — Western year, spoken in short form
DBY	Date — Buddha year, spoken as a whole number
DBYD	Date — Buddha year, spoken digit by digit
DBYS	Date — Buddha year, spoken in short form
T	Time — same as THMAM
THMAM	Time — hour, minute, AM/PM
TH24M	Time — hour, minute, 24-hour clock
N	Number — integer (default)
ND0	Decimal number — no decimal places
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places
ND8	Decimal number — 8 decimal places
ND9	Decimal number — 9 decimal places
N$	Currency — same as N$T0
N$T0	Currency — whole Thai baht
N$T2	Currency — Thai baht and satang
N$D0	Currency — whole US dollars
N$D2	Currency — US dollars and cents
C	Characters — same as Cm
Cm	Characters — English, spoken individually, medial inflection
Cf	Characters — English, spoken individually, falling inflection
Cr	Characters — English, spoken individually, rising inflection
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).

UK English EBS Formats

Table 41. UK English EBS Formats

Format	Description
D	Date — same as DDMSPYY
DDMSP	Date — day and month
DDMSPY	Date — day, month, 2-digit year
DDMSPYY	Date — day, month, 4-digit year
DDMSPf	Date — day and month, falling inflection
DDMSPYf	Date — day, month, 2-digit year, falling inflection
DDMSPYYf	Date — day, month, 4-digit year, falling inflection
DDMSPO	Date — "on" day, month
DDMSPYO	Date — "on" day, month, 2-digit year
DDMSPYYO	Date — "on" day, month, 4-digit year
DDMSPOf	Date — "on" day, month, falling inflection
DDMSPYOf	Date — "on" day, month, 2-digit year, falling inflection
DDMSPYYOf	Date — "on" day, month, 4-digit year, falling inflection
DMDYY	Date — same as DDMSPYY
DMDY	Date — same as DDMSPY
DMD	Date — same as DDMSP
DMSPDYY	Date — same as DDMSPYY
DMSPDY	Date — same as DDMSPY
DMSPD	Date — same as DDMSP
T	Time — same as THM24
THMAM	Time — hour, minute, AM/PM
THMAMf	Time — hour, minute, AM/PM, falling inflection
THM24	Time — hour, minute, 24-hour clock
THM24f	Time — hour, minute, 24-hour clock, falling inflection
N	Number — same as Nrmm
Nrmf	Number — integer, total inflection
Nrmm	Number — integer, rising inflection
Nmmf	Number — same as Nrmm
Nmmm	Number — same as Nrmm
ND	Decimal number — same as ND0
ND0	Decimal number — no decimal places
ND1	Decimal number — one decimal place
ND2	Decimal number — two decimal places
	1 of 2

Table 41. UK English EBS Formats

Format	Description
ND3	Decimal number — three decimal places
ND4	Decimal number — four decimal places
ND5	Decimal number — five decimal places
ND6	Decimal number — six decimal places
ND7	Decimal number — seven decimal places
ND8	Decimal number — eight decimal places
ND9	Decimal number — nine decimal places
N$	Currency — pounds and pence, medial inflection
N$f	Currency — pounds and pence, falling inflection
N$EU	Currency — Euros and cents, medial inflection
N$fEU	Currency — Euros and cents, falling inflection
N$D0	Currency — pounds
N$D1	Currency — pounds with one decimal place
N$D2	Currency — pounds with two decimal places
N$D3	Currency — pounds with three decimal places
N$D4	Currency — pounds with four decimal places
N$D5	Currency — pounds with five decimal places
N$D6	Currency — pounds with six decimal places
N$D7	Currency — pounds with seven decimal places
N$D8	Currency — pounds with eight decimal places
N$D9	Currency — pounds with nine decimal places
C	Characters — same as Cmmm
Cmmm	Characters — spoken individually, medial inflection
Cmmf	Characters — spoken individually, falling inflection
Crmf	Characters — same as Cmmf
Crmm	Characters — same as Cmmm

2 of 2

US English EBS Formats

Table 42. US English EBS Formats

Format	Description
D	Date — same as DMSPDY
DMDY	Date — 2-digit month, day, year
DMDYY	Date — 2-digit month, day, 4-digit year
DMD	Date — 2-digit month, day
DMSPDYY	Date — spoken month, ordinal day, 4-digit year
DMSPDY	Date — spoken month, ordinal day, 2-digit year
DMSPD	Date — spoken month, ordinal day
T	Time — same as THMAM
THMAM	Time — hour, minute, AM/PM
N	Number — same as Nmmm
Nrmf	Number — total inflection
Nmmf	Number — falling inflection
Nrmm	Number — rising inflection
Nmmm	Number — medial inflection
ND	Decimal number — same as ND0
ND0	Decimal number - no decimal place
ND1	Decimal number — 1 decimal place
ND2	Decimal number — 2 decimal places
ND3	Decimal number — 3 decimal places
ND4	Decimal number — 4 decimal places
ND5	Decimal number — 5 decimal places
ND6	Decimal number — 6 decimal places
ND7	Decimal number — 7 decimal places,
ND8	Decimal number — 8 decimal places,
ND9	Decimal number — 9 decimal places,
N$	Currency — same as N$D0
N$D0	Currency — whole dollars
N$D1	Currency — dollars with 1 decimal place
N$D2	Currency — dollars "and" cents
N$D3	Currency — dollars with 3 decimal places
N$D4	Currency — dollars with 4 decimal places
N$D5	Currency — dollars with 5 decimal places
N$D6	Currency — dollars with 6 decimal places

1 of 2

Table 42. US English EBS Formats

Format	Description
N$D7	Currency — dollars with 7 decimal places
N$D8	Currency — dollars with 8 decimal places
N$D9	Currency — dollars with 9 decimal places
C	Characters — same as Cmmm
Crmf	Characters — spoken individually, total inflection
Crmm	Characters — spoken individually, rising inflection
Cmmf	Characters — spoken individually, falling inflection
Cmmm	Characters — spoken individually, medial inflection
NX	See Speaking Phrase Numbers and Packed Talkfile Numbers (on page 102).

2 of 2

B Speech File Formats

Overview

This appendix describes the encoding formats for speech files.

Topics covered include:

- Speech File Formats (on page 139)
  - ~ Speech Formats Supported by CONVERSANT Systems (on page 139)
  - ~ Converting *.wav Files to or from CONVERSANT-Supported Formats (on page 141)
- Speech Files (on page 146)

Speech File Formats

Speech Formats Supported by CONVERSANT Systems

Before a speech phrases can be digitized and stored, it must be converted to a supported format. This is accomplished by a process known as encoding speech. The CONVERSANT system supports the following encoding formats for speech phrases:

- PCM64 (pulse code modulation at 64 Kbps in Mu-law encoding format)
- ADPCM32 (adaptive differential pulse code modulation at 32 Kbps)
- ADPCM16 (adaptive differential pulse code modulation at 16 Kbps — not commonly used in applications)
- CELP16 (code excited linear prediction at 16 Kbps)

PCM Encoding Formats

In a PCM speech file, speech is sampled at 8000 times a second. Each sample is digitally coded into an 8-bit pattern (allowing 256 levels), resulting in 64,000 bits per second of speech. The header, which has the values `0xAA 0xFF 0x34 0x00`, is repeated at least 10 times for every second of speech (every 800 bytes). If the value `0xAA 0xFF` occurs anywhere within the speech portion of the file, it is changed to `0xAA 0xFE`. The speech data consists of contiguous 8-bit patterns representing sampled speech. One type of PCM format is available, Mu-law. In the Mu-law PCM format, PCM coding uses predefined quasi-logarithmic steps for speech levels, encoding more steps when the speech level is low.

ADPCM Encoding Formats

ADPCM speech encoding (which includes is based on the principle that it is possible to reduce the amount of information that is 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.

CELP16

CELP16 requires about half the disk space as ADPCM32 files and helps improve system performance by reducing the amount of data transferred between disk and memory. It is often most advantageous with systems where there are limited CPU or disk resources due to speech-related activities.

In addition, it often produces a better-sounding output than ADPCM32. It may, however, place a greater demand on SSP circuit cards. Before converting to and using this format, verify that you have sufficient SSP resources to support CELP16 playback and encoding.

Converting *.wav Files to or from CONVERSANT-Supported Formats

Avaya provides, as part of the standard software package, two software utilities available to assist you in converting sound files from one format to another:

- **wav2vis** converts Windows *.wav files to speech formats supported by CONVERSANT systems.
- **vis2wav** converts speech files from formats supported by CONVERSANT systems to Windows *.wav files.

These tools are located in the **/mtce/bin** directory.

⚠ **CAUTION:**
Because of the intense processing demands the **wav2vis** conversion tool places on systems while files are being converted, especially when converting to CELP16 format, it is recommended that you *not* use it on live production systems. Using it on live systems can have serious impacts on system performance. If runtime conversions are needed, it is recommended that you perform the file conversion on a separate UnixWare or Windows system and then transfer the converted file(s) to the desired CONVERSANT system.

Converting *.wav Files to CONVERSANT-Supported Formats

The **wav2vis** utility is a standalone tool which converts Windows *.wav files to encoded speech file formats that can be played by CONVERSANT Speech Signal and Processing (SSP) and Analog (AYC10) circuit cards. These formats include:

- PCM64
- ADPCM32
- CELP16

Note: There is a corresponding version of this tool, **wav2vis.exe**, which is designed to run on Windows 9x or NT systems. For more information about this tool, see Converting Speech Files on Windows Systems below.

Table 43 on page 142 lists the criteria that *.wav files must meet in order to be converted using the **wav2vis** tool.

Table 43. Criteria for Converting *.wav Files to CONVERSANT-Supported Formats

Format	PCM
Bits Per Sample	8 or 16
Possible Sample Rates (in KHz)	• 8.000 • 11.025 • 22.050 • 44.100
Channels	• Mono • Stereo **Note:** Only one channel of stereo is converted.

To use the **wav2vis** tool, do the following:

1 Copy the *.wav file to be converted to the CONVERSANT system.

2 At the UnixWare system prompt, enter some form of the following command (options are described in Table 44):

wav2vis [-l] [-n] [-e *type*] [-s *num*] *wavefile visfile*

Table 44. Options for wav2vis Conversion Tool

Option	Comments/Description
-l	This option causes *.wav file information to be printed.
-n	This option prevents speech headers from being written to the resulting *.vis file
-e *type*	This option defines the speech encoding type. Possible type values include **pcm64**, **adpcm32**, and **celp16**. Default = **adpcm32**
-s *num*	This option allows you to adjust the volume level of the converted file. *num* = the scaling factor. A value of 1 results in no change to the volume. A value greater than 1 increases the output volume. A value less than 1 decreases the output volume. Default = **1**

1 of 2

Table 44. Options for wav2vis Conversion Tool

Option	Comments/Description
wavfile	This is the directory path and name of the original *.wav file. As an option, you can enter a value of **-** (hyphen) to read from standard input.
visfile	This is the name and directory path for the CONVERSANT speech file to be created. As an option, you can enter a value of **-** (hyphen) to write to standard output.

2 of 2

Example:

Suppose you had a *.wav file named **greeting.wav** that you wanted to convert to CELP16 format, while increasing the volume slightly. You want to place the converted file in the /tmp/ directory. To accomplish this, you would enter the following at the system prompt:

```
wav2vis -s 1.2 -e celp16 greeting.wav /tmp/greeting.vis
```

Converting CONVERSANT-Supported Speech Files to *.wav Files

The **vis2wav** utility is a standalone tool which converts CONVERSANT-supported speech files to *.wav files that can be played in a Windows 9x or NT operating system environment. Formats which can be converted include:

- PCM64
- ADPCM32

Note: CELP16 file conversion is not supported in the current version of this tool.

The resulting file is an 8 KHz PCM *.wav file. By default, **vis2wav** creates an 8-bit file, though as an option, you can use it to create a 16-bit file.

Note: There is a corresponding version of this tool, **vis2wav.exe**, which is designed to run on Windows 9x or NT systems. For more information about this tool, see Converting Speech Files on Windows Systems below.

To use the **vis2wav** tool, do the following:

1. Change directories to the one in which the desired desired speech file resides.
2. At the UnixWare system prompt, enter some form of the following command (options are described in Table 45 on page 144):

   vis2wav [-l] [-e *type*] [-s *num*] *visfile wavefile*

Table 45. Options for vis2wav Conversion Tool

Option	Comments/Description
-l	This option causes file conversion information to be printed.
-e *type*	This option defines what type of PCM file you want to create. Possible type values include 8-bit or 16-bit. Default = **8**
-s *num*	This option allows you to adjust the volume level of the converted file. *num* = the scaling factor. A value of 1 results in no change to the volume. A value greater than 1 increases the output volume. A value less than 1 decreases the output volume. Default = **1**
visfile	This is the name of the CONVERSANT speech file to be converted. As an option, you can enter a value of **-** (hyphen) to read from standard input.
wavfile	This is the name of the original *.wav file. As an option, you can enter a value of **-** (hyphen) to write to standard output.

