CONVERSANT® System

Version 8.0

Application Design Guidelines

585-313-226
Issue 2
Comcode 108755026
December 2001

Copyright and Legal Notices

Acknowledgment

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

- 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。

Toll Fraud

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

Your Responsibility for Your System's Security

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

Avaya Fraud Intervention

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

Documentation Ordering Information

To order a document, contact Avaya and 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

Globalware Solutions
200 Ward Hill Avenue
Haverhill, MA 01835 USA

email: totalware@gwsmail.com

World Wide Web

Use a web browser to reach the following site:

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 the Avaya Publications Center (see Write, Call, or Fax (page vii)).

Contents

Copyright and Legal Notices iii

About This Book xiii

Purpose . . . xiii
Intended Audiences . . . xiii
Release History . . . xiii
Trademarks . . . xiii
How to Use This Book . . . xiv
Conventions Used in This Book . . . xiv
- Key Presses . . . xiv
- Screen Displays . . . xv
- Typography . . . xvi
- Safety and Security Alert Labels . . . xvi

CONVERSANT System Online Help . . . xvii
Technical Assistance . . . xvii
Related Resources . . . xviii
- Documentation . . . xviii
- Training . . . xix

Using the CD-ROM Documentation . . . xx
How to Comment on This Book . . . xxii

1 Introduction to Voice Response Application Design 1

Overview . . . 1
Purpose . . . 1
What is a Voice Response System? . . . 1
- What is a Voice Response Application? . . . 1
- Voice Response System Versus An Attendant . . . 3

Defining Successful Voice Response Applications . . . 4
- Defining Measurable Objectives . . . 4

Important Terminology . . . 5

2 Voice Response Advanced Technologies 9

Overview . . . 9
Purpose . . . 9
Touchtone and Dial Pulse Recognition . . . 9
- Touchtone Recognition . . . 9
- Dial Pulse Recognition . . . 10

Speech Recognition 11
WholeWord Speech Recognition 11
FlexWord Speech Recognition 14
Natural Language Speech Recognition 17
Speech Recognition Accuracy Measurement 18
Text-to-Speech 19
Text-to-Speech Uses 19
Text-to-Speech Capabilities 19
Text-to-Speech Accuracy 20
Proxy Text-to-Speech 20
PTTS Languages 20
PTTS Capabilities 21
FAX Actions 21
FAX Actions Uses 21
FAX Actions Capabilities 22

3 Planning a Voice Response Application 23

Overview 23
Purpose 23
Plan Your Application Design 23
Use Human Factors/Usability Engineering Resources 23
Use Prerecorded Speech 24
Offer a User Guide 25
Offer Caller Training 26
Provide an Attendant 26
Plan for Disabled Callers 26
Diagram Your Application Design 27

4 Designing a Voice Response Application 31

Overview 31
Purpose 31
Application Design Research 31
Know Your Callers 31
Use Simple and Natural Dialog 32
Minimize Demands on the Caller's Memory 32
Be Consistent 33
Provide Feedback 33
Provide Easy Exits 33
Offer Shortcuts 33
Prevent Errors 33
Prompts 34
Prompt Length 34
How to Word Prompts 35

Announcements 36
Feedback Announcements 36
Confirmation Announcements 36
Menus 39
Number of Menu Choices. 39
Menu Choice Sequence 40
Numbered Menu Options 40
Subdivided Menu Options 40
Digit Input 41
Constant-Length Digit Sequences 41
Variable-Length Digit Sequences 41
Entering Digit Sequences. 42
Validate a Digit Sequence Entry 43
Confirming Digit Entries with Callers 44
Yes/No Questions 45
Touch-Tone Input for Yes/No Questions 45
Spoken Input for Yes/No Questions 45
Yes/No Questions with Barge-in (WholeWord Speech Recognition) 45
Yes/No Question Without Barge-in 47
Pace the Application 48
When Callers Must Wait 48
Allow Time for Caller Responses 48
Adjustable Pacing. 49
Application Errors 50
When Caller Errors Occur 50
When Speech Recognition Errors Occur 50
How to Word Error Messages 51
Touch-Tone and Speech Recognition 52
Recognizer Differences Between Touch-Tone Input and Spoken Input 52
Application Differences Between TouchTone Input and Spoken Input 53
TouchTone Input Used with Spoken Input 54
Use Touch-Tone Input When Speech Recognition Fails 59
Modify a Touch-Tone-Only Application to Include Spoken Input. 60
Dial Through and Barge-in 61
Using Dial Through and Barge-in with Errors Messages 61
Using Dial Through and Barge-in Consistently 61
How to Word Prompts for Dial Through and Barge-in. 62
Dial Pulse Recognition in Applications 63
Dial Pulse Recognition Training 63
Dial Pulse Recognition Digit Input 63
Dial Pulse Recognition and Barge-in 63

FlexWord Speech Recognition in Applications 64
How to Choose Words for Your FlexWord Vocabulary 64
How to Segment Large Wordlists 65
How to Word Prompts for FlexWord Speech Recognition 66
Bilingual, Multilingual, and Non-US English Applications 66
Bilingual Applications 66
Multilingual Applications 70
Text-to-Speech in Applications 70
Use Text-to-Speech for Prompts and Announcements 71
Use Both Text-to-Speech and Prerecorded Speech Prompts and Announcements 71
How to Get the Most Out of Text-to-Speech 72
How to Test Text-to-Speech Applications 76
Using FAX Actions in Applications 77
Informing Callers of Available Information 77
Managing Delays 77
Assuring Faxes Are Sent Successfully 78
Exec_UNIX in Non-FAX Applications 78

5 Testing and Using a Voice Response Application Design 79

Overview 79
Purpose 79
Testing the Application 79
Plan a Preliminary Application Design 79
Set Usability Goals 80
Test the Preliminary Application 80
Refine the Preliminary Application Based on Usability Goals 83
Test the Application Again 83
Continue Testing Until the Usability Goals are Accomplished 83
Use the Tested and Refined Application in Service 84

Index 85

About This Book

Purpose

This book, *CONVERSANT System Application Design Guidelines,* 585-313-226, is an introduction to voice response technology, including advanced-technology features of the CONVERSANT system, application design principles, and important design guidelines.

Intended Audiences

The primary audiences for this document include:

- Application developers
- Readers interested in voice response technology
- Reader interested in application design principles

Release History

This is the second release of this book for the current release of the product.

Trademarks

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

- AUDIX, CONVERSANT, DEFINITY, and Voice Power are registered trademarks and FlexWord are trademarks of Avaya.
- 5ESS is a trademark of Lucent Technologies.
- 3M is a trademark of Minnesota Mining and Manufacturing.
- Phillips is a registered trademark of the Phillips Screw Co.
- Microsoft, MS, MS-DOS, Internet Explorer, and Excel are registered trademarks and Windows is a trademark of Microsoft Corporation.
- Truevoice is a registered trademark of AT&T.
- UnixWare is a registered trademark of The Santa Cruz Operation, Inc.
- UNIX is a registered trademark of UNIX System Laboratories, Inc.
- Novell is a registered trademark of Novell, Inc.

- ORACLE, ORACLE*Terminal, OBJECT*SQL, SQL*FORMS, SQL*Menu, SQL*Net, SQL*Plus, PRO*C, and SQL*ReportWriter are trademarks of the Oracle Corporation.
- IBM and VTAM are registered trademarks of International Business Machines Corporation.
- CLEO and LINKix are trademarks of CLEO Communications.
- Hayes and Smartmodem are trademarks of Hayes Microcomputer Products, Inc.
- Ethernet is a trademark of Xerox Corporation.
- Netscape Navigator is a trademark of Netscape Communications Corporation.

How to Use This Book

This book presents a general discussion of voice response technology and the best methods to use to design applications for the CONVERSANT system. This book contains useful guidelines and suggestions, but does not contain specific procedures to follow to develop applications. For information on application development, see "Related Resources" in this chapter.

Conventions Used in This Book

This section describes the conventions used in this book.

Key Presses

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

To record a message, press **1**.

Screen Displays

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

  Script Builder example

  Example 1:

  The grammars provided can be seen in the Choices menu for the `Recognition Type:` field on the Prompt & Collect screen.

  Example 2:

```
Prompt & Collect
   Prompt
      Speak with Interrupt
         Phrase: “Please enter your 5-digit
         customer number.”
```

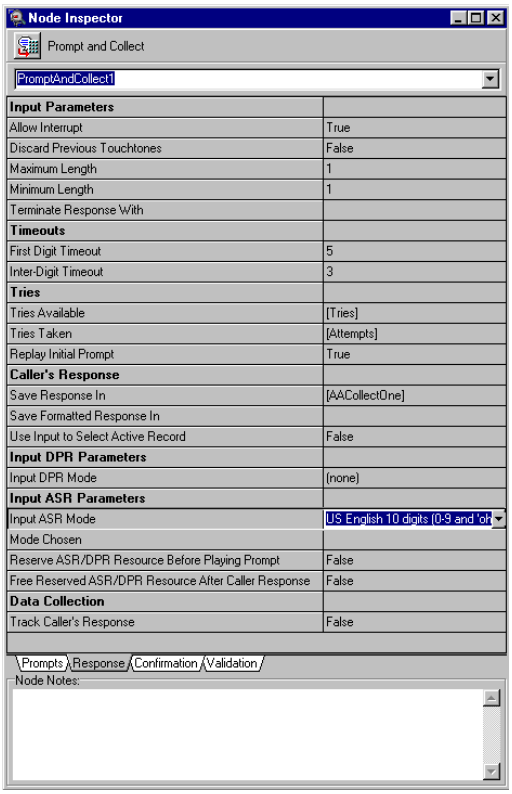
  Voice@Work examples:

  Example 1:

  The grammars provided can be seen in the drop down list for the Input ASR Mode field on the Response Tab in the Prompt and Collect node

  Example 2:

Figure 1. Node Inspector--Prompt and Collect



Typography

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

  Example 1:

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

  Example 2:

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

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, or service interruptions:

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.

CONVERSANT System Online 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 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.

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

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

Documentation

Note: The *CONVERSANT System Version 8.0 System Description*, 585-313-219, contains a detailed description of all books included in V8.0 CONVERSANT documentation library.

Always refer to the appropriate book for specific information on installing, administering, or maintaining a CONVERSANT system.

Application Development Documentation

To develop an application for your CONVERSANT system, use the following documentation:

- *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217
- *CONVERSANT® System Version 8.0 Application Development with Advanced Methods*, 585-313-216
- *CONVERSANT® System Version 8.0 Speech Development, Processing, and Recognition*, 585-313-218
- *CONVERSANT® System Version 8.0 Communication Development*, 585-313-220
- *Using Voice@Work,* 585-313-207

Additional Suggested Documentation

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

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

It is recommended that you access the following sites for additional information.

- UnixWare 7.1 documentation: http://www.sco.com/documentation/
- Update to CONVERSANT documentation: http//support.avaya.com/elmodocs2/conversant/index.jhtml

Obtaining Printed Versions of the Documentation

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

Training

To obtain training on the CONVERSANT product, call 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 systems
  - ~ BTE501W, CONVERSANT Administration for Technicians
  - ~ BTE502H, CONVERSANT Installation and Maintenance
- For technicians and administrators
  - ~ BTC344M, CONVERSANT 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

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: When viewing documents online, it is recommended that you use a separate 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 select **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 purchasing printed copies of the documents, see Obtaining Printed Versions of the Documentation (xviii).

If you would like 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. The document prints.
5. Close the file. Do not leave this file open while viewing the electronic documents.

Printing Part of a Document

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

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

   The document prints, one screen per side, two sides per sheet.

How to Comment on This Book

We are interested in your suggestions for improving this book. Please complete and return the reader comment card that is located at the back of the book.

1 Introduction to Voice Response Application Design

Overview

This chapter provides an introduction to the concept of voice response.

Purpose

The purpose of this chapter is to introduce you to the concept of voice response, including voice response applications and voice response application design.

What is a Voice Response System?

A voice response system consists of hardware and software that provides either full or partial automation of telephone transactions that would otherwise be performed by an operator/attendant, an employee, or a call center agent. The CONVERSANT system not only provides the basic voice response technology to automate telephone transactions, but also provides the mechanism needed to develop the software that guides such transactions. This software is known as an *application*.

What is a Voice Response Application?

An application is a set of instructions written for the voice response system that informs it how to carry out the automated transaction. Applications define the call flow and determine what callers hear and how callers respond to the system.

Voice Response Application Uses

Voice response applications can be used in a wide variety of situations. The following examples can help you determine some possibilities for the best use of voice response applications for your business.

Voice Response for Repetitive Information Requests

Repetitive information requests that can be fulfilled with a voice response application include:

- Interest rates, foreign currency exchange rates, and loan rates
- Real estate information on a property with a specific identifier such as a number
- Customer service information such as updates on product maintenance issues and their resolutions
- Status of automobile repairs being performed and their estimated costs
- Registration forms, entry blanks, or tax forms
- Conference/tradeshow information such as list of speakers and schedules
- Air travel information such as departure and arrival times
- Financial statements company shareholders

Other examples include park or recreation locations, branch office and store locations, restaurant locations, bus schedules and rates, theater schedule and rates, concert information, and sporting event times and results.

Voice Response for Database Information Requests

A voice response system can be connected to a system containing enterprise specific information to access or store information in the host. Enterprise specific information includes: published materials, bank accounts, and product descriptions. Database information requests include:

- Bank or investment accounts
- Magazine articles
- News wire stories
- Public records such as deeds or licences
- Telephone directory listings
- Government records
- Catalog item descriptions

Voice Response for Order Processing

You can design a voice response application to accept touchtone input, spoken input, dial-pulse input, or all three, so that callers can use the system to place orders, confirm orders, or check order status. Ordering processing includes:

- Merchandise orders, restocking — A manufacturer calls a supplier to request spare parts. A confirmation of the order can be faxed to the manufacturer.
- Hotel reservations — Callers can register and pay by telephone, then receive a confirmation fax including a map and directions.
- Travel reservations — Callers can reserve a seat and pay over the telephone. Confirmation can be sent by fax.
- Benefits registration — Callers sign up for a combination of benefits, and can request confirmation by fax.
- Brokerage services — Callers can buy and sell stocks and bonds, receiving a fax confirmation of the transaction.
- Talent agency — Callers can request information on actors with particular talents or physical attributes. Information can be spoken out, and photographs and resumes can be faxed to the caller.

Voice Response for Information Gathering

A voice response system can be used to gather information from those who call a designated number. Information gathering includes:

- Consumer or political surveys
- Customer feedback and complaints
- Employee feedback

Voice Response System Versus An Attendant

Do not eliminate all attendants. Even well-designed automated transactions cannot serve every callers' need. Some callers will have problems that cannot be handled by the voice response system. Callers may want and need to speak to a person, so it is important to provide some attendants that are familiar with your business policies and procedures.

Defining Successful Voice Response Applications

Consider multiple criteria when defining success in your voice response applications. You may have to measure some aspects of how you currently do business, and decide on one or more specific areas to target for improvement. Think about why you are interested in automating some of your business interactions. To save time? To save money? To better serve your customers? To free personnel for more challenging tasks? Take your most important reasons for automation, and determine measurable objectives for automation.

Defining Measurable Objectives

For example, if you want to use voice response to better serve your customers, you might choose as an objective to reduce call holding times. A measurable objective might be to reduce call holding times by 50%. You can also figure out the yearly savings expected from reduced holding times, and factor this into your plan.

Here are some examples of other types of measurable objectives that you may want to achieve through the design of your application:

- Reduce your cost per call by 30%.
- Increase call completion rate by 45%.
- Reduce the percentage of callers who get a busy signal by 50%.
- Increase customer satisfaction with the voice response system by 25%.

Remember to estimate the rate of savings for each objective.

Determining your true objectives, as well as how to measure them, will allow you to judge the effectiveness of your voice response applications.

Important Terminology

It is a good idea to become familiar with the some of the terms and concepts presented in the rest of this book. The list below is provided to help facilitate your familiarization.

announcement	Speech played by the system to the caller that informs, but does not instruct the caller to act. Compare to *prompt*.
application	The computer program that defines and controls the voice response transaction between the system and the caller.
barge-In	The capability that allows callers to respond while a prompt is being played. This is similar to dial through for touch tones.
caller	The person who calls for a service, gets connected to the system, and interacts with an application.
connected-digit	A sequence of digits spoken by a caller without intentional or regular pauses in between digits.
dial ahead	The touchtone recognition capability that allows the system to collect touch tones as they are entered by callers, even before they are asked for. The touchtone input is then used in the order in which they were received. This allows callers to respond to more than one prompt at a time, without having to listen to the intermediate prompts.
dial pulse recognition	A method of recognizing caller pulse inputs from a rotary telephone
dial through	The touchtone capability that allows callers to respond while a prompt is being played. The playback of speech ceases and the application responds to the key that was pressed. This is similar to barge-in for WholeWord speech recognition. Also known as *talk off*.
fax	The capability that allows an application to send stored or dynamically created fax messages at the caller's request, or receive incoming faxes from the caller.
FlexWord speech recognition	The optional system capability that recognizes callers spoken input based on matching caller speech to word models fashioned from representations of sub-words (phonemes), the smallest unit of speech.
FlexWord Toolkit	An optional software package that provides a point-and-click, graphical mechanism for you to define custom list of words for use with FlexWord Speech Recognition.
grammar	The set of rules by which speech is recognized by WholeWord speech recognition or dial pulse recognition. For example, a Prompt & Collect statement using the US_1_5 grammar will recognize the words "one," "two," "three," "four," and "five" in US English.

key word	One of the words in a list of words that the system is instructed to recognize at a particular point in the transaction.
menu	A prompt that gives callers a choice of two or more options. For example, “For sales, press or say 1. For service, press or say 2. For an attendant, press or say 0.”
Natural Language speech recognition	A form of speech recognition that can be used to recognize particular words and phrases and also interpret and assign meaning to the speech it recognizes.
phrase screening	The speech recognition capability that decides whether or not a candidate key word is a close enough match to be declared a valid key word.
Prompt & Collect action	The action used to play a prompt to the caller, accept the caller’s response, and go to the appropriate place in the application to process the caller’s request.
prompt	Speech played by the system that instructs the caller to enter information that is part of a Prompt & Collect action. Compare to *Announcement*.
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, provides for speech processing using one or more auxiliary computers connected to the CONVERSANT system in a client/server configuration
recognition	The process within the system that compares caller speech to internal models and returns a match to the application.
recognition type	• For Script Builder: The choices that are associated with the Recognition Type field in the Prompt & Collect action form. • For Voice@Work: The choices that are associated with the Input ASR Mode and Input DPR Mode fields in the Prompt and Collect node form. This value, along with the minimum number of digits and maximum of digits work in conjunction to allow the system to select a grammar to be used for that recognition event. See also *grammar*.
Script Builder	An optional software package that allows you to define and generate voice response applications to run on the CONVERSANT system.
speech, custom	The part of the system speech database that includes application-dependent, prerecorded speech phrases.

speech, enhanced basic	The part of the system speech database that includes prerecorded speech phrases corresponding to numbers, ordinal numbers, days of the week, and months of the year. Applications use these phrases to speak numbers, amounts, dates, and times in a natural-sounding way.
speech, prerecorded	A prompt or announcement that has been recorded by a person.
substitution error	An error made by the FlexWord speech recognition software where it mistakes one word for another.
talk off	See *dial through*.
Text-to-Speech	An optional software package that converts ASCII text into spoken, computer-generated prompts and announcements. This package is supported for United States English, United Kingdom English, and Australian English. See *Proxy Text-to-Speech*.
touchtone	The signal sent when a caller presses any of the 12 keys on a push-button telephone that sends dual tones rather than rotary pulses.
transaction	The exchange of information between the caller and an application. In a typical transaction, the caller dials in to or gets transferred to the system, then the system answers and plays a greeting. The caller enters information in response to spoken prompts and the system speaks information back until the interaction is complete. See also *application*.
usability	The system quality that reflects whether or not callers can learn and use the features successfully. A system with high usability is called *easy to use*, or *usable*.
user Interface	The aspect of the application with which callers interact; includes prompts and announcements from the system to which callers respond using touchtone, dial pulse, or speech.
vocabulary	The set of wordlists associated with a particular FlexWord application.
WholeWord speech recognition	The optional system capability that recognizes the language-specific words (and common synonyms) for the digits 0 through 9, “yes,” and “no”. Recognition is based on matching caller speech to word models fashioned from many samples of people saying each entire, *whole* word.

word	A FlexWord speech recognition vocabulary item consisting of either a single word or a phrase of several words.
word spotting	The speech recognition capability that allows the system to pick out key words from a stream of caller speech, which may include extraneous speech, background noise, or caller noises. Works in conjunction with phrase screening.
wordlist	A set of words used with FlexWord speech recognition that can be recognized by the system.

2 Voice Response Advanced Technologies

Overview

This chapter provides an introduction to the CONVERSANT® system advanced technologies including touchtone and dial pulse recognition, speech recognition (WholeWord, FlexWord, and Natural Language), Text-to-Speech (including Proxy Text-to-Speech), and the FAX Actions.

Purpose

The purpose of this chapter is to describe what each advanced technology does, the types of applications for which each is best suited, and how the technologies can work together.

Touchtone and Dial Pulse Recognition

The CONVERSANT system provides touchtone recognition and dial pulse recognition as two methods for callers to provide nonspoken input to the system.

Touchtone Recognition

Touchtone recognition is a common feature of voice response applications. The majority of the telephones in the United States are equipped with touchtone service, but touchtone telephone availability will be different based on locations. A greater proportion of business locations have touchtone capabilities than do residences. If your callers have touchtone telephones, it is economical and efficient to allow touchtone caller input for most transactions.

Touchtone Recognition Uses

Touchtone input may be used to select choices from a spoken menu. It may also be used to enter numerical data such as credit card numbers or personal identification numbers.

Touchtone Recognition Capabilities

Touchtone recognition recognizes the digits 0 through 9, as well as the asterisk (*) and pound sign (#).

With touchtone recognition, callers have the option to respond with a tone while a prompt is playing. This capability is known as *dial through* (may also be referred to *talkoff*). As soon as the system detects the tone, the prompt stops.

The system also supports *dial ahead*. This capability allows the system to collect touchtone input as it is entered by callers, even before they are prompted for it. The touchtone input is then used in the order in which it was received. This allows callers to respond to more than one prompt at a time, without having to listen to the intermediate prompts. If experienced callers are familiar with the upcoming prompts, they do not have to wait until the next prompt starts playing before pressing the required touchtone buttons.

Touchtone recognition can be used in conjunction with other recognition methods, such as the Dial Pulse Recognition (DPR) software package, as well as the two spoken-input recognition software packages (WholeWord and FlexWord™ speech recognition) discussed later in this chapter.

Touchtone Recognition Accuracy

The CONVERSANT system is very accurate when recognizing touchtone input. However, callers may make mistakes when entering touchtones, and a well designed application must handle caller errors gracefully.

Dial Pulse Recognition

If touchtone service is not widely available in your area or country of interest, you can offer dial pulse recognition to provide non-spoken input to the system.

Dial Pulse Recognition Uses

The DPR software allows callers with rotary telephones, or push-button telephones that generate dial pulses, to interact with system applications. Much like tone-touch input, dial pulse input may used to enter menu choices and numerical data such as a bank account number.

Dial Pulse Recognition Capabilities

The DPR software recognizes the digits 0 through 9, but *does not* recognize the asterisk (*) or pound sign (#).

DPR can be used together in an application with touchtone recognition and with speech recognition (WholeWord or FlexWord).

Dial ahead and dial through are not supported for DPR.

Dial Pulse Recognition Accuracy

DPR is slightly less accurate than touchtone recognition, but it does allow callers to interact with the application without talking to an agent. The accuracy of DPR can be improved, however, by the use of specific application design techniques. These techniques are discussed in greater detail in Chapter 4, Designing a Voice Response Application .

Speech Recognition

It makes good business sense to provide the capabilities offered by the speech recognition software. Speech recognition allows your callers to:

- Speak their responses; a more natural interface for caller input.
- Provide input if they do not have touchtone service.
- Provide input for some types of information, like names, that do not have a natural translation to touchtone input.

WholeWord Speech Recognition

The WholeWord speech recognition software package is used to recognize spoken input of connected digits and yes/no responses.

WholeWord Speech Recognition Uses

WholeWord speech recognition is most successful when it is used to augment a touchtone application to process callers who do not have touchtone telephones.

The best applications first ask callers to indicate whether they have a touchtone telephone (usually by pressing one on the keypad). If no tone is detected, the application prompts callers to respond with spoken input (instead of transferring the call to an attendant). In this manner, callers who want to provide spoken input can be served by the system, instead of requiring an attendant.

If your application requires input that can easily be mapped to touchtone signals, do not ignore touchtone input in favor of speech recognition. For longer digit sequences, touchtone input is more accurate.