Example:

Suppose you had a CONVERSANT speech file named **1234** residing in the /talkfiles/111 directory that you wanted to convert to a 16-bit PCM *.wav format, while decreasing the volume slightly. You want to place the resultant file in the /tmp directory. To accomplish this, you would enter the following at the system prompt:

```
vis2wav -s 0.8 -e 16 /home2/vfs/talkfiles/111/1234
/tmp/greeting.wav
```

Note: While it is possible to use the **vis2wav** tool in conjunction with the **wav2vis** tool to convert encoded speech files from one format to another (for example, from ADPCM32 to *.wav to CELP16), it is not recommended that you use these tools for that purpose. With every conversion, there is a certain amount of loss in the resulting encoded speech, and using these tools in that fashion can cause a noticeable degradation in the quality of the output.

Converting Speech Files on Windows Systems

The **wav2vis** and **vis2wav** tools both have versions designed to run in Windows 9x or NT environments and are named **wav2vis.exe** and **vis2wav.exe**, respectively. These tools are also provided as part of the standard CONVERSANT software package. They are located in the **/mtce/bin** directory of the CONVERSANT system. To transfer them to the Windows PC, you can use either FTP or floppy transfer.

To help you create a floppy with the necessary files that you can use to transfer these tools to your PC, Avaya provides another utility, called **mkwavdisk**. To use this utility, at the system prompt, enter **mkwavdisk** and follow the prompts and instructions. This utility copies the **wav2vis.exe** and **vis2wav.exe** files to the floppy diskette (after formatting the floppy diskette for PC use if necessary). After creating the floppy diskette, you can transfer the files to the desired directory on your PC using any standard Windows file transfer procedure.

Once the **wav2vis.exe** and **vis2wav.exe** files have been transferred to the PC, you can use them to convert speech files much as you would using their corresponding UnixWare versions.

If you want to use only the default options for these tools, you can run them by double-clicking their names/icons in Windows Explorer. Then follow the standard Windows interface to select the file to convert and to provide the name and location for the file to be created.

If, however, you want to use any of the options for converting files, you must click **Start**, then **Run...**, and then enter in the **Open:** field the same line command given in Converting *.wav Files to CONVERSANT-Supported Formats or Converting CONVERSANT-Supported Speech Files to *.wav Files. The only difference is that, in the Windows version, you must specify the full path to the executable file, as in the following example:

```
C:\Temp\wav2vis.exe -l -s 2 greeting.wav greeting.vis
```

The options are all the same as those for the UnixWare version of these tools (see Converting *.wav Files to CONVERSANT-Supported Formats or Converting CONVERSANT-Supported Speech Files to *.wav Files).

Note: If you have multiple files you want to convert, you might consider creating a batch (*.bat) file to handle all the file conversions at once. Alternately, you might also want to convert them using a Visual Basic script.

Speech Files

Speech files are comprised of two parts:

- A header section of the format shown in Figure 20 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 20. Speech File Header Section

```
    OxAA     OxFF       N       L
```

The following rules describe the header section:

- `0xAA` and `0xFF` are the 2 bytes of data with bit patterns `10011001` and `11111111`.
- `N` is a unique identifier code representing the speech encoding format, as shown in Table 46 on page 147.
- `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).

You must apply the following rules to the data section:

- Use an even number of bytes.
- Use no more than 500 bytes between headers.
- Use only CELP 16 for the pattern `0xAA 0xFF`.

The length field `L` identifies the number of words (2 bytes each) that follow that include user defined information.

Note: Coding and playback of CELP16 works properly only on the hardware that supports it, such as the SSP, IVC6, or NGTR circuit card. If coding or playback is performed on an unsupported circuit card, the result is silence or noise.

- When CELP16 is played on a circuit card that does not support CELP16, the result is silence or noise, depending on the phrase.
- When a circuit card that does not accept CELP16 is asked to code CELP16, it either runs a default mode or stays in the current mode and the coding circuit card fails to return a phrase.

Table 46. Identifier Codes in Speech Encoding

N	Value	Channel Capacity (Playback/Coding) Per SSP Card
PCM16 (Mu-law)	0 x 34	120/120
ADPCM32	0 x 32	120/120
ADPCM16	0 x 30	120/120
CELP16	0 x 40	120/60

Header bytes are inserted into the speech file so that the header appears at least five times for every second of speech. The headers are aligned on even byte boundaries.

C Text-to-Speech Formats

Overview

Text-to-Speech (TTS) allows you to speak ASCII text with a synthesized voice from an CONVERSANT voice application. When speaking a text field or phrase, assigning a field format helps standard TTS speak the text properly.

Table 47 on page 150 in this appendix describes the formats supported in the standard TTS package for US English and is designed to be used as a reference when you create applications using TTS.

Note: These formats do not necessarily apply to Proxy Text-to-Speech (PTTS) applications. Because PTTS is designed to work with any SAPI-compliant TTS engine, it is impossible to guarantee that any particular format will cause a given output. For more help with PTTS "Speak As" formatting, see the documentation for your TTS speech engine.

For information on setting "Speak As" formats using the Variables Manager in Voice@Work, see the chapter on variables in *Using Voice@Work*, 585-313-207. For information on further customizing TTS, see Appendix D, Advanced Text-to-Speech Features.

For information on using standard TTS in Script Builder applications, see “Using Text-to-Speech” in Chapter 8, “Using Optional Features,” in *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217.

US English TTS Formats

Table 47 shows the formats TTS uses to speak US English text.

Table 47. US English TTS Format Set

Format	Description
A (default)	Verbatim text — characters spoken as words, if possible Example: "Avaya" is spoken as "Avaya".
AC	Alphabetics — characters spoken individually Example: Avaya is spoken asA V A Y A.
ADMD	Numeric month and day, no year Example: “five twelve”
ADMDY	Numeric month and day, 2-digit year Example: “five twelve ninety-nine”
ADMDYY	Numeric month and day, 4-digit year Example: “five twelve nineteen ninety-nine”
ADMSPD	Spoken month, ordinal day, no year Example: “May twelfth”
ADMSPDY	Spoken month, ordinal day, 2-digit year Example: “May twelfth ninety-nine”
ADMSPDYY	Spoken month, ordinal day, 4-digit year Example: “May twelfth nineteen-ninety-nine”
AN$D2	Dollars and cents, with the last two digits as cents Example: 12345 = “one hundred twenty-three dollars and forty-five cents”
AN$D0 (default)	Whole dollars Example: 12345 = “twelve thousand three hundred forty-five dollars”
	1 of 2

Table 47. US English TTS Format Set

Format	Description
AN$D1 AN$D3 AN$D4 AN$D5 AN$D6 AN$D7 AN$D8 AN$D9	Dollars with the respective number of decimal places Example: 12345 spoken as AN$D1 = "one thousand two hundred thirty-four point five dollars" **Note:** If the number has fewer digits than the number of decimal places specified by the format, TTS inserts zeros before the number. Example: 12345 spoken as AN$D9 = "point zero zero zero zero one two three four five dollars"
AN	Whole number, no decimal places Example: 12345= “twelve thousand three hundred forty-five”
AND1 (default) AND2 AND3 AND4 AND5 AND6 AND7 AND8 AND9	Number with the respective number of decimal places Example: 12345 spoken as AND1 = “one thousand two hundred thirty-four point five” **Note:** If the number has fewer digits than the number of decimal places specified by the format, TTS inserts zeros before the number. Example: 12345 spoken as AND9 = "point zero zero zero zero one two three four five"
ATHMAM	Time with 2-digit hour, 2-digit minutes, AM/PM Example: 1234 = “twelve thirty-four PM” **Note:** Input to this field type is assumed to be in 24-hour time format. TTS designates AM and PM.
	2 of 2

D Advanced Text-to-Speech Features

Overview

For some Text-to-Speech (TTS) applications, you might need to customize the way TTS speaks to improve understandability or comprehension. The method to do so involves the use of escape sequences. This chapter covers the following topics for TTS customization:

- Escape Sequences (on page 153)
- Inserting Silent Delays and Pauses (on page 154)
- Changing the Gender of the Speaking Voice (on page 157)
- Changing the Rate of Speech (on page 158)
- Setting Text Classes and Detectors (on page 160)
- Enabling Barge-In for Risky Mode (on page 166)

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.

Escape Sequences

Customization of TTS involves using strings of characters called *escape sequences* that reset the TTS default parameters. These methods of customizing TTS can be used in:

- Script Builder applications in Prompt and Collect action steps and Announce action steps
- Voice@Work applications in Prompt and Collect, Announce, Menu, Automenu, and Voice Capture nodes.

The exact method of changing the defaults depends on the parameter and the application development tool.

Text and Variables

In both Script Builder and Voice@Work applications, you can enter the escape sequences in text or variable field definitions.

- For information on field definitions in Script Builder, see Chapter 3, "Data Management," in *CONVERSANT System Version 8.0 Application Development with Script Builder,* 585-313-217.
- For information on using the Variables Manager in Voice@Work, see the chapter on variables in *Using Voice@Work*, 585-313-207.

Controls

In Voice@Work applications, you can achieve some types of customization by inserting TTS controls with the Insert Control dialog box in the Prompt Editor. When code is generated, the controls are translated to the escape sequences. For information on inserting controls in Voice@Work, see the chapter on prompts in *Using Voice@Work*, 585-313-207.

Inserting Silent Delays and Pauses

Inclusion of silent delays or pauses in an application helps make the speech more natural sounding and may help increase listener comprehension. Table 48 lists the escape sequences used to insert TTS silences into synthesized speech.

Table 48. TTS Escape Sequences for Silence

Sequence	Description
\!si*n* (initial)	• Specifies *n* centiseconds of silence at the beginning of the current text phrase, where 1< *n* < 100 • Must be placed at the beginning of a text phrase and therefore cannot be used in the same phrase as a final silence (\!sf*n*) Since this sequence must occur at the beginning of a a text phrase, it must either be placed first in the Script Builder action step or Voice@Work node or the previous phrase must include end punctuation indicating the end of a phrase — either a period (.), question mark (?), or exclamation mark (!).
\!sf*n* (final)	• Specifies *n* centiseconds of silence at the end of the current text phrase, where 1< *n* < 100 • Must be placed at the end of a text phrase and therefore cannot be used in the same phrase as a final silence (\!sf*n*) Since this sequence must occur at the beginning of a a text phrase, it must either be placed first in the Script Builder action step or Voice@Work node or the previous phrase must include end punctuation indicating the end of a phrase— either a period (.), question mark (?), or exclamation mark (!).
\!(*[*n*])	Specifies *n* centiseconds of silence within the current phrase, where 1< *n* < 60. Can be placed anywhere within a text phrase and can be used in the same phrase with either an initial silence (\!si*n*) or a final silence (\!sf*n*).
Note: When typed in text or a variable, these sequences must be followed by one blank space.	

⚠ **CAUTION:**
Using values outside the limits can cause problems in your application.

Examples for Silence—Using Text

The following examples illustrate the use of escape sequences in text to add silent pauses and delays in an application.

- In the following example from a Script Builder application, a silence of 0.5 seconds occurs before "Hello" is spoken. The delay helps secure the listeners' attention.

```
Announce
  Speak with Interrupt
Text: "\!si50 Hello!"
Text: "Welcome to your Home Shopping Line!"
```

  The same text would be displayed as follows in the Voice@Work Prompt Editor for an Announce node.

```
\!si50 Hello!
Welcome to your Home Shopping Line!
```

- In the following example from a Script Builder application, a silence of 0.7 seconds is added after the phrase "Welcome to the Home Shopping Line!". There is no silence between the word "Hello!" and the text that follows. The silence helps to emphasize the end of the announcement.

```
Announce
  Speak with Interrupt
Text: "\!si50 Hello!"
Text: "\!sf70 Welcome to your Home Shopping Line!"
```

  The same text would be displayed as follows in the Voice@Work Prompt Editor for an Announce node.

```
\!si50 Hello!
\!sf70 Welcome to your Home Shopping Line!
```

Note: The end punctuation (exclamation point) after the first phrase ("Hello!") is necessary for TTS to speak the 0.7-second silence at the end of the second phrase ("Welcome to your Home Shopping Line!"). Without end punctuation in the first phrase, this escape sequence would be ignored.