WholeWord and FlexWord speech recognition can be used in the same application for increased flexibility. When developing your application, you can specify a Prompt & Collect action to use either WholeWord or FlexWord speech recognition, depending on what you want the caller to say. For example, you can first prompt “Please say your account number.” After the selection, the next prompt is “Choose from the following transactions. Say ‘account balance,’ or say ‘transfer,’ or say ‘attendant’ to speak to a service representative.”

See FlexWord Speech Recognition below for more information on FlexWord speech recognition.

WholeWord Speech Recognition Capabilities

This section describes the different WholeWord speech recognition capabilities.

WholeWord Speech Recognition Languages

The WholeWord Speech Recognition software is used to recognize words in the following languages:

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

Each language can recognize:

- Equivalents for “yes” and “no”
- Single digits (zero through nine) and commonly used synonyms
- A series of digits (also known as connected digits)

Bilingual Applications

Any two of the languages listed above can be used together on a single system to support bilingual applications.

A bilingual person can recognize two languages simultaneously. A bilingual application, however, can only recognize speech in a single language at any one Prompt & Collect action. You can design an application that asks callers to indicate their preferred language, then play prompts and announcements, and recognize the words “yes” and “no” as well as spoken digits in one of the two languages installed in your system. This allows your system to understand callers who respond in either language you make available. See Bilingual, Multilingual, and Non-US English Applications in Chapter 4, “Designing a Voice Response Application,” for more information.

WholeWord Speech Recognition and Key Word Spotting

WholeWord speech recognition also supports key word spotting. A key word is one of the words in a list of words that the system is instructed to recognize at a particular point in the transaction. Key word spotting is the ability of the recognizer to isolate a key word from a stream of caller input, including extraneous speech, background noise, or caller noises. For this reason, callers do not have to say the key word in isolation. For example, if the recognizer is listening for callers to say "yes" or "no," it can also recognize "yes" if callers say "Yes, I do." However, the recognizer finds a key word most accurately when it is said alone, without any other words or other noises before or after it.

WholeWord Speech Recognition and Barge-in

WholeWord speech recognition supports barge-in. Experienced callers do not have to wait until the end of a prompt to begin speaking their responses. As soon as the system recognizes something the caller says, the prompt stops playing. This allows a single application to support both inexperienced and experienced callers.

WholeWord Speech Recognition Accuracy

Decide carefully where to allow spoken input. Spoken input is not the most appropriate input for all applications. touchtone input may be faster and more accurate if your callers are often speaking from a noisy environment such as a car or an airport. Some callers may have security concerns about speaking private information (like an account number) aloud if they use your service from outside their homes. If callers must enter a long series of digits or several data items, touchtone input may achieve better accuracy than WholeWord speech recognition.

FlexWord Speech Recognition

Another way for the system to process spoken input is with the FlexWord Speech Recognition software package. FlexWord speech recognition is used to recognize spoken words from a specific set of words, a *vocabulary*, that you, the application designer, define. Allowing callers to say the option they want instead of saying a number assigned to the option can make the interaction more natural and easier to use.

FlexWord Speech Recognition Uses

FlexWord speech recognition provides an intuitive, natural interface to the callers and may be most successful in applications where it would be awkward or inconvenient for callers to enter touchtone input, such as when entering a name. FlexWord speech recognition applications can support more menu choices than either touchtone input or WholeWord speech recognition. Menus can range from small, such as a choice of clothing sizes, up through large, such as the names of all 500 people in your company.

FlexWord speech recognition and WholeWord speech recognition can be used together for greater flexibility in your applications. You can define a Prompt & Collect action to use WholeWord speech recognition If you want callers to say “yes” or “no,” a series of digits, or single digits. If you want callers to say a word or phrase from your vocabulary, you would define the Prompt & Collect to use FlexWord speech recognition and specify the wordlist.

FlexWord Speech Recognition Capabilities

This section describes the FlexWord speech recognition unique capabilities.

FlexWord Speech Recognition Languages

The FlexWord Speech Recognition software is used to recognize spoken input for the following languages:

- French
- German
- Japanese
- Spanish
- US English

Only one FlexWord speech recognition language can be installed on your system at a time.

FlexWord Speech Recognition Vocabularies

As mentioned before, FlexWord speech recognition uses a custom, tailored vocabulary that is specific to each application.

FlexWord speech recognition vocabulary items can be single words or phrases. Each item in a vocabulary is known as a *word*, even if the item is a phrase of several words. For example, in a FlexWord speech recognition vocabulary, “checking” and “mutual fund” are both considered *words*, even though “mutual fund” is a phrase.

The words in a vocabulary must be divided into groups called *wordlists.* A wordlist includes all the words that can be spoken at a particular prompt. See Table 1 on page 15 for an example of how words are grouped into wordlists to form a complete vocabulary. This sample consists of 53 words that are divided into eight wordlists.

Each wordlist can contain up to 500 words, and you can have up to 200 wordlists. However, the total number of words cannot exceed 2000. It is possible for applications to share wordlists.

Table 1. Sample Vocabulary (Consisting of 53 Words Divided into Eight Wordlists)

Wordlist 1 (11 words)	**Wordlist 2 (6 words)**	**Wordlist 3 (6 words)**	**Wordlist 4 (8 words)**
skirt blouse dress pants shorts sandals shoes socks belt help attendant	small medium large extra large help attendant	petite small medium large help attendant	sky blue cranberry brown natural teal green rust help attendant
Wordlist 5 (6 words)	**Wordlist 6 (5 words)**	**Wordlist 7 (5 words)**	**Wordlist 8 (6 words)**
black brown light tan olive help attendant	women’s men’s child’s help attendant	narrow medium wide help attendant	visa master card american express discover help attendant

There are three ways to create your vocabularies:

- Use Voice@Work. See *Using Voice@Work*, 585-313-207, for additional information.
- Use Avaya custom vocabulary service.

  The custom vocabulary service is also an available option to build your vocabularies, for a fee.
- Use the services of an independent software vendor (ISV).

Consider the size of your vocabularies and how many FlexWord applications you plan to have when deciding to purchase the FlexWord Toolkit or use a custom vocabulary service.

FlexWord Speech Recognition and Key Word Spotting

FlexWord speech recognition supports key word spotting. As explained above in "WholeWord Speech Recognition and Key Word Spotting," key word spotting is the ability of the recognizer to isolate a key word out of a sequence of other words or noises. As with WholeWord key word spotting, the caller does not have to say the key word by itself to be recognized. For example, if the caller says "I want my checking account," the recognizer can isolate the key word "checking." Remember, though, that the recognizer finds a key word most accurately when the word is said all by itself.

FlexWord Speech Recognition and Barge-in

Unlike WholeWord speech recognition, FlexWord speech recognition does not support barge-in. Callers must wait until the end of a prompt to begin speaking their response.

FlexWord Speech Recognition Accuracy

Like WholeWord speech recognition, FlexWord speech recognition is most successful when applications first offer callers a chance to indicate that they have a touchtone telephone (usually by pressing one on the keypad). If no tone is detected, the application prompts the caller to respond with spoken input (instead of transferring the call to an attendant). In this manner, callers who want to provide spoken input can be served by the system, instead of requiring an attendant. You should continue to provide some attendants for your callers to maintain customer satisfaction.

As with WholeWord speech recognition, be careful about always allowing spoken input. If your callers may often be speaking from a noisy environment like an automobile or an airport, touchtone input will be faster and more accurate.

Natural Language Speech Recognition

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 ASR, 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.

See *Natural Language Speech Recognition with the Intuity CONVERSANT System*, 585-310-774, for additional information.

Speech Recognition Accuracy Measurement

The concept of recognition accuracy is similar for both WholeWord and FlexWord speech recognition. Since WholeWord vocabulary is fixed at only a few items in each language, we can measure and then expect accuracy rates. However, since FlexWord vocabularies are completely customized for each application, there is no absolute accuracy rate. For FlexWord speech recognition, accuracy can only be measured on each application, because each application will have a unique set of words and phrases that are grouped into wordlists. The recognition accuracy of each separate wordlist can be measured with some effort.

Recognition accuracy assumes that callers are saying something the recognizer can recognize. When a caller gives an invalid input, the recognizer should reject it. Accuracy is measured over many speakers, different types of applications, various environmental conditions, and regional accents. The average WholeWord accuracy rate for a speaker saying a single digit is 97%.

A 97% accuracy rate means that for a large number of different types of callers over a long time and under different conditions, 97% of *valid* inputs will be correctly recognized. It does not mean that every caller speaking the digit will be correctly recognized 97% of the time. Nor does it mean that 97% of people who call in will be recognized correctly.

Accuracy decreases slightly as digit sequences get longer, because of the difficulty to detect boundaries between digits. When two or more digits are spoken in a row (know as *connected digits*), the accuracy rate is 0.97n, where *n* is the number of digits that the recognizer expects callers to say. For example, a three-digit sequence is expected to have an accuracy rate of 0.97 x 3, or 91%. In general, you can subtract about three percentage points for each additional digit spoken in a row.

Improving Speech Recognition Accuracy

The guidelines in Chapter 4, “Designing a Voice Response Application,” will help you optimize the caller-application interface design, make choices on whether to use custom grammars, create affective FlexWord wordlists, and inform callers about what to say.

Text-to-Speech

This advanced speech technology allows your application to convert ASCII text into spoken words that are said to the caller; simply put, it is a *reading machine*.

Text-to-Speech Uses

Text-to-Speech is especially useful when it is not practical to prerecord information to be spoken, for example, if the information is lengthy, changes frequently, and/or comes from a database.

Information you could consider speaking to your callers with Text-to-Speech includes:

- News wire stories
- Rules and regulations
- Names and telephone numbers
- Telephone directory entries

In most of these cases, the cost of having a professional speaker prerecord all the information is prohibitive. Text-to-Speech is an economical way to provide callers with access to extensive or quick-changing information.

Text-to-Speech can also be used in place of prerecorded speech when prompting callers, thus saving you time when you demonstrate your application for testing purposes.

Text-to-Speech Capabilities

This section describes the Text-to-Speech unique capabilities.

Text-To-Speech Languages

Text-to-Speech is supported for United States English, United Kingdom English, and Austrailian English. Text-to-Speech speaks words in a male or female voice with a US English accent. Most callers find the speech easy to understand.

Text-to-Speech and Abbreviations

The Text-to-Speech feature expands abbreviations. If the text you want spoken contains abbreviations such as “Ave.” and “Mr.,” Text-to-Speech can speak them out as “Avenue” and “Mister.” Further, Text-to-Speech also allows you to specify the type of information that is being spoken. If you prepare Text-to-Speech to read an address, it will read “Dr.” as “Drive,” but if you prepare it to read a name, it will read “Dr.” as “Doctor.”

Text-to-Speech and touchtone Recognition

Text-to-Speech is fully compatible with touchtone recognition. An application design can allow callers to press a touchtone button to dial through a Text-to-Speech prompt or announcement the same as if it were a prerecorded prompt or announcement. Similarly, you can also allow callers to dial ahead, and respond to several prompts in a row.

Text-to-Speech and Speech Recognition

Text-to-Speech is also compatible with WholeWord and/or FlexWord speech recognition. An application can be designed so that callers can respond during a Text-to-Speech prompt or announcement the same as if it were a prerecorded prompt or announcement. However, when using Text-to-Speech prompts with FlexWord speech recognition, barge-in is not available and the caller must wait until the entire prompt is complete to speak.

Text-to-Speech Accuracy

Text-to-Speak assumes that the information it is reading is structured in standard US English sentences (complete with punctuation, capitalization, subject, object, and verb). If the information you want spoken is not written in complete sentences (for example, a series of database fields like name, address, and telephone numbers), the output will not be as understandable as full sentences. See Text-to-Speech in Applications in Chapter 4, “Designing a Voice Response Application,” for more information.

Proxy Text-to-Speech

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, 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).

PTTS Languages

The PTTS feature supports multiple languages, and even the use of multiple languages within a single Interactive Voice Response (IVR) application. Each language typically includes, at the least, one voice tag for each gender. For example, US English supports the following voice tags:

- "John-T", a male voice
- "Grace-T", a female voice.

PTTS currently 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

For a complete list of the currently supported languages for this feature, see the *Readme.txt* file that came with your 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).

PTTS Capabilities

The PTTS feature enables voice applications running on a CONVERSANT client system to request spoken playback of text phrases using:

- External functions called from either the Voice@Work or the Script Builder application development tool
- Intuity Response Application Programming Interface (IRAPI) functions

Text to be converted to speech can be stored in either a buffer or a file. It can be accessed by means of custom data interface processes (DIPs) that communicate with external databases. Or it can be downloaded to the CONVERSANT system and stored as text data files (normally in the **/vs/data/tts_files** directory).

A complete list of the currently supported voice tags can be found in the *Readme.txt* file that came on your **Proxy Text-to-Speech** CD-ROM.

Characteristics of the PTTS speaking voice, such as gender, rate of speech, volume, pitch, and intonation can be customized. Barge-in (talkoff) can be enabled to allow a caller to interrupt speech playback.

For the Japanese language only, a customer-configured pronunciation dictionary residing on the PTTS servers can be used to customize the conversion of selected words and phrases. For SAPI-compliant languages, each language provider must provide the pronunciation dictionary for the language, if one is desired.

FAX Actions

You can provide fax service to your callers in your applications with the FAX Actions software package. This software provides you with the capability to have your application send fax messages, and receive fax messages from callers.

FAX Actions Uses

You can send graphic and textual information to callers' fax machines upon their request. Your applications can give callers the opportunity to request information such as:

- Blank forms (such as tax forms, applications, or entry blanks)
- Customer service information
- Financial or medical records
- Reservation confirmations
- Advertising brochures

FAX Actions Capabilities

The FAX Actions offer the following capabilities for fax applications:

- Customize your cover pages.
- Send either pre-stored faxes or faxes dynamically created directly by the application.
- Schedule fax messages so that they are delivered when telephone rates are low.
- Redial the caller's fax machine if it is busy.
- Create an application to execute a UnixWare shell command to create text files. (Applications can fax anything that can be put into a text file, including host screens and ORACLE database reports.
- Receive faxes from callers into the system.

The FAX Actions can be used in the same application as touchtone input, DPR, WholeWord speech recognition, FlexWord speech recognition, NLSR, and Text-to-Speech. This allows you many options for designing fax applications. For example, you could design an application that allows physicians access to hospital patients' medical records.

Once the physician dials into the system and gives a personal identification number (using either WholeWord speech recognition, Natural Language speech recognition, or touchtone input), he or she enters a patient identification number to either listen to medical records spoken using Text-to-Speech or to have the medical record faxed to his or her office.

3 Planning a Voice Response Application

Overview

This chapter describes some guidelines to keep in mind when planning the design of your voice response applications. These are general guidelines and are not specific to the advanced technologies of the CONVERSANT system. Specific guidelines are described in Chapter 4, “Designing a Voice Response Application.”

Purpose

The purpose of this chapter is to provide you with information to plan your voice response application designs.

Plan Your Application Design

This section offers guidelines to use as you plan the design of your voice response applications.

Use Human Factors/Usability Engineering Resources

Human factors and usability consultants are experts in the design and testing of interfaces between people and computers. Some consultants specialize in telephone-based user interfaces. If your company does not have people trained in usability or human factors, independent consultants are available to assist you in application design, testing, and deployment.

Contracting professional design and usability testing services can be cost effective as well as save you time and frustration. Customer satisfaction can also be increased, since you will be more likely to offer your customers a well-designed, extensively tested application.

Avaya offers human factors design consultation services on a Professional Services basis. For more information, contact your Avaya representative. For a directory of individuals and companies offering independent consulting in human factors and usability, call the Human Factors and Ergonomics Society at (310) 394-1811.

Use Prerecorded Speech

Prerecorded prompts and announcements spoken by a professional speaker (such as a voice-over artist or a disc jockey) can enhance the quality of your application. It is usually appropriate to use the same recorded voice throughout the application. An exception is when you have a good reason to try to distinguish between different parts of the application. For instance, you may want to have "help" speech recorded by a male voice and the remainder of your prompts and announcements in a female voice. This consistency may help your callers to know their location within an application.

This section describes two methods to include professional speech in your applications.

Purchase Enhanced Basic Speech

The CONVERSANT system professionally recorded speech, known as *enhanced basic speech* in the following languages (the recorded voice is female unless separate male and female are indicated):

- Australian English
- Brazilian Portuguese
- Canadian French
- Cantonese Chinese
- Castilian Spanish
- Czech
- Dutch
- French
- German
- Hindi
- Hungarian
- Indonesian
- Japanese
- Korean
- Latin-American Spanish
- Malay
- Mandarin Chinese
- Polish
- Slovak
- Thai
- UK English
- US English (male and female)

Avaya also offers professional custom phrase recording services. If you are interested in purchasing custom phrases, contact your Avaya representative.

Contract a Professional Speaker

If you choose to contract a professional speaker on your own, consider the following guidelines:

- Record the speaker's voice to ensure you like the way it sounds. You may want to record several different speakers to compare the quality of their recorded voices.
- Make sure the speaker can maintain a constant speaking rhythm and intonation throughout the recording session.
- Make sure the speaker can maintain a constant, acceptable level of volume and distance from the telephone or microphone while recording.
- Make sure the speaker's pronunciation is clear and that words are not over-enunciated.
- Make sure the recording environment is as quiet as possible. A room with carpeted walls and floor is usually sufficient.
- Prepare the phrases for the speaker in advance of the recording session.

Offer a User Guide

If you know your calling population, you can take advantage of this knowledge and provide a user guide. User guides can be very simple, providing just enough information so that callers will not be surprised by the automated transaction. More detailed user guides can include all the prompts and options, so callers can prepare answers ahead of time and become oriented to the application.

The nature of your application and calling population will determine whether or not a user guide is appropriate. User guides are most useful when you know and can easily reach the members of your calling population. Examples of known caller populations include:

- Subscribers to your service
- Customers to whom you send bills
- Students, teachers, and parents of the school you serve
- Employees in your company

If you distribute a user guide and expect that most callers will use it, you may be able to provide shorter and more concise prompts than if callers do not have a guide. Be aware, however, that shorter prompts may adversely affect callers who forget to use the guide.

Offer Caller Training

If you know that a particular group of callers will use the application extensively, consider offering training classes. Training is most appropriate when you have a limited calling population, and/or a potentially complex or unusual application. The most useful training classes allow callers to call in to the application and use it.

Provide an Attendant

There will always be callers who need extra assistance, especially those calling for the first time. Provide a person who can handle callers having trouble with the voice response applications. As a general rule, allow callers to try to enter information only two or three times, then, if they are not successful, transfer the caller to an attendant. Allow callers to speak with an attendant by using a key press (like zero), or by speaking a key word (like "attendant," "operator," or "help").

Plan for Disabled Callers

Applications should be written to conform with Section 255 of the Telecommunications Act of 1996, which mandates accessibility by callers with disabilities. Applications used by the federal government must meet the accessibility requirements of Section 508 of the Rehabilitation Act of 1998.

In part, this means that an application should be usable by callers:

- without vision
- without hearing
- without speech
- without time-dependent controls that change meaning after a time-out
- with limited cognitive skills

Often it is a good idea to dedicate specific telephone numbers (corresponding to CONVERSANT ports that run the appropriate applications) for disabled callers to use.

Note: For more information about Section 255 of the Telecommunications Act, see the FCC website http://www.fcc.gov/cib/dro/section255.html. For more information regarding Section 508 of the Rehabilitation Act, see the FCC website http://www.fcc.gov/cib/dro/ab508.html.

Diagram Your Application Design

When planning your application, create a diagram of how you want the application to appear to callers. A diagram is helpful to depict the structure of the application. The diagram also allows you to plan what to do when errors occur, an aspect of applications that is frequently overlooked. If the same person or team is not the application designer and developer, a thorough diagram is a very valuable communication tool. Even when one person or team is both designing and developing the application, the diagram will help to separate the tasks, so that each task can receive adequate attention.

Flowcharts and outlines are two ways to diagram the application. If you already have a way to communicate the structure of the application that works for you, use it. Whatever method you choose, plan your application carefully before implementing it.

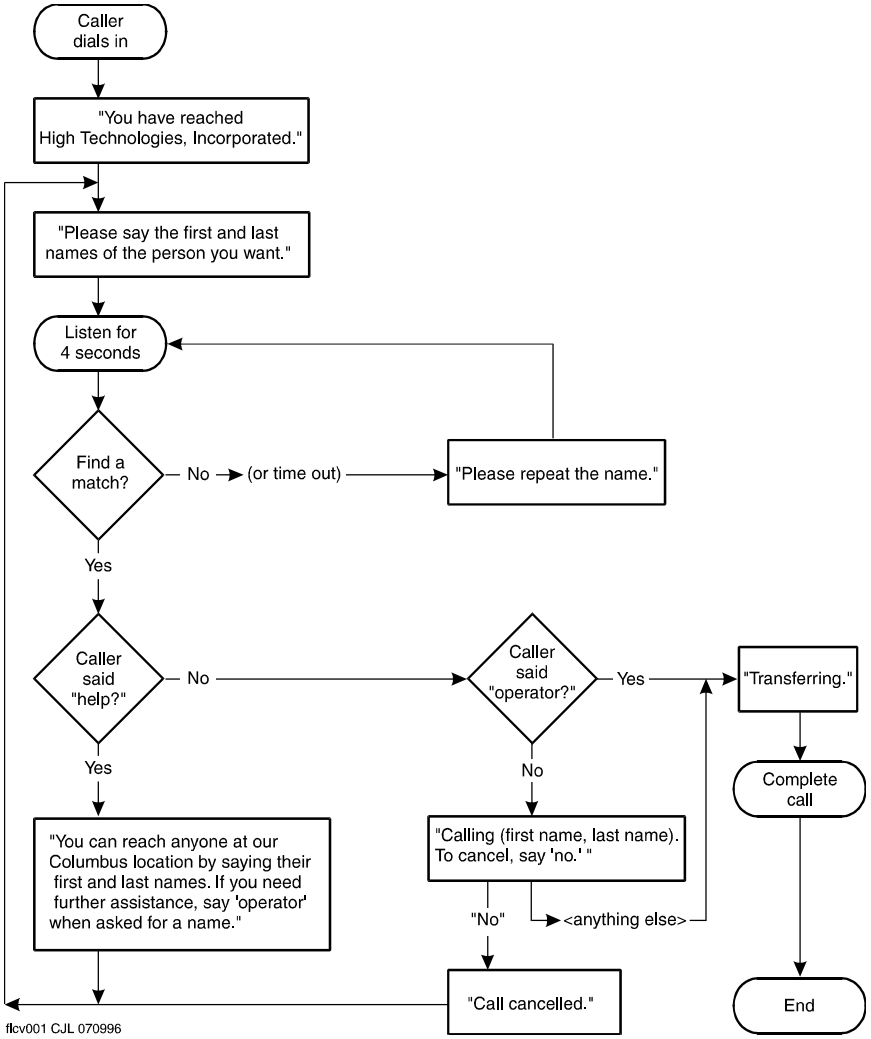
Be sure to list all prompts and announcements to be played to callers. Make sure you show what should happen each time you expect caller input. Include what should happen when callers give correct (valid) input.

Error cases are also important. What happens if callers give incorrect (invalid) input? How many times will you reprompt after hearing incorrect input? Will the call be transferred to an attendant? What if callers do not respond? How many times will you reprompt after hearing no response? A good flowchart or outline will cover all possible cases.

Use Flowcharts

Flowcharts are more formal and specialized, but they are a good way to show the application in a diagram. As shown in Figure 2, flowcharts use a few simple symbols to represent prompts, announcements, and decision points within the application, and arrows to show the path from one to another. Identify the symbols you use so that people can easily understand them.

Figure 2. Sample Application Flowchart



Use Outlines

Another way to diagram the application is an outline, as shown below. Outlines are easier for some people because they only use words, not symbols.

Sample outline:

Caller dials in.

Prompted with:"You have reached High Technologies, Incorporated."

1: "Please say the first and last names of the person you want."
2: (listen for four seconds; listen for anything on Wordlist 1, that contains all names).

If there is no caller response or silence:
Prompt with: "Please repeat the name."
Go to 2.

If the caller response matches the word "help:"
Prompt with: "You can reach anyone at our location by saying their first and last names. If you need further assistance, say 'operator' when asked for a name."
Go to 1.

If the caller response matches the word "operator:"
Prompt with: "Transferring."
Transfer the call to the operator's extension.

Finished.

If the caller response matches a name on the wordlist:
"Calling {first name, last name.} To cancel, say 'no'."
(listen for two seconds; listen for "yes" or "no").

If match the word "no:"
Prompt with "Call canceled."
Go to 1.

If match the word "yes" or nothing (no caller response or silence):
Prompt with: "Transferring."
Transfer the call to the named person's extension.

Finished.