- In the following example from a Script Builder application, a text phrase contains two silences— a 0.5-second silence at the beginning of the phrase and a 0.7-second silence between the words "Hello" and "and." The placement of these silences helps to make the synthesized speech more natural sounding.

```
Announce
 Speak with Interrupt
Text: "\!si50 Hello \!(*[70]) and welcome to your"
Text: "Home Shopping Line!"
```

  The same text would be displayed as follows in the Voice@Work Prompt Editor for an Announce node.

```
\!si50 Hello \!(*[70]) and welcome to your
Home Shopping Line!
```

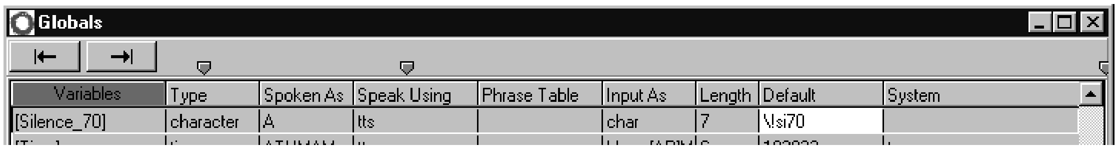
Examples for Silence— Using a Variable

The following example from a Script Builder application shows a silence delay defined as a variable called Silence_70 and used in an Announce action step. A silence of 0.7 seconds occurs between the first and second phrase.

```
Answer Phone
Set Field Value
  Field: Silence_70 = "\!si70 "
Announce
  Speak With Interrupt
    Text: "Welcome to your Home Shopping Line!"
    Field: Silence_70 As A
    Text: "To place an order, please have your"
    Text: "account number ready."
```

Figure 21 shows a detail of the Voice@Work Variables Manager where a variable named [Silence_70] has been created for a silent pause of 0.7 seconds. The escape sequence \!si70 is set in the `Default` field.

Figure 21. Variable Definition for a Silent Delay in Voice@Work



Globals

Variables	Type	Spoken As	Speak Using	Phrase Table	Input As	Length	Default	System
[Silence_70]	character	A	tts		char	7	\!si70	

Once the variable has been created, you can insert it into the application using the Insert Variable dialog box in the Prompt Editor. In the following example, two 0.7-second silences occur, one before the first phrase is spoken and one before the second phrase is spoken. As the example illustrates, the Prompt Editor displays variables in square brackets — [ ].

```
[Silence_70]
Welcome to your Home Shopping Line!
[Silence_70]
To place an order, please have your account number ready!
```

Changing the Gender of the Speaking Voice

TTS permits speech in both male and female voices. The default is male voice. Female voice provides the same capabilities and features as male voice. Voices can be switched by use of the escape sequences shown in Table 49.

Table 49. TTS Escape Sequences for Speech Gender Selection

Sequence	Description
\!uSm	Male voice (default)
\!uSf	Female voice
Note:	When typed in text or a variable, these sequences must be followed by one blank space.

The voice selection stays in effect until an event occurs that removes speech inhibition. At that point, speech reverts to the default (male) voice. Therefore, to create an entire application in female voice, you must reset the voice repeatedly.

The following events remove speech inhibition:

- In a Script Builder application, one of the following action steps occurs:
  - ~ Get Host Screen
  - ~ Send Host Screen
  - ~ Read Table
  - ~ Modify Table
  - ~ Transfer Call
- In a Voice@Work application any Database node (including Query Table, Update Table, Add Record, and Delete Record) or Host node (including Host Connection, Host Send, and Host Send-Receive) occurs
- A message is played that does not allow barge-in (interrupt) in a Prompt and Collect or an Announce action (Script Builder) or in a Prompt and Collect or an Announce node (Voice@Work)
- An external function is called that contains a wait-causing instruction

Example for Speaking Voice Gender — Using Text

The following example from a Script Builder application shows use of the escape sequence to set the TTS female speaking voice.

```
Prompt & Collect
  Prompt
    Speak with Interrupt
      Text: "\!uSf Using touch tones, please enter"
      Text: "your 3-digit account number."
```

- The same text would be displayed as follows in the Voice@Work Prompt Editor for a Prompt and Collect node:

```
\!uSf Using touch tones, please enter
your 3-digit account number.
```

Examples for Speaking Voice Gender— Using a Variable

- In the Script Builder example above, you can define a variable as follows instead of using text to change the speaking voice gender:

```
Set Field Value
  Field: Female_Voice = "\!uSf "
Prompt & Collect
  Prompt
    Speak with Interrupt
      Field: Female_Voice as A
      Text: "Using touch tones, please enter"
      Text: "your 3-digit account number."
```

- For a Voice@Work application, you can define the same variable as in the example above by entering \!uSf in the `Default` field in the Variables Manager. (See Figure 21 on page 156 above for an illustration of the Voice@Work Variables Manager.) Once you insert the variable into a Prompt and Collect node by using the Insert Variable dialog box in the Prompt Editor, the Prompt Editor displays the variable in square brackets.

```
[Female_Voice]
Using touch tones, please enter
your 3-digit account number.
```

Changing the Rate of Speech

Slowing the rate of speech for important words can provide emphasis and increase listener comprehension in an application. Table 50 shows escape sequences used to control the rate of speech for TTS.

Table 50. TTS Escape Sequences for Speaking Rate

Sequence	Description
\!rfastest \!rfast \!rnormal \!rslow \!rslowest	The default rate is "normal" (\!rnormal). Each rate is effective until the end of a sentence, indicated by punctuation such as a period, question mark, or exclamation mark. In text, to change the rate before the end of a sentence, insert a different escape sequence. In Voice@Work, equivalents of \!rfastest, \!rnormal, and \!rslowest can be inserted as controls (\!rfast and \!rslow are not supported as controls).
Note: When typed in text or a variable, these sequences must be followed by one blank space.	

In Voice@Work applications, some of the sequences can also be inserted as controls. When code is generated, the controls are translated to the escape sequences shown in Table 50 on page 158. See “Voice@Work Controls for Rate of Speech” below for more details regarding Voice@Work.

Note: The escape sequences in Table 50 on page 158 are used for TTS on the speech and signal processor (SSP) circuit card only and cannot be used for TTS on the older signal processor (SP) circuit card. (The SP circuit card is not supported for CONVERSANT system releases beginning with Version 8.0). Equivalent sequences supported for the SP circuit card were: \!r0 (fastest), \!r1, \!r2, \!r3, and \!r4 (slowest).

Example for Rate of Speech— Using Text

- In the following example from a Script Builder application, the slower rate is in effect only while the order number is spoken. The rest of the announcement is spoken with the default (\!rnormal) rate.

```
Announce
  Speak with Interrupt
    Text: "Your order number is"
    Text: "\!rslowest 104235\!rnormal."
    Text: "Please use this number"
    Text: "when checking your order."
```

- The same text would be displayed as follows in the Voice@Work Prompt Editor for an Announce node:

```
Your order number is
\!rslowest 104235\!rnormal.
Please use this number
when checking your order.
```

Examples for Rate of Speech — Using a Variable

- In the Script Builder example above, you can set a variable such as the following instead of using text to change the rate of speech.

```
Set Field Value
  Field: Rate_Slowest = "\!rslowest "
```

- In a Voice@Work application, you can define the same variable as in the example above by entering \!rslowest in the `Default` field in the Variables Manager. (See Figure 21 on page 156 above for an illustration of the Voice@Work Variables Manager.) Once you insert the variable into the application by using the Insert Variable dialog box in the Prompt Editor, the Prompt Editor displays the variable as follows:

```
[Rate_Slowest]
```

Voice@Work Controls for Rate of Speech

In Voice@Work applications, you can use the Insert Control dialog box in the Prompt Editor to insert controls that change the rate of speech. When code is generated, these controls are translated to the escape sequences shown in Table 50 on page 158. The following controls are available:

- {tts fast} — equivalent to \!rfastest
- {tts normal} — equivalent to \!rnormal
- {tts slow} — equivalent to \!rslowest

Note: When you use these controls in a Version 8.0 application, ensure that the `Use SP style TTS speech rate codes` checkbox is unselected on the Voice@Work Code Generation Options window. This allows the appropriate escape sequences for the SSP circuit card to be generated.
Select the `Use SP style speech rate codes` checkbox only for applications on systems using the older SP circuit card (not supported in CONVERSANT system Version 8.0). See the chapter on the code generation tool in *Using Voice@Work*, 585-313-207, for information.

Setting Text Classes and Detectors

Background

In an application, you can designate the format of a field (Script Builder) or a variable (Voice@Work) to be spoken by TTS as:

- Character
- Time
- Date
- Number
- Dollar amount
- Currency real (Voice@Work only)

(See US English TTS Formats for a list of the formats supported for TTS).

This information is often sufficient for TTS to speak text correctly. However, when the context for the text is ambiguous or requires a specific style of speaking, more information must be provided. To illustrate the problem, consider the example of TTS trying to speak separate fields of information containing a person’s name, street address, and city, state, and zip code obtained from a database in the form below. TTS may not know whether to speak "22587" as part of an address or as a number because it is a separate field from the rest, presented with little context, and it could be either.

customer name field = `Jane Doe`
city/state field = `Columbus OH 43213`
zip code field = `43213`
address field =`22587 S. Hamilton St.`

Classes

To solve this sort of problem, TTS allows you to explicitly assign text to any of the following four classes:

- Telephone number
- Proper name
- Address
- Fraction

Class Detector Modes

For TTS to speak text that is assigned to these classes, you must also assign one of the following class detector modes:

- Off
- Conservative mode (default)
- Risky mode

Note: Initially all detectors are set to conservative mode.

The detector mode determines how TTS speaks ambiguous text.

- Risky mode sets a priority for the designated class to resolve ambiguity about how the information is to be classified. The setting of risky mode will be the highest priority at the time for the field to be spoken. TTS will then know, for example, that "22587" in the problem described above is part of an address.
- Conservative mode requires more contextual information to resolve ambiguous text.

Table 51 on page 162 shows the escape sequences that are used for class detector modes.

- In both Script Builder and Voice@Work applications, these sequences can either be entered in text or defined as variables. The class detector mode remains in the new state until it is changed or until the application ends.

Note: One technique of resetting the mode is to use an Announce action step or node to speak only the escape sequence.

- In Voice@Work applications, a subset of the class detector modes can be inserted as controls. When code is generated, the controls are translated to the escape sequences shown in Table 51 on page 162. The control sets the class detector to risky mode and affects only the next line in the Prompt Editor. The class detector is then automatically reset to conservative mode.

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.

Since TTS speaks text better in context than in isolation, you may want to concatenate fields to be spoken.

- For Voice@Work, see information on the **concat** expression in sections describing the Set and Test node in the chapter on node descriptions in *Voice@Work*, 585-313-297.
- For Script Builder, see information on the **concat** function in Chapter 11, "Using Advanced Features," of *CONVERSANT System Version 8.0 Application Development with Script Builder,* 585-313-217.

Table 51. TTS Escape Sequences for Class Detectors

Sequence	Description
Telephone number `\!npo` (off) `\!npr` (risky mode) `\!npc` (conservative mode)	In conservative mode, TTS speaks telephone numbers as individual digits, except for 800 in the area code position, which is spoken "eight hundred". For example, 555 1365 is spoken "five five five (pause) one three six five," not "five million (pause) five hundred fifty-one thousand (pause) three hundred sixty-five." Risky mode also ensures that "vanity" telephone numbers, such as 1 800 SHOPPER, are spelled out.
Proper name `\!nno` (off) `\!nnr` (risky mode) `\!nnc` (conservative mode)	Risky mode is needed only when the data might contain name-related abbreviations or titles. Otherwise, use conservative mode. In risky mode, titles such as “Mr.” and ambiguous abbreviations such as “Dr.” are spoken correctly. Abbreviations such as “Sr.” (as in “John Doe, Sr.”) and “Jr.” expand correctly in either risky or conservative mode because there is only one possible interpretation of the abbreviations. **Note:** TTS 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.

1 of 2

Table 51. TTS Escape Sequences for Class Detectors

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” rather than “doctor” for “Dr.” and “street” rather than “saint” for "St." Standard post office abbreviations apply.
Fraction `\!nfo` (off) `\!nfr` (risky mode) `\!nfc` (conservative mode	Use risky mode so that, for example, “1/4” is spoken as “one fourth” and not “one slash four.” (To have TTS speak “1/4” as a date, use conservative date mode.)
Note: When typed in text or a variable, these sequences must be followed by one blank space. Sequences to turn detectors off (\!npo, \!nno, \!nao, \!nfo), are not available as controls in Voice@Work. For more information, see Voice@Work Controls for Class Detectors (on page 165).	
	2 of 2

Examples for Class Detectors — Using Text

The examples below illustrate use of class detectors set in text for the telephone number, address, and proper name classes in Script Builder applications. Examples for Voice@Work applications displayed in the Prompt Editor would be similar except that no quotation marks are required around the text.