4 Designing a Voice Response Application

Overview

This chapter describes specific guidelines to use when designing your voice response applications. Many of the guidelines in this chapter are specific and directly related to one or more of the CONVERSANT system advanced technologies described in Chapter 2, Voice Response Advanced Technologies.

Purpose

The purpose of this chapter is to provide specific guidelines to help you design robust and useful applications.

Application Design Research

In order to design the most successful application possible, it is a good idea to research some basic design principles. Human factors experts call these usability principles, since they are critical to designing a usable application.

Because of experiences with poorly designed applications, some callers are wary of all voice response systems. Be careful that *your* application does not alienate your customers. Your goal is to make an automated transaction at least as attractive and efficient as interacting with an attendant, if not more.

Know Your Callers

Choices you make in the application design will affect those who the application will serve. Find out as much as you can about the people who will call the system, and use the information when designing your applications.

Relevant information to gather includes the following:

- Who are your callers?

  Your callers have personal attributes that affect how they interact with the application you are designing.

  ~ What language do they speak? If you expect that many callers will have an accent that is different than what the speech recognition package you are using expects, consider offering a bilingual service. If you are offering service in only one language, but anticipate having many callers with accents, you may want to use DPR or a combination of DPR and speech recognition, since accented speech may not be recognized as accurately as nonaccented speech.

- ~ How old are they? If you expect that a high percentage of callers may be children, you should use simple language and very short menus. Also, since speech recognition works best with deeper voices, you may want to use touchtone input or dial-pulse recognition only. If you expect that many of your callers are over the age of 65, you might want to allow them more time to respond. Make sure to reduce confusion by making the transaction as simple as possible.
- ~ Are they employees of your company? You may want a script designed for in-house use to appear and perform differently than one designed for your customers. For example, if you provide training classes and/or user guides, you can probably use shorter and more concise prompts. Jargon and technical terms may be appropriate for employees, if you can assume that they all will know the terms.

Note any other personal attributes that could affect how callers will interact with your application, and work to make sure that your application serves all callers well.

- How often do they call?

  Some applications are meant to be used only one time, while others can be used more often. Callers might even access some applications several times a day.

  If you expect callers to use the application rarely, your prompts and announcements may need to contain explicit information and instructions. Prompts for applications used very often can be made extremely short, since most callers will likely be experienced. If you expect your callers to be mixed in their experience levels, make it possible for the more experienced callers to dial through or barge in during prompts.

- How well do they know the subject matter?

  If you expect that most of your callers will be unfamiliar with the subject matter (for example, if you are describing a new service or product), you should take special care in structuring and presenting the information so that your audience can understand it.

Use Simple and Natural Dialog

Learning as much as you can about callers will allow you to determine what is “simple and natural” for them. Design uncomplicated, straightforward applications that use terms familiar to the callers.

Minimize Demands on the Caller’s Memory

Psychological research has shown that a caller’s short-term memory can hold no more than five to nine separate pieces of information at one time. For each menu item you present to callers, two different pieces of information must be remembered: the option (“For loan rate information...”) and the action required to choose the option (“press 2.”) Menus should offer no more than four or five items at once, since callers cannot remember them all.

The fewer items callers have to remember, the more likely it is that they will remember them. Try to make the interaction easy for the callers. Keep menus short, and avoid wordy prompts and announcements.

Be Consistent

Consistency in the structure of menu choices and in the presentation of information helps prevent callers from becoming confused. By using consistent language and requesting consistent action of callers, you simplify the interaction and reduce demands on memory.

Provide Feedback

Whenever callers press touch tones or speak in response to a prompt, let them know how the response was interpreted. This clarifies the transaction for callers. For instance, if callers press zero for an attendant, the application could say “Please hold for an attendant.” Although the application does not explicitly state that it heard the caller press zero, the interpretation of the key press is implicit in the announcement that follows.

Provide Easy Exits

To prevent callers from getting trapped in an application script, tell them how to reach an attendant, end a transaction, or return to the main menu. You can give this information at the beginning of the transaction. Provide these types of escapes within your application so that callers can always return to a place they know or obtain help as necessary. This gives your callers more control over the interaction, and makes them feel as if they are not powerless in the face of automation.

Offer Shortcuts

If you expect some callers to call often enough to become experts, provide ways to shorten their interaction with the application. For example, allow them to respond before the end of a prompt so that they can move through the interaction at a pace that is comfortable for them. You may want to consider providing separate scripts (or separate branches of a single script) for novice and expert callers.

Prevent Errors

By following the above principles, and the specific recommendations in this chapter, you can prevent errors from happening. Well designed applications take into account the strengths and weaknesses of both the callers and the technologies.

Prompts

The prompts used in your application are one of the only forms of system interaction with callers. For this reason, It is very important that you design the application prompts to facilitate a successful interaction.

Use the guidelines in this section for prompt design.

Prompt Length

Make prompts as short as possible while remaining polite and informative. Keep the prompts brief, but not abrupt. The more quickly callers can do what they called to do, the more satisfied they will be. For example:

Instead of prompting...	prompt...
"To listen to information on travel to California, say 1."	"For California travel information, say 1."
"If you are calling to register for classes, press 3."	"To register for classes, press 3."

There are some trade-offs when considering prompt length. If you expect to have a high proportion of one-time or first-time callers, you might want to provide more information in prompts and make the prompts longer. Longer prompts are acceptable if you think your callers will need the extra information. Include only helpful information and express it in a short, direct manner.

Use longer, more informative prompts if you are asking callers to enter information in a way that may be unfamiliar. For instance, consider a caller entering a date. Most people think of months by name. For a touchtone application, callers are asked to press the one- or two-digit number for the month. For example, the month of December would be **12**. With WholeWord speech recognition, the month of December are represented by the words "one, two" which callers may say as "twelve." Both of these examples require that callers think of the information in a way that is unusual, and so prompts should be clear enough for callers to understand. These longer prompts should include examples, such as:

"Next, enter the month listed on your bill. Say two numbers, such as 'one, two' for December, or 'zero, five' for May."

How to Word Prompts

Use the following the guidelines when determining the wording of prompts:

- Present the option before the action.

  In menu prompts, always list the option first, and the action second. For example:

Instead of prompting...	prompt...
"Say 2 to receive the information by fax."	"To receive the information by fax, say 2."

  When the option is listed first, callers only need to listen to the one action that pertains to their choice. When the action is listed first, they must remember it until they hear the option and decide if it is the one they want. Listing the option first will help both experienced and new callers.

- Use the same structure for each option-action pair in a menu prompt. This helps the caller to know what to expect. For example:

Instead of prompting...	prompt...
"To hear sales information, press 1.	"For sales, press 1.
For services, press 2.	For services, press 2.
If you would like to speak to an attendant, press 0."	For an attendant, press 0."

- Make the menu choices clearly different.

  Be sure the menu choices are different from each other, so callers can decide which option they want. If callers are confused about which option they want, consider rewording the choices.

- Avoid jargon and acronyms.

  Assume that callers do not use the same technical words you do. Avoid acronyms and abbreviations that may confuse your callers. Speak out the complete words. Use words that your typical callers will understand. Use technical terms only if your audience is a small group of people who are trained in your business.

Announcements

As with prompts, it is very important that you design the application announcements to facilitate a successful interaction with your callers. Announcements are used to inform the caller. Be selective when deciding whether or not and how many announcements to include in your design.

Use the guidelines in this section for announcement design.

Feedback Announcements

Feedback announcements should be short and clear. If you want to include "please" and "thank you," include them only occasionally so that callers can complete the call as quickly as possible.

Avoid referring to the system as *I* or *we*. If callers think the system is a person, they may use more words to make their answers more friendly, and thus make speech recognition more difficult. For example:

Instead of prompting...	prompt...
"Please wait while I locate your records."	"Please wait while your records are located."

If a request will take more than two or three seconds to fill, assure callers that work is being done. For example:

"Looking for your records. Please wait."
or
"Please wait while your information is located."

Confirmation Announcements

There are times where you should read the entry back and allow the caller to confirm it. Other times, a feedback announcement (such as "Please wait while your information is located") is more appropriate than a confirmation announcement.

Use the information in this section to determine when you need confirmation or not.

When You Need Confirmation

In cases where a recognition error or caller touchtone error would have a significant impact (like credit card numbers or call destinations), you should give callers an opportunity to confirm the entry. To do this, repeat the entry and prompt callers to say “yes” or “no” to confirm it as in the following examples:

- With WholeWord speech recognition:

  “What is your customer number?”
  <**432886**>
  “Customer Number 432886. Is this correct?”
  {pause}
  “Please say yes or no, now.”

- With FlexWord speech recognition:

  “What department would you like?”
  <**Hardware**>
  “Calling hardware. Yes or no?”

- With touch tones:

  “Please enter your customer number.”
  <**432886**>
  “432886. If this is correct, press 1. If not, press 2."

- With dial pulses

  “Please enter your customer number.”
  <**432886**>
  “432886. If this is correct, enter 1. If not, enter 2."

With Script Builder and Voice@Work, you can include the Confirm action step in two different ways:

- Use the system Confirm when defining the original Prompt & Collect action.
  - ~ When collecting caller data with WholeWord speech recognition and touchtone recognition, use the Confirm within the Prompt & Collect.
  - ~ When collecting caller data with FlexWord speech recognition, use the Confirm within the Prompt & Collect action *only* if you have the US English version of WholeWord speech recognition and Text-to-Speech installed in the same system *and* Text-to-Speech is being used to play play speech to the caller
- Use a separate Prompt & Collect action to do the confirmation.
  - ~ When collecting caller data with a WholeWord speech recognition language other than US English, you must confirm with a separate Prompt & Collect action. A prerecorded speech phrase for each digit, “yes,” and “no” is also needed.
  - ~ When collecting caller data with FlexWord speech recognition, WholeWord speech recognition and Text-to-Speech software installed in the same system, confirm with a separate Prompt & Collect action.

See Table 2 on page 38 for a summary of confirming an entry within Script Builder and Voice@Work.

Table 2.Confirming an Entry within Script Builder and Voice@Work

Data collecting using...	should be confirmed with...
Touchtone input	The Confirm within Prompt & Collect action
WholeWord speech recognition US English	The Confirm within Prompt & Collect action
WholeWord speech recognition • Australian English • Brazilian Portuguese • Canadian French • Castilian Spanish • Dutch • French • German • Japanese • Latin-American Spanish • UK English	A separate Confirm action
FlexWord speech recognition (with US English WholeWord speech recognition and US English Text-to-Speech)	The Confirm within Prompt & Collect action
FlexWord speech recognition (with US English WholeWord only)	A separate Confirm action and recorded phrases for all words
FlexWord speech recognition (with Text-to-Speech only)[1]	A separate Confirm action and a FlexWord yes/no wordlist
FlexWord speech recognition (with no other packages)[1]	A separate Confirm action, recorded phrases for all words, and a FlexWord yes/no wordlist

1.In cases where you do not have the US English WholeWord speech recognition software, you could have the caller confirm with touchtone input.

When You Do Not Need Confirmation

In cases where a recognition error or a caller touchtone error would not cause a big problem (that is, if it is easy for callers to return to the point in the application where they really wanted to go), you may choose not to allow callers to confirm an entry. Also skipping the confirmation of nonessential entries can speed the call.

Even if you do not confirm an entry, it is still important to let callers know how the recognizer interpreted the input. To do so, include context-relevant information in the next announcement you play. For example:

"What month?"
<**August**>
"Artists performing at the Palace Theater for August include..."

Menus

Use the guidelines in this section when designing your applications with menus.

Number of Menu Choices

As mentioned in "Minimize Demands on the Caller's Memory," too many menu choices presented at once may confuse callers. In general, do not offer more than four or five choices in a single menu.

If you have more than four or five choices, separate them into small related groups and present more than one menu. For example, if you are giving information on seven parks, do not list them all in one menu. Your first menu could separate them into groups, and say:

"For parks with camping facilities, press 1.
For other parks, press 2."

If the caller presses 1:
"For Stony Ridge, press 1.
For Cantwell Cliffs, press 2.
For Old Man's Cave, press 3."

If the caller presses 2:
"For Seneca Rocks, press 1.
For Rialto Beach, press 2.
For Lugano Falls, press 3.
For Monroe Bluff, press 4."

In certain cases, it might be acceptable to present more than five menu choices. If all callers are trained, having longer menus may save time as long as you allow them to dial through or barge-in to interrupt the menu prompt. See Dial Through and Barge-in below for more information.

Menu Choice Sequence

To decrease the call length, present the most likely menu choice first, the second most likely second, and so on. If you do not know the preferred order, make a guess and adjust if necessary during your testing. See Chapter 5, "Testing and Using a Voice Response Application Design," for more information.