Telephone Number

The following example sets the telephone number detector to risky mode, speaks a "vanity" telephone number, and then turns the detector off because no more telephone numbers are expected.

```
Announce
 Speak with Interrupt
   Text: "Please call \!npr 1-800 SHOPPER \!npo "
   Text: "for more information."
```

Address

The example below assumes that a Script Builder application has obtained from a database the following data for the specified fields:

Customer_address:	12345 Mulberry St.
Customer_state	CT

The Announce action sets the address detector to risky mode, speaks an address from the `Customer_address` and `Customer_state` database fields, and then sets the detector back to conservative mode. Risky mode is needed to rank possibilities for the address so that "St." is spoken as "street" (and not "saint" and CT" is spoken as "Connecticut" (and not "court" or "count"). The detector is set back to conservative mode, without priority, in preparation to process the next field.

```
Announce
  Speak with Interrupt
    Text: "\!nar"
    Text: "Your address is"
  Field: Customer_address as A.
  Field: Customer_state as A
    Text: "\!nac"
```

Proper Name

The following example sets the proper name detector to risky mode, speaks a customer name from a `Customer_name` database field, and then sets the detector back to conservative mode. Risky mode is needed because the data may contain name-related abbreviations or titles.

```
Prompt & Collect
  Prompt
    Speak with Interrupt
      Text: "\!nnr"
      Text: "Your name is"
    Field: Customer_Name as A
      Text: "\!nnc"
      Text: "Press 1 for yes, 2 for no."
```

Examples for Class Detectors — Using a Variable

- In the Script Builder example above for the proper name detector, you can set variables such as the following instead of using text to change the class detector:

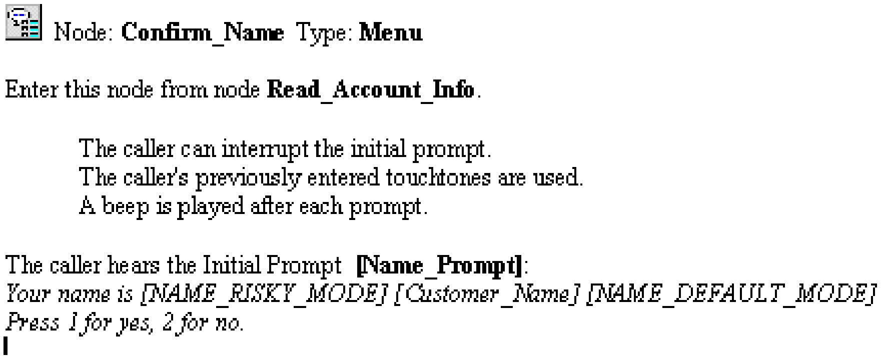
```
Set Field Value
  Field: Name_Risky_Mode = "\!nnr "
  Field: Name_Default_Mode = "\!nnc "
```

- In an equivalent Voice@Work application for the proper name detector, you can achieve the same result by creating variables with the following definitions in the Variables Manager (see Figure 21 on page 156 above for an illustration of the Voice@Work Variables Manager):

Name	Type	Spoken As	Speak Using	Phrase Table	Input As	Length	Default
[NAME_DEFAULT_MODE]	character	A	tts		char	10	\!nnc
[NAME_RISKY_MODE]	character	A	tts		char	10	\!nnr

You can then use the Insert Variable dialog box in the Prompt Editor, to insert the variables into the application. If you print the node information, the output for this portion of the node resembles Figure 22.

Figure 22. Example of Address Detector Variables in Voice@Work Node Information



Node: **Confirm_Name** Type: **Menu**

Enter this node from node **Read_Account_Info**.

The caller can interrupt the initial prompt.
The caller's previously entered touchtones are used.
A beep is played after each prompt.

The caller hears the Initial Prompt **[Name_Prompt]**:
Your name is [NAME_RISKY_MODE] [Customer_Name] [NAME_DEFAULT_MODE]
Press 1 for yes, 2 for no.

Voice@Work Controls for Class Detectors

Voice@Work allows you to use the Insert Control dialog box in the Prompt Editor for the following class detectors:

- {tts address}
- {tts fraction}
- {tts proper name}
- {tts phone number}

Inserting the control sets the class detector to risky mode and affects only the next line in the Prompt Editor. The class detector is then automatically reset to the default (conservative mode). Therefore, if risky mode is needed for several lines, you can either insert the control repeatedly or use an escape sequence to set the class detector in a variable or text statement.

Note: The class detectors cannot be turned off (o mode) with controls. To turn off the class detectors, you must use an escape sequence in text or a variable.

Enabling Barge-In for Risky Mode

For Script Builder and Voice@Work applications, the barge-in feature works for TTS in risky mode only if a field is used to turn it on, not a text statement (or a control in Voice@Work).

For example, in a Script Builder application the address detector can be set to risky mode in text as follows when the ability of the caller to barge in is *not* needed.

```
Announce
  Speak With Interrupt
    Text: "\!nar"
```

If barge-in is needed, a variable such as the following can be defined for the address detector in risky mode:

```
Set Field Value
  Field: Adsress_Risky_Mode = "\!nar"
```

When needed, the variable can be included as:

```
Field: Address_Risky_Mode As A
```

E Recognition Post-Processing

Overview

This appendix describe the post-processing features that can be used in Script Builder applications.

- Further Processing (on page 167) discusses data interface processes (DIPs) associated with dial pulse recognition (DPR), WholeWord speech recognition, and FlexWord speech recognition.
- The following sections discuss processing features that increase the system accuracy of recognition:
  - ~ Return Top 4 (RT4) (on page 169)
  - ~ Prefix List Check (LSC) (on page 170)
  - ~ Luhn Check (LNC) (on page 170)
  - ~ Expected Value (EXV) (on page 171)
  - ~ Close Match (CLM) (on page 171)

Note: These features are supported for Script Builder applications only.

Further Processing

A data interface process (DIP) called **recog_dip** is associated with DPR, WholeWord speech recognition, and FlexWord speech recognition. The **recog_dip** is an optional, post-processing mechanism for enhancing the performance of some recognition tasks. It furnishes additional recognition processing beyond what is already provided by the models and grammars of the recognizer.

This DIP is located in **/vs/bin/vrs/recog_dip**. Whether the **recog_dip** is used during a particular recognition task is determined in the **/vs/data/sr_file** file.

When the **recog_dip** is in use, the recognizer sends its top four choices (and a score for each) to the DIP, which in turn makes a call to the customer-editable function **find_best.c** located in the source code file **/att/asr/find_best.c**. The **find_best.c** routine is responsible for making a final selection from the list of recognition choices and passing it back to the script as $CI_VALUE or other specified customer input variable.

Many general post-processing algorithms have been provided in the generic version. Look to see if the existing functionality exists before adding custom features. If you do edit this code, be sure to back it up prior to updating generics or reinstalling the recognizer package.

The default version of the **find_best.c** routine (the one that comes with the recognition package) simply selects the result with the highest score that falls within the constraint set for the minimum and maximum number of digits. Therefore, if the recognizer is applying a fixed-length grammar, the default version of the **find_best.c** routine adds no value to the recognition procedure. If the recognizer is applying a variable-length grammar, the default routine **find_best.c** is useful for range checking.

Note: Where possible, application constraints should be placed in a custom grammar to gain the most improvement in accuracy.

To access the post-processing features described in this appendix, it is necessary to create a custom recognition type. Custom recognition types should be defined in directory **/att/asr/sr_files/**.

Figure 23 is an example of defining the recognition type file format.

Figure 23. Example of Defining the Recognition Type File Format

```
class type   min  max  recognizer      interrupt  grammar  dip       subr buf_sz
------------------------------------------------------------------------------------
SR RT4_an    8    8   IRD_WHOLE_WORD    Y        BP_yn   im."recog_dip"
```

After supplying the custom recognition type, run the command **/att/asr/tools/bin/merge** to update the Intuity CONVERSANT system.

The real power of the **recog_dip** becomes apparent when the **find_best.c** routine is customized to place added constraints on the recognizer results. This helps limit the valid recognition results and thereby improves recognition accuracy. For example, an application is attempting to recognize a 10-digit telephone number and it knows that when the area code is 614, the final 4 digits of the telephone number will always fall between 5,500 and 9,000, and when the area code is 216 or 513 the final 4 digits will always be between 1,000 and 4,400. Customizing the **find_best.c** routine to account for these data dependencies, should provide an increase in recognition performance. The point to note here is that a certain amount of dependency between digits can be accomplished with a custom grammar, which gets implemented at the recognizer level. However, more complex patterns or dependencies within the expected user input are best handled with the **find_best.c** routine and the **recog_dip**.

After making and compiling changes made to the **find_best.c** routine, you will need to remake the **recog_dip**. **Makefile** for this is included in the **/att/asr** directory. Test any changes you make thoroughly, as mistakes could severely impact your application’s recognition accuracy.

Return Top 4 (RT4)

RT4 is a processing feature that returns up to four choices, separated by the pound (#) character, for what the system determined the caller said, sorted by recognition score. For example, the candidates can be validated in order against a database within the script. By using this processing feature, more information is used to ensure greater correctness.

In the examples of returns from RT4 processing in Figure 24, `N` refers to a digit [0-9].

Figure 24. Example of Returns from RT4 Processing

`NNN#NNN#NNN#NNN#`	Four results were returned, each of length three.
`?#NNN#`	The best result was rejected ("?"), with only one other valid result.
`####`	All results were rejected (empty string).

Script Builder Example

The program in Figure 25 was created in Script Builder to demonstrate the use of the RT4 recognition DIP:

Figure 25. Example of Use of the RT4 Recognition DIP

```
1.  Answer Phone
    main:
2.  Prompt & Collect
     Prompt
       Speak With Interrupt
         Text: "say a 5 digit number"
       Input
         Min Number Of Digits: 05
         Max Number Of Digits: 05
         RECOG: SR   RT4_BP5
       Checklist
         Case: "Input Ok"
           Continue
         Case: "Initial Timeout"
           Reprompt
         Case: "Too Few Digits"
           Reprompt
         Case: "No More Tries"
           Quit
   End Prompt & Collect
   loop:
3.   External Function
     Function Name: length
     e Arguments: $CI_VALUE
       Return Field:  Len
4.   Evaluate
   If Len   > 5
5.   External Function
     Function Name: parse
     Use Arguments: cand  $CI_VALUE  "#"
6.   Announce
      Speak With Interrupt
        Text: "You said"
        Field: cand As AC
7.   Goto loop
     End Evaluate
8.   Goto main
9.   Announce
     Speak With Interrupt
       Text: "good bye"
10. Disconnect
```

This example illustrates the following key points:

- In the Prompt and Collect action, the recognition type is set to **RT4_DIG** (to enable the RT4 DIP, the first three characters must be “RT4”). In this example **RT4_DIG** is a custom recognition type from a custom package.
- The **loop** label and **Goto loop** action provides the looping construct that helps unpack the values from the returned variable `temp_val`.
- The external function **length** allows the application to decide how many values are packed in `temp_val` using the evaluate action.
- The external function **parse** provides the unpacking mechanism of `temp_val`. It also strips the trailing “#” symbol from the value.

Prefix List Check (LSC)

LSC compares recognizer results with values in the file **/att/asr/lst.prefix**. The entries in the list need not be the same length. The post-processing marks as valid only those results that match any list entry as a prefix. This feature currently supports only one list with a maximum of 256 entries per list.

Luhn Check (LNC)

The Luhn check is a processing feature that tests the validity of an account number. For example, most credit card numbers are encoded with a “check digit”, usually appended to a number, that validates the authenticity of the number. A simple algorithm is applied to the other digits of the number which yields the check digit. By running the algorithm, and comparing the check digit obtained from the algorithm with the check digit encoded with the credit card number, you can verify that you have correctly read all of the digits and that they are a valid combination.

Validation Formula

Check **/att/asr/mod10.c** for an explanation of the check sum formula. The algorithm can be found in the comments section of the **mod10.c** file that accompanies the WholeWord recognition package.

Expected Value (EXV)

An EXV for a recognition can be passed from a script to the recognition post-processing function and can be used to validate or to reject a particular speech recognition candidate.

In the script, the expected value should be placed in the system variable $FINDBEST prior to calling the recognizer in the Prompt and Collect action. The $FINDBEST value is accessible in the customer-editable function `find_best`, which is within the recognizer post-processing routine **recog_dip**.

For example, assume that at a point in the call flow it has been determined that the caller is located in Columbus, Ohio, and that the caller is asked to speak his or her US English 5-digit zip code. The EXV can be set to 432 because the first three digits for all zip codes in this city are 432. In the post-processing for the case of the recognition type for zip code, the first 3 digits of the recognition result can be compared to 432. The first candidate that matches the pattern can be returned as the best choice.

Close Match (CLM)

CLM is a function available to the customer-editable post processing that compares two strings. The function does an **align in /att/asr/aligner.c** and if the two strings differ only by one substitution error or one insertion/deletion pair error, the function returns 1. An example of the insertion/deletion pair would be if the caller speaks the digits, “43230” and the system recognizes, “64320.” In this case, the system has incorrectly inserted the “6” and deleted the “3” before the zero. As already stated, this function returns 1.

Combining the use of CLM and EXV can improve application accuracy. For example, assume that at a point in the call flow the caller has been identified and that a caller profile has been read into the script. If the script requests verification of the house number of the caller's address, the EXV can be set to the house number prior to the Prompt & Collect step. In the post-processing for the recognition type for the house number, the recognition result string can be compared to the EXV string to determine if they closely match. This assumes that if the strings do not exactly match, it is most likely a recognition error. If the strings match within 1 error, then the result is accepted as correct.

Further Processing Prefixes and Combinations

Each of the processing functions above and all combinations are invoked by using an appropriate prefix to specify a custom recognition type. Table 52 shows the prefix names for the further processing functions.

Table 52. Prefix Names For Further Processing Functions

Function	Prefix
Returns (up to) the top four results to the script	RT4
Performs a Luhn (**mod10**) checksum calculation	LNC
Performs a prefix list check	LSC
Performs an expected value check	EXV
Performs a close match check	CLM

For combinations, prefixes are strung together at the beginning of the recognition type. Table 53 illustrates the functions of some combination prefixes.

Table 53. Examples of Combination Prefixes and Functions

Function	Prefix
Returns (up to) the top four results that satisfy the Luhn check	LNCRT4
Returns (up to) the top four results that satisfy the close match check	CLMRT4
Returns (up to) the top four results that satisfy the Luhn check and prefix list check	LNCLSCRT4

Glossary

Numerics

23B+D

23 bearer (communication) and 1 data (signaling) channel on a T1 PRI circuit card.

30B+D

30 bearer (communication) and 1 data (signaling) channel (plus framing channel 0) on an E1 PRI circuit card.

3270 interface

A link between one or more CONVERSANT machines and a host mainframe. In CONVERSANT system documentation, the 3270 interface specifically means the link between one or more system machines and an IBM host mainframe.

47B+D

47 bearer (communication) and 1 data (signaling) channel on two T1 PRI circuit cards.

4ESS®

A large Lucent central office switch used to route calls through the telephone network.

A

AC

alternating current

ACD

automatic call distributor

AD

application dispatch

AD-API

application dispatch application programming interface

adaptive differential pulse code modulation

A means of encoding analog voice signals into digital signals by adaptively predicting future encoded voice signals. This adaptive modulation method reduces the number of bits required to encode voice. See also “pulse code modulation.”

adjunct products

Products (for example, the Adjunct/Switch Application Interface) that the system administers via cut-through access to the inherent management capabilities of the product itself. This is in opposition to the ability of the system to administer the switch directly.

Adjunct/Switch Application Interface

An optional feature package that provides an Integrated Services Digital Network-based interface between Avaya PBXs and adjunct processors.

ADPCM

adaptive differential pulse code modulation

ADU

asynchronous data unit

advanced speech recognition

A speech recognition ability that allows the system to understand WholeWord, FlexWord, and Natural Language Speech Recognition inputs from callers.

affiliate

A business organization that Avaya controls or with which Avaya is in partnership.

AGL

application generation language

ALERT

System alerter process

alerter

A system process that responds to patterns of events logged by the “logdaemon” process.

American Standard Code for Information Interchange

A standard code for data representation that represents alphanumeric characters as binary numbers. The code includes 128 uppercase and lowercase letters, numerals, and special characters. Each alphanumeric and special character has an ASCII code (binary) equivalent that is 1 byte long.

analog

An analog signal, such as voice or music, that varies in a continuous manner. An analog signal may be contrasted with a digital signal, which represents only discrete states.

ANI

automatic number identification

announcement

A message the system plays to the caller to provide information. The caller is not asked to give a response. Compare to prompt.

API

Application programming interface

application

The automated transaction (interactions) among the caller, the voice response system, and any databases or host computers required for your business. See also application script.

application administration

The component of the system that provides access to the available applications and helps you manage and administer them.

application installation

A two-step process in which the CONVERSANT system invokes the TSM script assembler for the specific application name and moves files to the appropriate directories.

application script

The computer program that controls the application (the transaction between the caller and the system). The CONVERSANT system provides several methods for creating application scripts, including Voice@Work, Script Builder, Transaction Assembler Script (TAS) language, and the Intuity Response Application Programming Interface (IRAPI).

application simulation

A process in which the system simulates the behavior of an application as it is expected to behave on the CONVERSANT system. It is useful as a debugging tool.

application verification

A process in which the system verifies that all the components needed by an application are complete.

ASCII

American Standard Code for Information Interchange

ASI

analog switch integration

ASR

advanced speech recognition

asynchronous communication

A method of data transmission in which bits or characters are sent at irregular intervals and spaced by start and stop bits rather than by time. Compare to synchronous communication.

asynchronous data unit

An electronic communications device that allows computer systems to communicate over asynchronous lines more than 50 feet (15 meters) in length.

asynchronous event

An event detected by the system that disrupts the normal flow of an application that is running. At present, the CONVERSANT system recognizes only one type of asynchronous event—a hang up.

automatic call distributor

That part of a telephone system that recognizes and answers incoming calls and completes these calls based on a set of instructions contained in a database. The ACD can send the call to an operator or group of operators as soon as the operator has completed a previous call or after the system has played a message to the caller.

automatic number identification

A method of identifying the calling party by automatically receiving a string of digits that identifies the calling station of a particular customer.

B

back up

The preservation of the information in a file in a different location so that the data is not lost in the event of hardware or system failure.

backing up an application

Using a utility that makes an archive copy of a completed application or an interim copy of an application in progress. The backup copy can be restored to the system if the online version is damaged, or if you make revisions and want to go back to the previous version.

barge-in

A capability provided by WholeWord speech recognition, Dial Pulse Recognition (DPR), and Natural Language Speech Recognition (NLSR) that allows callers to speak or enter their responses during the prompt and have those responses recognized (similar to the Speak with Interrupt capability). See also echo cancellation.

batch file

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

BB

bulletin board

binary synchronous communications

A character-oriented synchronous link protocol.

blind transfer protocol

A protocol in which a call is completed as soon as the extension is dialed, without having to wait to see if the telephone is busy or if the caller answered.

bps

bits per second

BRDG

call bridging process

bridging

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

BSC

binary synchronous communications

bundle

In the context of the Enhanced File Transfer package, this term is used to denote a single file, a group of files (package), or a combination of both.

byte

A unit of storage in the computer. On many systems, a byte is 8 bits (binary digits), which is the equivalent of one character of text.

C

call classification analysis

A process that enables application designers to use information available within the system to classify the disposition of originated and transferred calls. Intelligent CCA is provided with the system. Full CCA is an optional feature package.

call data event

A parameter that specifies a list of variables that are appended to a call data record at the end of each call.

call data handler process

A software process that accumulates generic call statistics and application events.

called party number

The number dialed by the person making a telephone call. Telephone switching equipment can use this number to selectively route an incoming call to a particular department or agent.

caller

The party who calls for a service, gets connected to the system, and interacts with it. Because the system can also make outbound calls for service, the caller can also be the person who responds to those outbound calls.

call flow

See transaction.

call progress tones

Standard telephony sounds that indicate the status of the call. These sounds include busy, fast busy, ringback, reorder, etc.

card cage

An area within a hardware platform that contains and secures all of the standard and optional circuit cards used in the system.

cartridge tape drive

A high-capacity data storage and retrieval device that can be used to transfer large amounts of information onto high-density magnetic cartridge tape based on a predetermined format. This tape can be removed from the system and stored as a backup or used on another system.

CAS

channel associated signalling

caution

An admonishment or advisory statement used in the system documentation to alert the user to the possibility of a service interruption or a loss of data.

CCA

call classification analysis

CDH

call data handler process

CELP

code excited linear prediction

central office

A location in which large telecommunication devices such as telephone switches and network access facilities are maintained. These locations follow strict installation and operation requirements.

central processing unit

See processor.

CGEN

Voice system general message class

channel

See port.

channel associated signaling

A type of signaling that can be used on E1 circuit cards. It occurs on channel 16.

CICS

Customer Information Control System

circuit card upgrade

A new circuit card that replaces an existing card in the platform. Usually the replacement is an updated version of the original circuit card to replace technology made obsolete by industry trends or a new system release.

cluster controller

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

CMS

Call Management System

CO

central office

code excited linear prediction

A means of encoding analog voice signals into digital signals that provides excellent quality with use of minimum disk space.

command

An instruction or request the user issues to the system software to make the system perform a particular function. An entire command consists of the command name and options.

configuration

The arrangement of the software and hardware of a computer system or network. The system configuration includes either a standard or custom processor, peripheral equipment (for example, printers and modems), and software applications. Configuration also refers to the way in which the switch network is set up; that is, the types of products that are in the network and how those products communicate.

configuration management

The component of the system that allows you to manage the current configuration of voice channels, host sessions, and database connections, assign scripts to run on specific voice channels or host sessions, assign functionality to SSP and E1/T1 circuit cards, and perform various maintenance functions.

connect and disconnect (C and D) tones

DTMF tones that inform the system when the attendant has been connected (C) and when the caller has been disconnected (D).

connected digits

A sequence of digits that the system can process as a group, rather than requiring the caller to enter the digits one at a time.

Converse Data Return (conv_data)

A Voice@Work external function or a Script Builder external action that supports the DEFINITY® call vectoring (routing) feature by enabling the switch to retain control of vector processing in the system environment. It supports the DEFINITY “converse” vector command to establish a two-way routing mechanism between the switch and the system to facilitate data passing and return.

controller circuit card

A circuit card used on a computer system that controls its basic functionality and makes the system operational. These circuit cards are used to control magnetic peripherals, video monitors, and basic system communications.

copying an application

A utility in which information from a source application is directed into the destination application.

coresidency

The ability of two products or services to operate and interact with each other on a single hardware platform.

CPE

customer-provided equipment or customer premise equipment

CPN

called party number

CPT

call progress tones

CPU

central processing unit

crash

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

CSU

channel service unit

custom grammar

See custom vocabulary.

custom speech

Unique words or phrases to be used in system voice prompts that Avaya records on a per-customer basis.

custom vocabulary

A specialized package of unique words or phrases created on a per-customer basis and used by WholeWord or FlexWord speech recognition.

Customer Information Control System

Part of the operating system that manages resources for running applications (for example, IND$FILE). Note that TSO and CMS provide analogous functionality in other host environments.

CVS

converse vector step

D

danger

An admonishment or advisory statement used in the system documentation to alert the user to the possibility of personal injury or death.

data interface process

A software process that communicates with interactive voice response (IVR) applications.

database

A structured set of files, records, or tables.

database field

A field used to extract values from a local database and form the structure upon which a database is built.

database record

The information in a database for a person, product, event, and so on. The database record is made up of individual fields for each information item.

database table

A structure, made up of columns and rows, that holds information in a database. Database tables provide a means of storing information that changes too often to “hard-code,” or store permanently, in the transaction outline.

dB

decibel

DB

database

DBC

database checking process

DBMS

database management system

DC

direct current

DCE

data communications equipment

DCP

digital communications protocol

debug

The process of locating and correcting errors in computer programs; also referred to as troubleshooting.

default

The way a computer performs a task in the absence of other instructions.

default owner

The owner of a channel when no process takes ownership of that channel. The default owner holds all idle, in-service channels. In terms of the IRAPI, this is typically the Application Dispatch process.

diagnose

The process of performing diagnostics on a bus or on circuit cards.

dial ahead

The ability to collect and process touchtone inputs in sequence, even when they are received before the prompts.

dial pulse recognition

A method of recognizing caller pulse inputs from a rotary telephone.

dialed number identification service

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

dial through

A capability provided by touchtone and dial pulse recognition that allows callers to enter their responses during the prompt and have those responses recognized (similar to the Speak with Interrupt capability). See also barge-in and echo cancellation.

DIMM

dual in-line memory module

DIO

disk input and output process

DIP

data interface process

directory

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

display errdata

A command that displays system errors sent to the logger.

DMA

direct memory address

DNIS

dialed number identification service

DPR

dial pulse recognition

DSP

digital signal processor

DTE

data terminal equipment

DTMF

dual tone multi-frequency

DTR

data terminal ready

dual 3270 links

A feature that provides an additional physical unit (PU) for a cost-effective means of connecting to two host computers. The customer can connect a system to two separate FEPs or to a single FEP shared by one or more host computers. Each link supports a maximum of 32 LUs.

dual tone multi-frequency

A touchtone sound that is an audio signal including two different frequencies. *DTMF feedback* is the process of the switch providing this information to the system. *DTMF muting* is the process of ignoring these tones (which might be simulated by human speech) when they are not needed for the application.

dump space

An area of the disk that is fixed in size and should equal the amount of RAM on the system. The operating system “dumps” an image of core memory when the system shuts down automatically. The dump can be fetched after rebooting to help in analyzing the cause of the shutdown.