Numbered Menu Options

When presenting menu options that a caller can choose by number, present them in numerical order. Avoid skipping numbers.

If you change the sequence of your menu to present the more likely choices sooner, be sure to change the numbers of the menu prompts so that the caller can choose the first option by selecting 1, the second by selecting 2, and so on.

Keep in mind that experienced callers, and those with user guides, probably do not listen to all menu prompts. When you change one or more menus in your application, consider adding an announcement during the greeting, such as "The menu has changed; please listen carefully." After some time, you can remove this message.

Subdivided Menu Options

If your menu tree is complex, or you think that callers will want to get information from many different menus, give callers a choice to move to a different menu. You could add a choice to allow callers to go back to the previous menu, or a choice that would allow callers to go to the top of the menu tree. For example:

"For parks with camping facilities, press 1.
For other parks, press 2."
<**1**>

"For Stony Ridge, press 1.
For Cantwell Cliffs, press 2.
For Old Man's Cave, press 3.
For the previous menu, press 4."
<**2**>

"Cantwell Cliffs offers 15 tent camping sites, each with a water spigot. Cooking fires are permitted. Toilet and shower facilities are available. Reservations are accepted from April first through November first."

"To make a reservation, press 1.
For directions, press 2.
For the previous menu, press 3.
To start at the beginning, press 4."

Although the menu should automatically repeat if the caller does not respond, you could include an option like "To repeat the menu, press 9."

Digit Input

Use the guidelines in this section as you design your applications with digit input.

Constant-Length Digit Sequences

A constant-length digit sequence is a series that always contains the same number of digits, for example a United States social security number has nine digits.

The system recognizes constant-length sequences more accurately than variable-length sequences. Whenever possible, specify the maximum length of the variable-digit sequence to be recognized, to increase recognition accuracy.

Variable-Length Digit Sequences

A variable-length digit sequence can contain different numbers of digits, for example a house number in a street address is a variable-length digit sequence.

If your application must accept a variable-length digit sequence, you can increase recognition accuracy by using a two-step entry process. First ask for the number of digits, then ask for the digits. The recognizer will know how many digits to expect from the caller. For example:

"How many digits are in the next code?"
<**3**>
"Please say the three-digit code now."

If the digit sequence can have more than nine digits, remind the caller to say the number in digit format ("one, two") rather than as a natural number ("twelve"). See Use Examples Within a Digit Entry Prompt below.

Different types of credit cards may have different length numbers. If you allow callers to use one of several types of credit cards, use a menu to prompt them for the type of card, then for the numbers. This lets you know how many numbers to expect from the caller, so the recognizer will look for a constant-length number. For example:

"What type of credit card will you be using today? American Express, Discover, MasterCard, Visa, or Universal Card?"
<**Visa**>
"What are the first four digits on your Visa card?"

Entering Digit Sequences

For entry of sequences of 10 or fewer digits, the system provides constant-length grammars. For entry of sequences greater than 10 digits, it is recommended that you use a custom grammar. Otherwise, consider grouping the input. That is, prompt callers to read the series of digits in segments (groups), then confirm each group before going to the next. This technique is especially useful if the digit sequence cannot be validated with a checksum, custom grammar, or with a database, as described below in "Validate a Digit Sequence Entry." If the number already has natural groups like in a telephone number, use those groups. For example:

"Please say the first four digits of your 12-digit number."
<**4384**>
"4384. Is that correct?" {pause} "Please say yes or no."
<**yes**>
"What are the next four digits?"
<**9556**>
"9556. Is that correct?"
<**yes**>
"And the next four digits?"
<**8833**>
"883. Is that correct?"
<**no**>
"Please repeat the four digits."
<**8833**>
"8833. Is that correct?"
<**yes**>

This call is shorter if callers are allowed to barge-in during the prompt to answer the yes/no questions quickly.

Connected-Digit Versus Tone-Paced Digit Entry

WholeWord speech recognition supports the entry of connected-digit sequences, in all available languages. That is, callers can say a series of digits in one long breath, without intentional pauses in between. Many callers find this the most natural way to enter a series of digits.

You may have had experience with voice response systems that do not support connected-digit entry. In order to simulate connected digit input, other systems use pacer tones. When an application uses pacer tones, callers are prompted to speak the first digit, then wait for another tone before speaking the next digit. The tones then continue until all digits have been spoken.

If you prefer tone-paced method of digit entry, you can write your applications to use pacer tones. However, this type of data entry has not been proven to increase recognition accuracy rate for digit sequence. Pacer tones may even result in lower accuracy, since callers may become confused by the entry method. Connected-digit entry remains the recommended digit entry method for use with the CONVERSANT system.

Use Examples Within a Digit Entry Prompt

You may need to prompt callers to enter information that is not usually expressed in digit format. For example, dates are usually spoken “June eighteenth” or “six eighteen.” However, WholeWord speech recognition only recognizes single digits. June eighteenth must be spoken “six, one, eight” or “zero, six, one, eight.” This format is awkward for many callers. In order to make this easier, include an example in the prompt. For example:

“Next, enter the month and day of the month you were born. Use two digits for the month, and two digits for the day. For example, for June eighteenth, you would say ‘oh six, one eight.’ Please enter the month and day now.”

Use an example whenever you think that inexperienced callers will be less confused if you give them some help.

Validate a Digit Sequence Entry

When you know that a certain group of digit sequences are valid, use whatever techniques you can to validate callers’ inputs. This is especially important with spoken input and dial pulses, but can also be done for touchtone input. These validation methods help you increase the recognition accuracy for connected-digit sequences. The following sections describe three ways to validate an entry.

Custom Grammars

You can limit the results of the recognizer to valid sequences by using a custom grammar obtained from Avaya. A grammar is the set of rules the recognizer uses to understand data.

A number of grammars are already provided with Script Builder (a list can be seen in the Choices menu for the `Recognition Type` field on the Prompt & Collect screen) and Voice@Work (a list can be seen in the Input ASR Mode field on the Prompt & Collect Response tab). Custom grammars are special additions to this standard set, designed for your specific needs. For example, if you know that all valid callers will have telephone numbers from only a few telephone area codes, request a custom grammar that only allows the area codes you want. This helps the recognizer make accurate judgments about caller speech.

Checksums

Another way to validate an entry is to use a checksum. Some numbers (like credit card numbers) have a checksum built in to the structure of the number, so that the number can be checked with formula to determine if it is valid.

If the digit series you are accepting has a built-in checksum, use a data interface process (DIP), Script Builder **Evaluate** statement, Voice@Work Set and Test node, or external function to evaluate the checksum. Once the system determines that the entry is valid, it can speak the entry back for confirmation, if necessary. See Confirmation Announcements for more information.

Database Lookups

You can also validate an entry by looking for it in a database of valid numbers. For example, when accepting a telephone number, locate the number in the database of all customer telephone numbers. If the number is found, speak it back to the caller for confirmation, if necessary. If it is not found, the system can search the database for a telephone number that matches the second-best recognition result. If the second choice is found, the system assumes that this is what the caller entered, and can speak it back for confirmation.

Confirming Digit Entries with Callers

As with many of the examples in this section, it is important to confirm digit entries with callers.

Confirming Single Digit Entries

In order for the system to speak back digits for confirmation, you must ensure that there is a recorded phrase associated with each digit. Script Builder and Voice@Work will supply the phrase tags for these digits, and allow you to record them. You can also use enhanced basic speech in the language of your choice to speak back digits.

Confirming Digit Sequences

With non-US English languages, the only reliable way to speak caller numeric entries back to them is as a series of digits, rather than a natural number. For example, a caller that speaks “one-hundred, twenty-three” will hear their input read back as “one two three.” This should be adequate for most applications.

As discussed above in “Confirmation Announcements,” the recognition Confirm with the Prompt & Collect actions works only when you are using US English WholeWord speech recognition. WholeWord speech recognition in other languages must use a separate Prompt & Collect action to confirm caller entries.

Yes/No Questions

Use the guidelines in this section as you design your applications with questions that require a "yes" or "no" response, also known as *yes/no questions*.

Touch-Tone Input for Yes/No Questions

In situations where you want callers to respond with touchtone input for "yes" or "no," ask them to press **1** for yes or **2** for no. For example:

"If this is correct, press '1'. If not, press '2'."

It is best to instruct callers to press **1** and **2**, although you may want to use **1** and **0**. Even if you prompt callers to press only **1** and **2**, you can set up your application to accept additional numbers. For example, **1** for yes and **2** and **0** for no. Whatever numbers you use to represent yes and no, use the same numbers throughout the application. Do not confuse callers by using **2** for no in one place and **0** for no in another.

Spoken Input for Yes/No Questions

In situations where you want callers to respond with spoken input, instruct them to say "yes" or "no," rather than a number. Callers find this natural, and make fewer mistakes than if you ask them to say a number meaning yes or no. For example:

Instead of prompting...	prompt...
"If this is correct, say 1. If it is not correct, say 2."	"Is this correct? Please say yes or no, now.

Write your yes/no questions similarly throughout the application, so that callers will know what to expect.

Recognizing "yes" and "no" is best done with WholeWord speech recognition. If you do not plan to purchase the WholeWord speech recognition software, you can use FlexWord speech recognition with a wordlist including "yes" and "no." Be aware that the accuracy for the FlexWord speech recognition version may be lower than the WholeWord speech recognition version.

Yes/No Questions with Barge-in (WholeWord Speech Recognition)

After asking a yes/no question, pause to give the caller time to respond, then present the possible answers. The prompt will stop playing as soon as the recognizer detects a spoken "yes" or "no" or a touchtone signal. For example:

"You said '64587'. Is this correct?"
{a 1.5 second pause}
"Please say 'yes' or 'no'."

Use the Prompt & Collect action to ask the question, play a series of silence phrases, then present the options. Figure 3 shows an example of how part of your Script Builder code will look if you ask the caller for five digits, then confirm the entry within the Prompt & Collect action.

Figure 3. Script Builder Code for Yes/No Questions with Barge-in

```
Prompt & Collect
   Prompt
      Speak with Interrupt
         Phrase: “Please enter your 5-digit customer
         number.”
   Input
      Mode: US_DIG
      Min Number of Digits: 05
      Max Number of Digits: 05
   Checklist
      Case: “Input Ok”
         Speak with Interrupt
            Phrase: “You said”
            Field: $CI_VALUE As C
            Phrase: “Is this correct?”
            Phrase: “sil.500”
            Phrase: “sil.500”
            Phrase: “sil.500”
            Phrase: “Please say yes or no.”
         Confirm
      Case: “Initial Timeout”
         Reprompt
      Case: “Too Few Digits”
         Reprompt
      Case: “No More Tries”
         Quit
End Prompt & Collect
```

Figure 4. Voice@Work Example for Yes/No Questions with Barge-In

 Node: **Get_Customer_Account** Type: **Prompt and Collect**

First node in the Call Flow.

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 [Customer_Number_Prompt]:
“Please enter your 5 digit customer number.”

Caller responds by speaking any valid word/touchtone sequence from the ASR Grammar list, **US English 5 digits (0-9 and 'oh').**

5 seconds to begin speaking.
[Tries] tries to correctly respond.
Number of tries used stored in **[Attempts]**.
Input saved in **[CollectedDigits].**

For a valid response, the caller hears the Confirm Prompt **[Confirm_Customer_Number]**:
"You said [*CollectedDigits*] Is this correct? sil.500 sil.500 sil.500 Please say 'yes' or 'no'."

The caller must confirm their response by speaking**[YesDigit].**
The caller can confirm by speaking any valid word/Touchtone sequence from the ASR Grammar list, **US English Yes or No**.

Caller ResponseGo To BranchExecute NodePage #
Any valid responsePass(none)
Failure on the last tryFail(none)

If the caller responds incorrectly and has more tries, then:

For an invalid response, the caller hears the **Bad Input Prompt [Bad_Input]:**
"I'm sorry that was not valid."

For no response, the caller hears the Timeout Prompt **[No_Response]**:
"You must enter a response."

The caller is given another chance to respond and hears the Initial Prompt.

Yes/No Question Without Barge-in

If barge-in is not allowed, you have two options:

- Ask the question, but do not list the options. For example:

  "You said '645987'. Is this correct?"

  This method is very successful for experienced callers. If callers are confused, try the second method.

- List the options, but do not leave an extended pause between the question and the options. For example:

  "Would you like to place another order? Please say 'yes' or 'no'."

Pace the Application

Use the guidelines in this section to design the pace and speed of your applications.

When Callers Must Wait

If you know callers will have to wait, it is a good idea to let them know. For example, the application could say:

"Please wait while your request is processed."

If you know that the caller may have to wait for longer than a few seconds for the application to continue (because of a database call or host connection, for example), it is a good idea to fill the gap during the wait. You could record a long phrase, perhaps music, and play it to the caller while the information is being located. Or, play a different phrase every seven to ten seconds to let callers know that work is being done. To do this, you must use an external function. For an example of the TFLUSH external function in *CONVERSANT System Version 8.0 Application Development with Advanced Methods*, 585-313-216. Pay special attention to the description of the wait indicator.

Allow Time for Caller Responses

Pace the interaction so that callers have time to enter the required information and listen to the prompts and announcements. You can determine how comfortable callers are with the pace of the interaction by testing it as described in Chapter 5, Testing and Using a Voice Response Application Design.

Do not let the application respond to caller actions too quickly. After callers are asked to make an entry, play a speech phrase consisting of about a half second of silence before the next prompt or error message. Since many telephones have the keypad built into the receiver, this pause leaves time for callers to lift the receiver back to their ears after pressing the keys. If your calling population includes senior citizens, this pause can give them more time to react.

Adjustable Pacing

Write your application to allow pauses to be adjusted easily. You can include a pause as mentioned earlier, but you can also tune the length of the pause during your application testing. In the beginning of your application, define a field called **entry_pause**. Assign to **entry_pause** the speech phrase number associated with one of the silence phrases. Consider the following Script Builder example:

```
Set Field Value

Field: entry_pause = 1074
```

Then, throughout the application, speak the field **entry_pause** before the prompts that follow caller entries. For example:

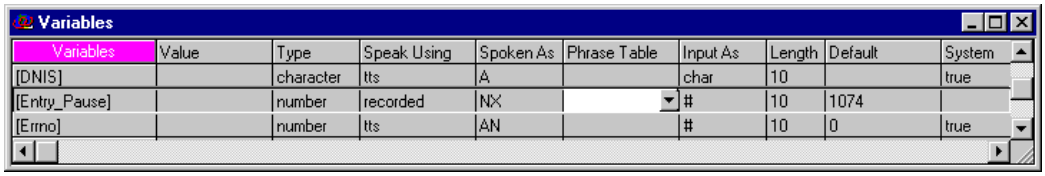
```
Announce

Speak without Interrupt
   Field: entry_pause As NX
   Phrase: "Next, please enter your 5-digit PIN."
```

If you find during testing that the application responds too quickly, all you need to do is go to where you set the field value and use a longer silence phrase. With this method, you can make many changes in the application by changing only one line. You can do the same thing for other pauses in your application, such as pauses between menu items.

Figure 5 shows an example for Voice@Work

Figure 5. Adjusting Pauses in Voice@Work



Application Errors

Use the guidelines in this section when designing your applications to process errors.

When Caller Errors Occur

Caller errors occur when callers enter information that the application considers incorrect, or when callers enter nothing at all. When callers make errors, provide them with an informative error message; tell them what went wrong and how to correct it. To assure that callers hear error messages, do not allow barge-in or dial through during at least the first part of error messages.

Allow callers no more than two or three tries to enter information. After the first error (and second, if allowing three), the application should speak out a prompt that contains more information, so that callers can get a better idea of what is required. After the last try, tell callers to hold for an attendant, then transfer the call.

When Speech Recognition Errors Occur

Speech recognition errors occur when the recognizer cannot match the caller's speech to a phrase it is prepared to recognize.

Use the following guidelines for processing speech recognition errors:

- Choose one concise phrase and use it consistently within a single application. For example:

Instead of phrases such as...	Use phrases such as...
"We're sorry, but your speech cannot be understood. Please repeat the category you want."	"Sorry. Please repeat."

- When using a FlexWord wordlist of five or fewer items, reprompt with a "sorry" phrase, then give a list of the valid options. For example:

  "Sorry. Please say 'checking', 'savings', 'interest rates', 'loan rates', or 'operator', now."

- When using a large wordlist, include some helpful information in the reprompt. For example:

  "Department, please."
  <**Excuse me?**>
  "Sorry. Please say the name of the department you want, or hold to speak with an attendant."

How to Word Error Messages

To speak out an informative error message using the Script Builder Prompt & Collect action, use either the Try Again or Reprompt action. Within Voice@Work, use the Initial Prompt field on the Response tab within Prompt and Collect. Set this field to False for Try Again and True for Reprompt. Be sure to specify the informative error message in the associated Voice Response field. Specify an appropriate error message for each type of error that can occur (Initial Timeout, Too Few Digits, or Not on List), as shown in the following examples:

- With touchtone or dial pulse input:
  - ~ If the first prompt is:

    "What time would you like your wake-up call?"
    and the caller does nothing (Initial Timeout error)

    A more informative prompt could be:

    "Please enter the time you would like your wake-up call. Use the telephone keypad to enter two digits for the hour and two digits for the minutes."

  - ~ If the first prompt is:

    "Enter the date of the lottery."
    and the caller presses some touch tones, but not enough (Too Few Digits error)

    A more informative prompt could be:

    "Please enter the date of the lottery for which you want to hear winning numbers. Use the telephone keypad to enter two digits for the month, and two digits for the day. If a month or day has only one digit, use a zero before the digit."

- With FlexWord speech recognition:

  If the first prompt is:

  "What catalog will you be ordering from."
  and the caller makes an invalid entry or the recognizer does not recognize a spoken entry as valid (Not On List error)

  A more informative prompt could be:

  "Sorry. You can order from the Furniture or the Linen catalogs. Please speak the name of the catalog."

Avoid using negative or accusing words when indicating an error. Words such as "invalid," "bad," "wrong," or "incorrect" can make callers feel threatened. "Sorry" is a brief way to say that the input was not recognized or was incorrect, and "Please repeat" or "Please enter again" are brief ways to ask callers to act again.

Touch-Tone and Speech Recognition

Although you might already know how to design good touchtone applications, here are some special things you need to know that will help you approach speech recognition wisely.

Why are the differences between touchtone recognition and speech recognition so important? A successful application takes advantage of the best aspects of speech recognition, while minimizing the flaws. The following information will help you understand and appreciate the differences between touchtone recognition and speech recognition, and how the technologies affect application design.

Recognizer Differences Between Touch-Tone Input and Spoken Input

The way in which the CONVERSANT system processes touchtone input and spoken input is fundamentally different, so the application designer must take care to use each feature in a way that is tailored to how callers will interact with the application. Table 3 compares and contrasts touchtone interactions with speech recognition interactions. The following sections then further explain what is depicted in the table.

Table 3.Differences Between TouchTone Input and Spoken Input

Possible Input	What the System Can Recognize	How the System Analyzes the Input	Conclusion
One or more tones	Touchtone signals	Takes touchtone signals and maps to a number on the keypad	Excellent mapping between input and what the recognizer “knows”
Any noise that a person can make, or any background noise that can be heard over the telephone	A set of word models restricted by grammar	Takes a speech sample and tries to match it to the active word models	Imperfect mapping between input and what the recognizer “knows”

Application Differences Between TouchTone Input and Spoken Input

Suppose that you are working on an application that callers will use to report problems with telephone service. Part of your transaction involves asking callers for a telephone number. For a touchtone interaction, you could prompt "Enter your 10-digit telephone number." The caller can do one of three things: press 10 keys, press fewer than 10 keys, or do nothing.

- If the caller presses 10 keys, the application reads back the telephone number for confirmation: "Did you enter 614-860-4001?"
- If the caller presses fewer than 10 keys, the application detects this and reprompts the caller for a valid entry.
- If the caller does nothing, the application receives no touchtones. When this happens, the application can reprompt, providing more information.

For a similar speech recognition interaction, you could prompt, "Say your 10-digit telephone number." The caller can say anything at all in response (words or other sounds) or say nothing. The recognizer listens, then attempts to map the sounds to the appropriate grammar (US_DIG, with minimum and maximum digits = 10), and come up with a series of digits. No matter what the caller says, the recognizer does the same thing: it listens to the sounds coming through the telephone, and attempts to map 10 digits, or using phrase screening, rejects the utterance.

- If the caller says 10 digits, the recognizer tries to map the speech to 10 digits, then reads back the result for confirmation: "Did you say 614-860-4001?"
- If the caller says fewer than 10 digits, the recognizer attempts to map the speech to 10 digits, then reads back the result for confirmation: "Did you say 618-604-0001?"
- If the caller remains silent and there is background or line noise, the recognizer rejects the utterance. The recognizer returns a null string to the application, and the application can reprompt, giving the caller more information. This is considered a "correct rejection."
- If the caller says "I do not have a phone number yet," the recognizer rejects the utterance.

What can be done to improve the application design in this example? What could increase the chance that callers will say what the recognizer is prepared to accept? Your application could first ask if the caller has the required information, by prompting: "Do you have your telephone number available?" Callers who say no can be transferred to the attendant, while callers who say yes can be prompted for the number.

TouchTone Input Used with Spoken Input

You can use WholeWord speech recognition to allow callers without touchtone telephones to use the voice response system by speaking “yes” or “no.”

There are three different approaches for planning the application:

- Write two, separate applications that are nearly identical; one to process touchtone input and one to process spoken input
- Write a single application to process both touchtone and spoken input
- Modify a touchtone input application to process both types of input

Separate Touchtone and Speech Recognition Applications

This approach allows you to use prompts that are best for touchtone entry in one application and those that are best for speech recognition entry in the other.

In many cases, touchtone recognition may be more accurate and cost less than speech recognition. Therefore, you probably want callers with touchtone telephones to use their keypads to interact with the application. To encourage this, start your application with a *dial 1* prompt. A dial 1 prompt gives callers a chance to indicate that they have a touchtone telephone, and helps speed the call.

Figure 6 illustrates a dial 1 prompt in a flowchart. Figure 7 shows an example of the Script Builder code for the dial 1 prompt. Figure 8 shows an example of the Voice@Work code for dial 1 prompt.

Figure 6. Example of a Dial 1 Prompt Flowchart

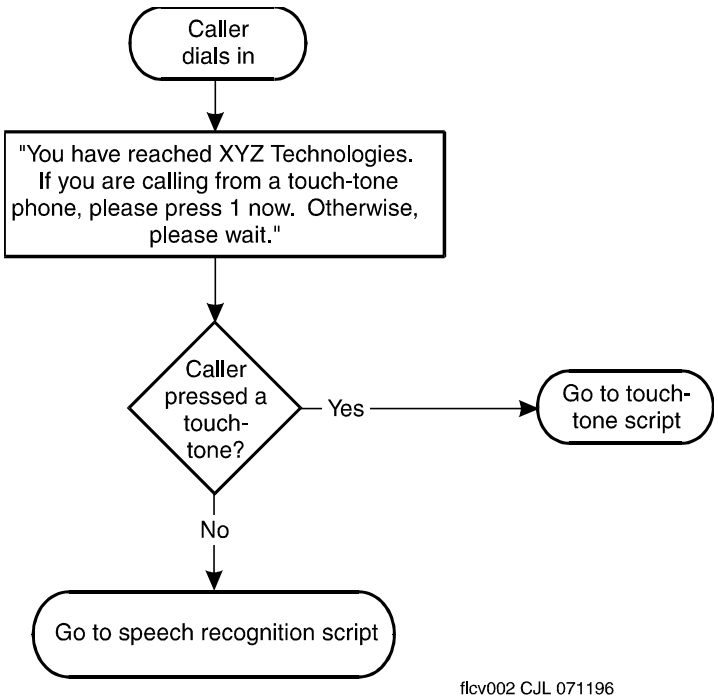


Figure 7. Script Builder Code for Dial 1 Prompt

```
start:
Answer Phone
Prompt & Collect
Prompt
    Speak with Interrupt
        Phrase: “You have reached XYZ Technologies.”
        Phrase: “If you are calling from a
        touchtone phone...”
Input
    Max Number of Digits: 01
Checklist
    Case: “Input OK”
        Continue
    Case: “Initial Timeout”
        Goto sr_script
    Case: “Too Few Digits”
        Goto sr_script
    Case: “No More Tries”
        Quit
End Prompt & Collect


touchtone_script:
#put touchtone application here


sr_script:
#put speech recognition application here
```

Figure 8. Voice@Work Example for Dial 1 Prompt

 Node: **Answer_Phone** Type: **Answer Call**

First node in the Call Flow.

Answer the incoming call.

AfterGo To BranchExecute NodePage #
Call AnsweredPassCheck_for_Touch_...5

 Node: **Check_for_Touch_Tone** Type: **Prompt and Collect**

Enter this node from node **Answer_Phone**.