E

E&M

Ear and Mouth

E1 / T1

Digital telephony interfaces, commonly called *trunks*. E1 is an international standard at 2.048 Mbps. T1 is a North American standard at 1.544 Mbps.

Ear and Mouth

A common T1 trunking protocol for connection between two switches.

EBCDIC

Extended Binary Coded Decimal Interexchange Code

echo cancellation

The process of making the channel quiet enough so that the system can hear and recognize WholeWord, dial pulse, and Natural Language inputs during the prompt. See also barge-in.

ECS

Enterprise Communications Server

editor system

A system that allows speech phrases to be displayed and edited by a user.

EFT

Enhanced File Transfer

EIA

Electronic Industries Association

EISA

Extended Industry Standard Architecture

EMI

electromagnetic interference

emulator

Software on one operating system that imitates or reproduces the behavior of input and output on a different operating system.

engine

The software used to perform speech recognition or text-to-speech functions. Usually used with reference to proxy software and systems. See also Proxy Text-to-Speech (PTTS) and Natural Language Speech Recognition (NLSR).

enhanced basic speech

Prerecorded speech available from Avaya in several languages. Sometimes called standard speech.

Enhanced File Transfer

A feature that allows the transferring of files automatically between the CONVERSANT system and a synchronous host processor on a designated logical unit.

Enhanced Serial Data Interface

A software-controlled and hardware-controlled method used to store data on magnetic peripherals.

Enterprise Communications Server

The telephony equipment that connects your business to the telephone network. Sometimes called a switch.

error message

A message on the screen indicating that something is wrong with the system, often with a suggestion of how to correct it.

ESD

electrostatic discharge

ESDI

Enhanced Serial Data Interface

ESS

electronic switching system

EST

Enhanced Software Technologies, Inc.

ET

error tracker

Ethernet

A name for a local area network that follows IEEE Standard 802.3. Supported implementations are 10Baset and 100Baset.

event

The notification given to an application when some condition occurs that is generally not encountered in normal operation.

EXTA

external alarms feature message class

external actions

Specific predefined (or customer-created) system tasks that Script Builder can call or *invoke* to interact with other products or services. When an external action is invoked, the systems displays a form that provides choices in each field for the application developer to select. Examples are Call_Bridge, Make_Call, SP_Allocate, SR_Prompt, and so on. In Voice@Work, external actions are called external functions.

external functions

Specific predefined (or customer-created) system tasks that Voice@Work can call or *invoke* to interact with other products or services. The function allows the application developer to enter the arguments for the function to act on. Examples are concat, getarg, length, substring, and so on. In Script Builder, external functions are called external actions.

F

FAX Actions

An optional feature package that allows the system to send fax messages.

FCC

Federal Communications Commission

FDD

floppy disk drive

feature

A function or capability of a product or an application within the system.

feature package

An optional package that may contain both hardware and software resources to provide additional functionality to a standard system.

feature_tst script package

A standard system software program that allows a user to perform self-tests of critical hardware and software functionality.

FEP

front end processor

field

See database field.

FIFO

first-in-first-out processing order

file

A collection of data treated as a basic unit of storage.

file transfer

An option that allows you to transfer files interactively or directly to and from UNIX using the file transfer system (FTS).

filename

Alphabetic characters used to identify a particular file.

FlexWord™ speech recognition

A type of speech recognition based on subword technology that recognizes phonemes or parts of words in a specific language. See also subword technology.

foos

facility out-of-service state

FTS

file transfer process message class

Full CCA

A feature package that augments the types of call dispositions that Intelligent CCA can provide.

function key

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

G

GEN

PRISM logger and alerter general message class

grammar

The inputs that a recognizer can match (identify) from a caller.

GUI

graphical user interface

H

hard disk drive

A high-capacity data storage and retrieval device that is located inside a computer platform. A hard disk drive stores data on nonremovable high-density magnetic media based on a predetermined format for retrieval by the system at a later date.

hardware

The physical components of a computer system. The central processing unit, disks, tape and diskette drives, and so on, are all hardware.

hardware upgrade

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

HDD

hard disk drive

High Level Language Applications Programming Interface

An application programming interface that allows a user to write custom applications that can communicate with a host computer via an API.

HLLAPI

High Level Language Applications Programming Interface

HOST

host interface process message class

host computer

A computer linked to a network to provide a range of services, such as database access and computation. The host computer operates in a time-sharing manner with other computers linked to it via the network.

hwoos

hardware out-of-service state

Hz

Hertz

I

IBM

International Business Machines

iCk or ICK

The system integrity checking process.

ID

identification

IDE

integrated disk electronics

idle channel

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

IE

information element

IEEE

Institute of Electrical and Electronic Engineers

IND$FILE

The standard SNA file transfer utility that runs as an application under CICS, TSO, and CMS. IND$FILE is independent of link-level protocols such as BISYNC and SDLC.

independent software vendor

A company that has an agreement with Avaya to develop software to work with the system to provide additional features required by customers.

indexed table

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

industry standard architecture

A PC bus standard that allows processors and other circuit cards to communicate with each other.

INIT

voice system initialization message class

initialize

To start up the system for the first time.

inserv

in-service state

Integrated Services Digital Network

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

intelligent CCA

Monitoring the line after dialing is complete to determine whether a busy, reorder (fast busy), or other failure has been encountered. Intelligent CCA also recognizes when the extension is answered or if the extension is not answered after a specified number of rings. The monitoring capabilities are dependent on the network interface circuit card and protocol used

interface

The access point of a system. The interface is designed to provide you with easy access to the software capabilities of the system.

interrupt

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

Intuity Response Application Programming Interface

A library of commands that provide a standard development interface for voice-telephony applications.

IOB

I/O companion card to the SBC. This is part of the CPU Complex.

IPC

interprocess communication

IRAPI

Intuity Response Application Programming Interface

IRQ

interrupt request

ISA

industry standard architecture

ISDN

Integrated Services Digital Network

ISV

independent software vendor

ITAC

International Technical Assistance Center

K

Kbps

kilobytes per second

KB

kilobyte

keyboard mapping

In emulation mode, this feature enables the keyboard to send 3270 keyboard codes to the host according to a configuration table set up during installation.

keyword spotting

A capability provided by WholeWord speech recognition, FlexWord speech recognition, and Natural Language speech recognition that allows the system to recognize a single word in the middle of an entire phrase spoken by a caller in response to a prompt.

L

LAN

local area network

LDB

local database

LED

light-emitting diode

library states

The state information about channel activities maintained by the IRAPI.

LIFO

last-in-first-out processing order

line side E1

A digital method of interfacing a system to a PBX or switch using E1-related hardware and software.

line side T1

A digital method of interfacing a system to a PBX or switch using T1-related hardware and software.

listfile

An ASCII catalog that lists the contents of one or more talkfiles. Each application script is typically associated with a separate listfile. The listfile maps speech phrase strings used by application scripts into speech phrase numbers.

local area network

A data communications network in a limited geographical area. The LAN provides communications between computers and peripherals.

local database

A database residing on the system.

LOG

System logger process message class

logical unit

A type of SNA Network Addressable Unit.

logdaemon

A UNIX system information and error logging process.

logger

See logdaemon.

logging on/off

Entering or exiting the system software.

LSE1

line side E1

LST1

line side T1

LU

logical unit

M

magnetic peripherals

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

main screen

The system screen from which you are able to enter either the System Administration or Voice System Administration menu.

maintenance process

A software process that runs temporary diagnostics and maintains the state of circuit cards and channels.

manoos

manually out-of-service state

masked event

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

master

A circuit card that provides clock information to the TDM bus.

Mbps

megabits per second

MB

megabyte

megabyte

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

menu

Options presented to a user on a computer screen or with voice prompts.

MF

multifrequency

MHz

megahertz

mirroring

A method of data backup that allows all of the data transactions to the primary hard disk drive to be copied and maintained on a second identical drive in near real time. If the primary disk drive fails or becomes disabled, all of the data stored on it (up to 1.2 billion bytes of information) is accessible on the second mirrored disk drive.

ms

millisecond

msec

millisecond

MS-DOS

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

MTC

maintenance process

multifrequency

Dual tone digit signaling (similar to DTMF), used for trunk addressing between network switches or by network operators.

multithreaded application

A single process or application that controls several channels. Each thread of the application is managed explicitly. Typically this means state information for each thread is maintained and the state of the application on each channel is tracked.

N

Natural Language Speech Recognition (NLSR)

An advanced type of speech recognition. Like WholeWord and Flexword speech recognition, NLSR can recognize particular words and phrases, but it can also interpret and assign meaning to those words and phrases. NLSR can also recognize natural numbers and currency amounts. Because of the greater vocabulary and grammar requirements associated with NLSR, it works best with an external speech recognition or "proxy" server.

NCP

Network Control Program

NEBS

Network Equipment Building Standards

NEMA

National Electrical Manufacturers Association

netoos

network out-of-service state

NetView

An optional feature package that transmits high-priority (major or critical) messages to the host as operator-generated alerts (OGAs) over the 3270 host link. The NetView Alarm feature package does not require a dedicated LU.

NFAS

non-facility associated signaling

NFS

network file sharing

NM-API

Network Management - Application Programming Interface

NMVT

network management vector transport

nonex

nonexistent state

nonindexed table

A table that can be searched only in a sequential manner and not via a field name.

nonmasked event

An event that must be sent to the application. Generally, an event is nonmaskable if the application is likely to encounter state transition errors by trying to ignore it.

NRZ

non return to zero

NRZI

non return to zero inverted

null value

An entry containing no value. A field containing a null value is normally displayed as blank and is different from a field containing a value of zero.

O

OEM

original equipment manufacturer

OGA

operator-generated alert

online help

Messages or information that appear on the user’s screen when a function key (F1 through F8) is pressed or a "Help" menu item or icon is clicked.

operator-generated alert

A system-monitoring message that is transmitted from the CONVERSANT system or other computer system to an IBM host computer and is classified as critical or major.

option

An argument used in a command line to modify program output by modifying the execution of a command. When you do not specify any options, the command executes according to its default options.

ORACLE

A company that produces relational database management software. It is also used as a generic term that identifies a database residing on a local or remote system that is created and maintained using an ORACLE RDBMS product.

P

P&C

Prompt and Collect Voice@Work node or Script Builder action step

PBX

private branch exchange

PC

personal computer

PCB

printed circuit board

PCI

peripheral component interconnect

PCI Mezzanine Card

A PCI module, usch as a LAN or RAID controller, that connects to the CPU Complex IOB companion card.

PCM

pulse code modulation

PEC

price element code

peripheral (device)

Equipment such as printers or terminals that is in addition to the basic processor.

peripheral component interconnect

A newer, higher speed PC bus that is gradually displacing ISA for many components.

permanent process

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

phoneme

A single basic sound of a particular spoken language. For example, the English language contains 40 phonemes that represent all basic sounds used with the language. The English word “one” can be represented with three phonemes, “w” - “uh” - “n.” Phonemes vary between languages because of guttural and nasal inflections and syllable constructs.

phrase

A set of one or more words used within an application. Examples include "Thank you for calling XZY Business," "One," and "At the tone, press—."

phrase filtering (screening)

The rejection of unrecognized speech. The WholeWord, FlexWord, and Natural Language speech recognition packages can be programmed to reprompt the caller if the system does not recognize a spoken response.

phrase number

An identification number associated with a particular phrase in a speech pool.

phrase tag

A string of up to 50 characters that identifies the contents of a speech phrase used by an application script.

platform migration

See platform upgrade.

platform upgrade

The process of replacing the existing platform with a new platform.

pluggable

A term usually used with speech technologies, in particular standard speech, to indicate that a basic algorithmic technique has been implemented to accept one or more sets of parameters that tailors the algorithm to perform in one or more languages.

PMC

PCI Mezzanine Card

poll

A message sent from a central controller to an individual station on a multipoint network inviting that station to send if it has any traffic.

polling

A network arrangement whereby a central computer asks each remote location whether it wants to send information. This arrangement enables each user or remote data terminal to transmit and receive information on shared facilities.

port

A connection or link between two devices that allows information to travel to a desired location. See telephone network connection.

PRI

Primary Rate Interface

Primary Rate Interface

An ISDN term for connections over E1 or T1 facilities that are usually treated as trunks.

private branch exchange

A private switching system, either manual or automatic, usually serving an organization, such as a business or government agency, and usually located on the customer’s premises.

processor

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

prompt

A message played to a caller that gives the caller a choice of selections in a menu and asks for a response. Compare to announcement.

prompt and collect (P and C)

A message played to a caller that gives the caller a choice of selections in a menu and asks for a response. The response is collected and the script progresses based on the caller’s response.

proxy server

A server external to the CONVERSANT system used in a client/server configuration to perform processor-intensive functions, such as Natural Language Speech Recognition or text-to-speech beyond the capabilities of the CONVERSANT system. See also Natural Language Speech Recognition (NLSR) and Proxy Text-to-Speech (PTTS).

Proxy Text-to-Speech (PTTS)

The capability to do text-to-speech processing using one or more auxiliary computers that are connected to the CONVERSANT in a client/server configuration. PTTS is an alternative to the standard Text-toSpeech feature for use in applications where the demand is very high or where a language is needed that is not supported on the SSP circuit card. See also Text-to-Speech.

pseudo driver

A driver that does not control any hardware.

PSTN

public switch telephone network

pulse code modulation

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

R

RAID

redundant array of independent disks

RAID array

An assembly of disk drives configured to provide some level of RAID functionality.

RAM

random access memory

RDMBS

ORACLE relational database management system

RECOG

speech recognition feature message class

recognition type

The type of input the recognizer can understand. Available types include touchtone, dial pulse, and Advanced Speech Recognition (ASR), which includes WholeWord, FlexWord, and Natural Language speech recognition.

recognizer

The part of the system that compares caller input to a grammar to correctly match (identify) the caller input.

record

See database record.

recovery

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

remote database

Information stored on a system other than your current system that can be accessed by the CONVERSANT system.

remote maintenance circuit card

A CONVERSANT system circuit card, available with a built-in modem, that allows remote personnel (for example, field support) to access all CONVERSANT system machines. This card is standard equipment on all new purchases.

REN

ringer equivalence number

reports administration

The component of the system that provides access to system reports, including call classification, call data detail, call data summary, message log, and traffic reports.

restore

The process of recovering lost or damaged files by retrieving them from available backup tapes or from another disk device. See also recovery.

restore application

A utility that replaces a damaged application or restores an older version of an application.

reuse

The concept of using a component from a source system in a target system after a software upgrade or platform migration.

RFS

remote file sharing

RM

resource manager

RMB

remote maintenance circuit card

roll back

To cancel changes to a database since the point at which changes were last committed.

rollback segment

A portion of the database that records actions that should be undone under certain circumstances. Rollback segments are used to provide transaction rollback, read consistency, and recovery.

RTS

request to send

S

SBC

(1) sub-band coding; (2) a single-board computing circuit card that is part of the CPU Complex

SCA

single connector architecture

screen pop

A method of delivering a screen of information to a telephone operator at the same time a telephone call is delivered. This is accomplished by a complex chain of tasks that include identifying the calling party number, using that information to access a local or remote ORACLE database, and pulling a “form” full of information from the database using an ORACLE database utility package.

script

The set of instructions for the CONVERSANT system to follow during a transaction.

Script Builder

An optional software package that provides a menu-oriented interface designed to assist in the development of custom voice response applications on the CONVERSANT system (see also Voice@Work).

SCSI

small computer system interface

SDLC

synchronous data link control

SDN

software defined network

shared database table

A database table that is used in more than one application.

shared speech

Speech that is a part of more than one application.

shared speech pools

A parameter that allows the user of a voice application to share speech components with other applications.

SID

station identification

signal processor circuit card

A speech processing circuit card that is an older, lower-capacity version of the speech and signal processor (SSP) circuit card.

single-threaded application

An application that runs on a single voice channel.

slave

A circuit card that depends on the TDM bus for clock information.

SLIP

serial line interface protocol

small computer system interface

A disk drive control technology in which a single SCSI adapter circuit card plugged into a PC slot is capable of controlling as many as seven different hard disks, optical disks, tape drives, and so on.

SNA

systems network architecture

SNMP

simple network management protocol

software

The set or sets of programs that instruct the computer hardware to perform a task or series of tasks, for example, UnixWare software and the system software.

software upgrade

The installation of a new version of software in which the existing platform and circuit cards are retained.

source system

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

speech and signal processor circuit card

A high-performance signal processing circuit card capable of simultaneous support for various speech technologies.

speech energy

The amount of energy in an audio signal. Literally translated, it is the output level of the sound in every phonetic utterance.

speech envelope

The linear representation of voltage on a line. It reflects the sound wave amplitude at different intervals of time. This envelope can be plotted on a graph to represent the oscillation of an audio signal between the positive and negative extremes.

speech file

A file containing an encoded speech phrase.

speech filesystem

A collection of several talkfiles. The filesystem is organized into 16-KB blocks for efficient management and retrieval of talkfiles.

speech modeling

The process of creating WholeWord speech recognition algorithms by collecting thousands of different speech samples of a single word and comparing them all to obtain a statistical average of the word. This average is then used by a WholeWord speech recognition program to recognize a single spoken word.

speech space

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

speech phrase

A continuous speech segment encoded into a digital string.

speech recognition

The ability of the system to understand input from callers.

speech recognition engine

See engine.

SPIP

signal processor interface process

SPPLIB

speech processing library

SQL

structured query language

SR

speech recognition

SSP

speech and signal processor circuit card

standard speech

The speech package available in several languages containing simple words and phrases produced by Avaya for use with the system. This package includes digits, numbers, days of the week, and months, each spoken with initial, medial, and falling inflection. The speech is in digitized files stored on the hard disk to be used in voice prompts and messages to the caller. This feature is also called enhanced basic speech.

standard vocabulary

A standard package of simple word speech models provided by Avaya and used for WholeWord speech recognition. These phrases include the digits “zero” through “nine,” “yes,” “no,” and “oh,” or the equivalent words in a specific language.

string

A contiguous sequence of characters treated as a unit. Strings are normally bounded by white spaces, tabs, or a character designated as a separator. A string value is a specified group of characters symbolized by a variable.

structured query language

A standard data programming language used with data storage and data query applications.

subword technology

A method of speech recognition used in FlexWord recognition that recognizes phonemes or parts of words. Compare to WholeWord speech recognition.

switch

A software and hardware device that controls and directs voice and data traffic. A customer-based switch is known as a private branch exchange.

switch hook

The device at the top of most telephones that is depressed when the handset is resting in the cradle (in other words, is *on hook*). The device is raised when the handset is picked up (in other words, when the telephone is *off hook*).

switch hook flash

A signaling technique in which the signal is originated by momentarily depressing the switch hook.

switch interface administration

The component of the system that enables you to define the interaction between the system and switches by allowing you to establish and modify switch interface parameters and protocol options.

switch network

Two or more interconnected telephone switching systems.

synchronous communication

A method of data transmission in which bits or characters are sent at regular time intervals, rather than being spaced by start and stop bits. Compare to asynchronous communication.

SYS

UNIX system calls message class

sysgen

system generation

System 75

An advanced digital switch supporting up to 800 lines that provides voice and data communications for its users.

System 85

An advanced digital switch supporting up to 3000 lines that provides voice and data communications for its users.

system administrator

The person assigned the responsibility of monitoring all system software processing, performing daily system operations and preventive maintenance, and troubleshooting errors as required.

system architecture

The manner in which the system software is structured.

system message

An event or alarm generated by either the system or an end-user process.

system monitor

A component of the system that tests to verify that each incoming telephone line and its associated circuit card is functional. Through the “System Monitor” component, you are able to see displays of the Voice Channel and Host Session Monitors.

T

T1

A digital transmission link with a capacity of 1.544 Mbps.

table

See database table.

tag image file format

A format for storing and exchanging digital image data associated with fax modem data transfers and other applications. These files can be identified by the .tif extension.

talkfile

An ASCII file that contains the speech phrase tags and phrase tag numbers for all the phrases of a specific application. The speech phrases are organized and stored in groups. Each talkfile can contain up to 65,535 phrases, and the speech filesystem can contain multiple talkfiles.

talkoff

The process of a caller interrupting a prompt, so the prompt message stops playing.

target system

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

TAS

transaction assembler script

TCC

Technology Control Center

TCP/IP

transmission control protocol/internet protocol

TDM

time division multiplexing

TE

terminal emulator

telephone network connection

The point at which a telephone network connection terminates on a system. Supported telephone connections are T1 and E1.

terminal emulator

Software that allows a PC or UNIX process to look like a specific type of terminal. In particular, it allows the system to temporarily transform itself into a “look alike” of an IBM 3270 terminal. In addition to providing full 3270 functionality, the terminal emulator enables you to transfer files to and from UNIX.

Text-to-Speech

An optional feature that allows an application to play US English speech directly from ASCII text by converting that text to synthesized speech. The text can be used for prompts or for text retrieved from a database or host, and can be spoken in an application with prerecorded speech.

ThickNet

A 10-mm (10BASE5) coaxial cable used to provide interLAN communications.

ThinNet

A 5-mm (10BASE2) coaxial cable used to provide interLAN communications.

TIFF

tag image file format

time-division multiplex

A method of serving a number of simultaneous channels over a common transmission path by assigning the transmission path sequentially to the channels, with each assignment being for a discrete time interval.

token ring

A ring type of local area network that allows any station in the network to communicate with any other station.

trace

A command that can be used to monitor the execution of a script.

traffic

The flow of information or messages through a communications network for voice, data, or audio services.

transaction

The interactions (exchanges) between the caller and the voice response system. A transaction can involve one or more telephone network connections and voice responses from the system. It can also involve one or more of the system optional features, such as speech recognition.

transaction assembler script

The computer program code that controls the application operating on the voice response system. The code can be produced from Voice@Work, Script Builder, or by writing directly in TAS code.

transaction state machine process

A multi-channel IRAPI application that runs applications controlled by TAS script code.

transient process

A process that is created dynamically only when needed.

troubleshooting

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

TSO

(1) Technical Services Organization; (2) time share operation

TSM

transaction state machine process

TTS

Text-to-Speech

TWIP

T1 interface process

U

UK

United Kingdom

US

United States of America

UNIX operating system

A multiuser, multitasking computer operating system originally developed by Lucent Technologies.

UNIX shell

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

upgrade scenario

The particular combination of current hardware, software, application and target hardware, software, applications, and so on.

usability

A measurement of how easy an application is for callers to use. The measurement is made by making observations and by asking questions. An application should have high usability to be successful.

USOC

universal service ordering code

UVL

unified voice library

V

VDC

video display controller

vi editor

A screen editor used to create and change electronic files.

virtual channel

A channel that is not associated with an interface to the telephone network (T1, LSE1/LST1, or PRI). Virtual channels are intended to run “data-only” applications which do not interact with callers but may interact with DIPs. Voice or network functions (for example, coding or playing speech, call answer, origination, or transfer) will not work on a virtual channel. Virtual channel applications can be initiated only by a “virtual seizure” request to TSM from a DIP.

vocabulary

A collection of words that the system is able to recognize using either WholeWord, FlexWord, or Natural Language Speech Recognition.

vocabulary activation

The set of active vocabularies that define the words and wordlists known to the FlexWord recognizer.

vocabulary loading

The process of copying the vocabulary from the system where it was developed and adding it to the target system.

Voice@Work

An optional software package that provides a graphical interface to assist in the development of voice response applications on the system (see also Script Builder).

voice channel

A channel that is associated with an interface to the telephone network (T1, E1, LSE1/LST1, or PRI). Any system application can run on a voice channel. Voice channel applications can be initiated by being assigned to particular voice channels or dialed numbers to handle incoming calls or by a “soft seizure” request to TSM from a DIP or the **soft_szr** command.

voice processing co-marketer

A company licensed to purchase voice processing equipment to sell based on their own marketing strategies.

voice response output process

A software process that transfers digitized speech between system hardware (for example, SSP circuit cards) and data storage devices (for example, hard disk, and so on).

voice response unit

A computer connected to a telephone network that can play messages to callers, recognize caller inputs, access and update a databases, and transfer and monitor calls.

voice system administration

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

VPC

voice processing co-marketer

VROP

voice response output process

VRU

voice response unit

W

warning

An admonishment or advisory statement used in the system documentation to alert the user to the possibility of equipment damage.

WholeWord speech recognition

An optional feature, available in several languages, based on whole-word technology that can recognize the numbers one through zero, “yes”, and “no” (the key words). This feature is reliable, regardless of the individual speaker. This feature can identify the key words when spoken in phrases with other words. A string of key words, called *connected digits*, can be recognized. During the prompt announcement, the caller can speak or use touchtones (or dial pulses, if available). See also whole-word technology.

whole-word technology

The ability to recognize an entire word, rather than just the phoneme or a part of a word. Compare to subword technology.

wink signal

An interruption of current to a busy lamp indicating that there is a line on hold.

word

A unique utterance understood by the recognizer.

wordlist

A set of words available for FlexWord recognition by an application during a Prompt & Collect action step.

word spotting

The ability to search through extraneous speech during a recognition.

Index

A

accuracy
- DPR recognition 84
- FlexWord speech recognition 63–66
- WholeWord speech recognition 46–49

Acrobat Reader
- adjusting the window size xxiii
- hiding and displaying bookmarks xxiii
- navigating xxiv
- printing from xxiv
- searching xxiv
- setting the default magnification xxiii

adaptive differential pulse code modulation (ADPCM)
- ADPCM16 140
- ADPCM32 140

add command 2

ADPCM, see adaptive differential pulse code modulation (ADPCM)

allow interrupt, see barge-in

applications
- confirmation steps 65
- developing 15–29
- development tools
  - Script Builder 5
  - Voice@Work 5
- informative prompts 65
- installing 29
- reprompt steps 65
- speech
  - features 4

B

barge-in
- capability for TTS 91
- enabling for TTS in risky mode 166
- enabling or disabling 43
- encouraging and discouraging 43–44
- not supported with DPR 83
- not supported with FlexWord speech recognition 57
- response time compared with talkoff 44

barge-in, (continued)
- response time for WholeWord speech recognition 44
- see also talkoff
- supported with WholeWord speech recognition 42

bilingual speech recognition, see speech recognition

C

call flow, see transaction

CELP, see code excited linear prediction

code excited linear prediction (CELP) 139, 146

connected-digits
- and speech recognition 40
- string length impact on recognition accuracy 49

conservative mode, see Text-to-Speech (TTS)

copy command 2

custom
- DIPs for WholeWord speech recognition 45
- phrase tags 3
- vocabulary
  - WholeWord speech recognition 45

D

data interface process (DIP) 167
- custom for WholeWord speech recognition 45

DEFINITY switches
- limitations for DPR 85

dial pulse recognition (DPR)
- accuracy 84
- barge-in not supported 83
- capabilities 81
- compared with other recognition types 94
- confirmation of 83
- description 12, 84
- digits 82
- failure modes 86

dial pulse recognition (DPR), (continued)
- grammars 83
- processing 84
- recognizer 84
- requirements
  - hardware 81
  - hardware and software 13
- SSP channel capacity 13
- switch limitations 85
- training 84
- troubleshooting 85
- types 82
- using to best advantage 91
- using with NLSR 87
- using with other recognition types 81
- using with WholeWord speech recognition 88

DIP, see data interface process (DIP)

DPR, see dial pulse recognition

drives (hard disks)
- adding for speech 2
- recovering a corrupted speech disk 2

DTMF, see dual-tone multifrequency

dual tone multifrequency (DTMF)
- support for WholeWord speech recognition 44
- touch tones 44

E

electronic documentation, printing xxiv

encoding formats, see speech

enhanced basic speech (EBS)
- abbreviations 98
- Australian English formats 102
- Brazilian Portuguese formats 104
- Canadian French formats 105
- Cantonese Chinese formats 108
- Castillan Spanish formats 109
- common formats 99
- converting applications between languages 101
- Czech formats 110
- description 3
- Dutch formats 113
- format overview 97
- French formats 114
- gender
  - grammatical 100
  - speaking voice 98
- German formats 116
- Hindi formats 117
- Hungarian formats 119
- Indonesian formats 121

enhanced basic speech (EBS), (continued)
- inflections 100
- Italian formats 122
- Japanese formats 123
- Korean formats 124
- languages 102–138
- Latin-American Spanish formats 125
- Malay formats 127
- Mandarin Chinese formats 128
- maximum values for numbers and currency 101
- phrase pool names 98
- Polish formats 129
- purchasing the package 22
- Slovak formats 132
- speaking phrase numbers and packed talkfile numbers 102
- support for number and currency formats 101
- Thai formats 134
- UK English formats 135
- US English formats 137
- variations on common formats 99

erase command 2

escape sequences, see Text-to-Speech (TTS), escape sequences

F

feature_tst script 85

find_best.c function 167

FlexWord speech recognition
- accuracy 63–66, 94
- choosing a vocabulary 89
- compared to other recognition types 94
- description 10, 51
- designing an application 57–59
- designing prompts for 89
- keyword spotting 56
- languages supported 10, 52–55
- licensing 8
- phrase screening 57
- requirements, hardware and software 10
- SSP channel capacity 13
- synonyms 90
- use in applications 90
- using with NLSR 87
- using with other forms of speech recognition 93
- using with other recognition types 81
- using with Text-to-Speech (TTS) 92
- vocabulary 51
- word 51

FlexWord speech recognition, (continued)
- wordlists 51
  - accented characters 60
  - activating when resident on the system 62
  - copying to diskette for installation 61
  - deactivating 63
  - design guidelines 89
  - effective 64
  - ineffective 66
  - installing and activating from diskette 61
  - maximum number allowed 56
  - moving between active and inactive directories 61
  - name tag characters 56
  - number of entries allowed 56
  - removing 63
  - size 56
  - special characters 60
  - using in scripts 57
  - word choice 64, 89, 90

framing, see voice scripts

G

grammars
- dial pulse recognition 83
- for NLSR 78
- WholeWord speech recognition 45

I

inflections
- recording speech with frame words 27
- types 18

installing speech
- with TSM 29

K

keyword recognition types 38

keyword spotting, see FlexWord speech recognition

L

languages
- Australian English
  - EBS formats 102
  - WholeWord recognition 37, 38, 39
- Brazilian Portuguese
  - EBS formats 104
  - Flexword phonemes 53
  - WholeWord recognition 37, 38, 39

languages, (continued)
- Canadian French
  - EBS formats 105
  - WholeWord recognition 37, 38, 39
- Cantonese Chinese
  - EBS formats 108
- Castilian Spanish
  - EBS formats 109
  - WholeWord recognition 37, 38, 39
- Czech
  - EBS formats 110
- Dutch
  - EBS formats 113
  - WholeWord recognition 37, 38, 39
- French
  - EBS formats 114
  - FlexWord phonemes 53
  - WholeWord recognition 37, 38, 39
- German
  - EBS formats 116
  - FlexWord phonemes 54
  - WholeWord recognition 37, 38, 39
- Hindi
  - EBS formats 117
- Hungarian
  - EBS formats 119
- Indonesian
  - EBS formats 121
- Italian
  - EBS formats 122
  - WholeWord recognition 37, 38, 39
- Japanese
  - EBS formats 123
  - FlexWord phonemes 54
  - WholeWord recognition 37, 39
- Korean
  - EBS formats 124
- Latin-American Spanish
  - EBS formats 125
  - FlexWord phonemes 55
  - WholeWord recognition 37, 39
- Malay
  - EBS formats 127
- Mandarin Chinese
  - EBS formats 128
- Polish
  - EBS formats 129
- Slovak
  - EBS formats 132
- Thai
  - EBS formats 134
- UK English
  - EBS formats 135
  - WholeWord recognition 37, 39

languages, (continued)
- US English
  - EBS formats 137
  - FlexWord phonemes 55
  - TTS formats 150
  - WholeWord recognition 37, 39

licensing
- FlexWord speech recognition 8
- TTS and PTTS 8
- WholeWord speech recognition 8

list command 2

ln command 62

M

merge command 168

Mu-law PCM format 139

N

natural language speech recognition (NLSR)
- compared with other recognition types 94
- configuring
  - hardware 77
  - software 77
- creating and testing applications 78
- deploying applications 79
- description 11, 67
- designing applications 78
- developing grammars for 78
- offer architecture 69
  - minimum configuration model 69
  - multiple-client/multiple-server model 73
  - single-client/multiple-server model 71
- overview of offer 68
- proxy servers with 68
- requirements, software and hardware 12, 75–76
- tasks for using 76
- using to best advantage 90
- using with other features 87
- using with other forms of speech recognition 93
- using with other recognition types 81

NLSR, see natural language speech recognition (NLSR)

P

pack_phrNX function 102

PCM, see pulse code modulation

phrase screening 41, 57

prerecorded speech, using with TTS 92

prompts
- for FlexWord speech recognition 89
- informative 65
- structure and design 64

proxy server, defined 68

Proxy Text-to-Speech (PTTS) 91
- description 7
- languages supported 7
- licensing 8
- requirements, hardware and software 8
- see also Text-to-Speech (TTS)

PTTS, see Proxy Text-to-Speech (PTTS)

pulse code modulation (PCM) 139

R

recognition confirmation
- DPR 83
- WholeWord 42

recognition DIPs 41

recognition post-processing
- close match 171
- expected value 171
- find_best.c function 167
- Luhn check 170
- prefix list check 170
- recog_dip process 167
- return top 4 169

restoring speech files 2

risky mode, see Text-to-Speech (TTS)

S

Script Builder
- application development tool 5
- editing speech files 31
- installing speech 29
- overview of producing speech 5
- predefined phrase tags 3
- recording speech 21
- speech encoding 28

speech
- adding a second disk for 2
- custom
  - developing with Avaya 27
  - purchasing 21
- development methods 20–24
- editing
  - in Script Builder 31
  - in Voice@Work 31

speech, (continued)
- encoding
  - identifier codes 147
  - using Avaya services 28
  - using Script Builder 28
  - using Voice@Work 28
- encoding formats
  - overview 139
  - phrase blocks 1
  - rates 1
- encoding phrases 28
- features 4
- files
  - default directory 2, 4
  - description 146
  - formats 139
  - restoring 2
  - saving and restoring 2
  - storage space requirements 1
  - talkfiles 2, 4
- filesystem
  - structure 1, 4
- importing from other applications 24
- inflections 18
- installing 29
- phrases 15
- processing 15
- purchasing EBS package 22
- recording
  - alphabetic characters 26, 27
  - environmental conditions 25
  - equipment specifications 25
  - getting more natural sounding results 26
  - in Script Builder or Voice@Work 21
  - independently yourself 28
  - inflections 27
  - numbers 26, 27
  - options 25
  - professional speakers 20
  - with professional speakers 25
- recovering a corrupted speech disk 2
- sharing 23
- talkfiles 4
- using Text-to-Speech 22

speech and signal processor (SSP) circuit cards
- description 13
- DPR 13
- FlexWord speech recognition 10
- speech channel capacities 13
- TTS 13, 159
- WholeWord speech recognition 9, 36

speech recognition
- bilingual capabilities 36, 52
- comparison of types 94
- overview 8
- see also DPR, FlexWord speech recognition, natural language speech recognition (NLSR), and WholeWord speech recognition

SPEECHDIR variable 2

spres command 2

spsav command 2

standard vocabulary, see WholeWord speech recognition

switches
- central office limitations for DPR 85
- DEFINITY limitations for DPR 85

synthesized speech, see Text-to-Speech (TTS)

systems
- online help support xxi

T

talkfiles, see speech

talkoff
- DTMF support 44
- response time compared with barge-in 44
- see also barge-in

Text-to-Speech (TTS)
- advanced features 153–166
- conservative and risky mode 161
- customizing 23
- delays and pauses in 154
- description 5, 23
- enabling barge-in in risky mode 166
- escape sequences 153–166
- format overview 149
- format set 150
- gender of speaking voice 157
- process 22
- rate of speech 158
- requirements, hardware and software 7
- text classes and detectors 160
- tts_file external function 6
- uses 6
- using to best advantage 91
- using with FlexWord speech recognition 91, 92
- using with prerecorded speech 92
- using with touchtone input 91

transaction 16

TTS, see Text-to-Speech

tts_file external function 6

U

unpack_phrNX function 102

V

vdf command 2

vocabulary
- Flexword speech recognition 51
- WholeWord speech recognition 9, 34, 45

voice scripts
- analyzing speech inflections 18
- description 16
- framing
  - description 18
  - example 18
  - placing frame words 19
  - selecting speech sounds 19
  - using voiceless speech sounds 19
- framing sentences 26
- planning 16
- types of phrases 17
- writing 18

Voice@Work
- application development tool 5
- editing speech files 31
- installing speech 29
- overview of producing speech 5
- recording speech 21
- speech encoding 28

W

WholeWord speech recognition
- accents, regional and national 48
- accuracy 46, 94
- barge-in 42, 44
- bilingual recognition 36
- compared to other recognition types 94
- confirmation 42
- custom
  - DIPs 45
  - grammars 45
  - vocabulary 9, 45
- custom DIPs for 45
- description 9, 34
- dialects 48
- digits, fixed and variable length connected 39–40
- DTMF support 44
- grammars 38
- languages supported 9, 36
- licensing 8
- models 38
- phrase screening 41
- processing 34–35
- requirements, hardware and software 9
- SSP channel capacity 13
- standard vocabulary 9, 36
- types 37
- using with dial pulse recognition (DPR) 88
- using with NLSR 87
- using with other forms of speech recognition 93
- using with other recognition types 81
- word spotting 41

wl_copy command 61

wl_gen command 61, 62, 63

wl_install command 61

word spotting, see WholeWord speech recognition

wordlists, see FlexWord speech recognition