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 **[Touch_Tone_Prompt]**:
"You have reached XYZ Technologies. If you calling from a touchtone phone..."

Caller responds with touchtones.

5 seconds for first touchtone, 3 seconds for subsequent touchtones.
Exactly 1 touchtone inputs must be entered.
[Tries] tries to correctly respond.
Number of tries used stored in **[Attempts]**.
Input saved in **[CollectedDigits]**.

There is no Confirm Prompt to play.

Caller ResponseGo To BranchExecute NodePage #
Any valid responsePass[Touch_tone_Script...6
Failure on the last tryFail[Speech_Recognitio...Error! Bookmark not defined.

If the caller responds incorrectly and has more tries, then:

The caller is given another chance to respond and hears the Initial Prompt.

A Single Touchtone and Speech Recognition Application

An application that uses touchtone recognition and speech recognition together requires that every prompt be acceptable for both entry methods.

⚠ CAUTION:
Use this method only if your system resources can process the extra load, since many callers with touchtone telephones may choose to use speech recognition instead. Remember that recognition accuracy is lower for spoken input than for touchtone input.

When using a single application:

1. Structure your prompts so they give callers the choice of using the keypad or speaking.
2. At the beginning of the application, tell callers that entries may be made with either the keypad or by speaking. For example, the application might say:

   "Welcome to XYZ Technologies. During this call, you can use your telephone touchtone keypad to make entries, or you can simply speak your responses."

3 Make sure that the prompts are appropriate for either kind of input. There are two ways to do this:

- ~ Tell callers to "enter" the information. For example:

  "Enter the zip code of the restaurant you visited."

- ~ Present the prompt as a question. For example:

  "What is the zip code of the restaurant you visited?"

Either method is appropriate. To accomplish this in Script Builder or Voice@Work, write the application the same as you would for WholeWord speech recognition input. For digits, the application automatically accepts touchtone input in place of the spoken input. For "yes" and "no," you must identify in the application what touchtone keys are associated with the words "yes" and "no," and specify the action associated for each. For Script Builder, use the third page of the Prompt & Collect screen for these specifications. For Voice@Work, insert the appropriate branches on the Prompt and Collect node.

You also must tell callers to press a touchtone key instead of saying "yes" and "no," in case they want to indicate their response with a touchtone key. The prompts to accomplish this could be:

Prompt: "Would you like to leave us a message regarding the service agreement?"
{pause}
"Please enter yes or no."

Reprompt: "Please say 'yes' or 'no'. Or, for yes, press 1. For no, press 2."

Figure 9 shows an example of the Script Builder code to accept touchtone input and spoken input. Note that it is important to use Try Again instead of Reprompt in this Prompt & Collect action. If you use Reprompt, the application will repeat the phrase "Would you like to leave a message" after the phrase "Please say 'yes' or 'no'. Or for yes, press 1. For no, press 2." The application flow would be awkward.

Figure 9. Script Builder Code to Accept Touchtone Input and Spoken Input

```
Prompt & Collect
Prompt
   Speak with Interrupt
      Phrase: “Would you like to leave a
      message...?”
      Phrase: “sil.500”
      Phrase: “Please enter yes or no.”
Input
   Mode: US_YN
   Max Number of Digits: 01
Checklist
   Case: “Y”
      Continue
   Case: “N”
      Continue
   Case: “1”
      Continue
   Case: “2”
      Continue
   Case: “Not on List”
      Speak with Interrupt
         Phrase: “Say yes or no.  Or for yes, 1.
         For no, 2.”
      Try Again
   Case: “Initial Timeout”
      Speak with Interrupt
         Phrase: “Say yes or no.  Or for yes, 1.
         For no, 2.”
      Try Again
   Case: “Too Few Digits”
      Speak with Interrupt
         Phrase: “Say yes or no.  Or for yes, 1.
         For no, 2.”
      Try Again
   Case: “No More Tries”
      Quit
End Prompt & Collect
```

Figure 10.Voice@Work Code to Accept Touchtone Input and Spoken Input

 Node: **Prompt_for_Message** Type: **Prompt and Collect**

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 **[Message_Prompt]:**
“Would you like to leave a message...? sil.500 Please enter ‘yes’ or ’no’.”

Caller responds by speaking any valid word/touchtone sequence from the ASR Grammar list, **US English Yes or No**.

5 seconds to begin speaking.
[Tries] tries to correctly respond.
Number of tries used stored in **[Attempts]**.
Input saved in **[CollectedDigits]**.

There is no Confirm Prompt to play.

Caller ResponseGo To BranchExecute NodePage #
Any valid responsePass(none)
Failure on the last tryFail(none)

If the caller responds incorrectly and has more tries, then:

For an invalid response, the caller hears the Bad Input Prompt **[Bad_ Input]**:
"Say 'yes' or 'no'. Or for yes, '1'. For no, '2'."

For no response, the caller hears the Timeout Prompt **[Bad_Input]**:
"Say 'yes' or 'no'. Or for yes, '1'. For no, '2'."

When there are not enough touchtones in the response, the caller hears the Too Few Digits Prompt **[Bad_Input]**:
"Say 'yes' or 'no'. Or for yes, '1'. For no, '2'."

The caller is given another chance to respond.

Use Touch-Tone Input When Speech Recognition Fails

If you are using WholeWord speech recognition, you may increase the number of successful interactions by encouraging callers to use touchtone input when spoken entries have been recognized incorrectly.

Touchtone input may be more accurate than spoken input. You can reprompt callers to indicate their response with touchtone input if they have a touchtone telephone. The implementation is not difficult, since an application written to accept spoken input will also accept touchtone input.

A typical prompt set might look like this:

"Please enter the 6-digit code for the fax information you want."
<**"135683"**>
"You entered 135688. Is this correct?"
<**"No"**>
"Please reenter the code. If you have a touchtone telephone, you may want to enter the code using your keypad, instead."

Modify a Touch-Tone-Only Application to Include Spoken Input

When you modify a touchtone input application to support spoken input (in addition to or in place of touchtone input), remember the following guidelines:

- Be aware of differences in Script Builder and Voice@Work. For instance, the Initial Timeout in the Prompt & Collect action is different for touchtone than for speech recognition. For touchtone, initial timeout represents the number of seconds the application will wait for the first touchtone response from the caller. For speech recognition, initial timeout represents the number of seconds from the end of the prompt until the application demands a result from the recognizer (regardless of whether barge-in is on or off).
- Before you add speech recognition to a touchtone application, examine the application to see what the prompts say and what action is expected from callers. Think about the capabilities of the speech recognition software you will be using, and where they might fit in to the application.
- If you plan to use FlexWord speech recognition, pay attention to places where choices have numbers. For example, a touchtone prompt in a homework assignment application that says,

  "To listen to homework, press 1. To leave a message for the teacher, press 2. To get a different class, press 3."

  can be modified to take advantage of custom wordlists. You could change the prompt to say,

  "Would you like to listen to homework, leave a message for the teacher, or choose a different class?"

  Table 4 shows the words your wordlist might contain for a homework assignment application:

Table 4.Wordlist for a Homework Assignment Application

Word	Maps to Option
listen to homework	Listen to homework option
homework	Listen to homework option
listen	Listen to homework option
leave a message for the teacher	Leave a message for the teacher option
leave a message	Leave a message for the teacher option
teacher	Leave a message for the teacher option
choose a different class	Choose a different class option
different class	Choose a different class option
different	Choose a different class option
class	Choose a different class option

See, “FlexWord Speech Recognition in Applications,” below for more information.

- If you plan to use WholeWord speech recognition, your application will be able to recognize the words “yes” and “no” and connected digits. In places where the touchtone application prompted for digit entry (especially long sequences), you may want to introduce custom grammars, database checking, checksums, or grouping to get the highest recognition accuracy possible. For more information on digit entry, see “Digit Input,” above.

Dial Through and Barge-in

Use the guidelines in this section to design your applications that properly encourage or discourage dial through and barge-in.

Using Dial Through and Barge-in with Errors Messages

For error messages, require callers to listen to at least part of each error message, so the interaction stops and the mistake can be corrected. For touchtone and WholeWord speech recognition input, turn off dial through/barge-in during the first part of each error message. For example:

Turn dial through or barge-in off and prompt:
“Sorry. Please enter...”
turn dial through or barge-in on
“...your customer number.”

Using Dial Through and Barge-in Consistently

Callers expect and appreciate consistency. Therefore, it is a good idea to apply dial through and barge-in consistently in your applications. Allowing callers to barge-in during some prompts and not others in the same application could cause confusion.

For consistency, applications that use FlexWord speech recognition or dial pulse recognition, even at one prompt, should not allow dial through or barge-in. You may find, however, your callers may want to barge-in during some prompts, and may be able to accept some inconsistency. During your design testing, determine what will work best for your calling population, and design your applications accordingly. See Chapter 5, “Testing and Using a Voice Response Application Design,” for information about testing your application design.

How to Word Prompts for Dial Through and Barge-in

The wording, length, and structure of prompts can discourage callers from responding during a prompt. You want to ensure that the caller's responses are consistent with how you have implemented dial through/barge-in: enabled or disabled.

How to Encourage Dial Through and Barge-in

Experienced callers like to shorten the call by responding during the prompt, and thus save time because they know what to say or do. Therefore, if you have the dial through/barge-in capability enabled for a prompt, you should encourage callers to respond when they are ready, even during the prompt. To encourage dial through or barge-in for a menu, leave a pause after each option. Using the Prompt & Collect action, play an option, then play a series of silence phrases equal to 1.5 seconds. Silence is not needed after the final option. For example:

"For sales, say '1'. {1.5 second pause}
For service, say '2'. {1.5 second pause}
For an attendant, say 'zero'."

For information on how to encourage barge-in for a yes/no prompt, see "Yes/No Questions with Barge-in (WholeWord Speech Recognition)," above.

Barge-in uses system resources also used by speech recognition. If resources are a problem, you could disable barge-in until you have expanded your speech recognition resources.

How to Discourage Dial Through and Barge-in

If you have the dial through/barge-in capability disabled for a prompt or if you are using FlexWord speech recognition or DPR at this prompt, you want callers to respond only after the prompt is finished to optimize the recognition. To discourage dial through/barge-in for a menu, do not leave a long pause between options. For example:

"For sales, say '1'. For service, say '2'. For an attendant, say 'zero'."

You can also discourage barge-in for a yes/no prompt. See, "Yes/No Questions with Barge-in (WholeWord Speech Recognition)," above for more information.

When barge-in is turned off or unavailable, make sure that your recorded prompts have no silence at the end. Callers may respond during the silence. When this happens, the recognizer will not hear part of the caller's response.

When using FlexWord speech recognition and/or dial pulse recognition, always discourage barge-in. You can tell callers at the beginning of the application to wait until the end of each prompt before responding. Also, use short, direct prompts so callers will not try to answer too quickly.

Dial Pulse Recognition in Applications

Use the guidelines in this section to design applications that use dial pulse recognition (DPR).

Dial Pulse Recognition Training

To improve dial pulse recognition accuracy, the system uses a method called *training*. Training allows the system to learn things about the call such as the quality of the connection, the network, and the telephone being used to place the call. Training is automatically turned on in a DPR call on the first input of a digit 5 or higher. Then it is automatically turned off for the duration of the call. It is a good idea to tell your callers to dial 9, followed by the information you need at the first Prompt & Collect the caller will hear. You can also turn the training on again during your application if needed.

Dial Pulse Recognition Digit Input

DPR supports connected-digit input. Confirmation to the caller may be necessary for dial pulse inputs. You can increase accuracy by using built-in or custom grammars. One built-in grammar looks for inputs that are three numbers apart, for example 2, 5, and 8. For yes/no inputs, ask for 2 for “yes” and 5 for “no,” or 5 for “yes” and 8 for “no.” Remember to be consistent so your callers will not be confused. Avoid asking for 1 as an input for yes/no.

Dial Pulse Recognition and Barge-in

DPR does not support barge-in. Therefore, write the prompts to discourage barge-in, as described above in, “How to Discourage Dial Through and Barge-in.”

FlexWord Speech Recognition in Applications

Use the guidelines in this section to design applications that use FlexWord speech recognition.

The FlexWord Speech Recognition software recognizes callers speaking words from a vocabulary that you define for your applications. Allowing callers to say the option they want instead of a number or “yes” or “no” can make the interaction more natural and easy to use.

How to Choose Words for Your FlexWord Vocabulary

FlexWord speech recognition allows you to use your own custom vocabulary.

Each application has its own vocabulary comprised of a set of wordlists. Only one FlexWord language can be installed on the system, but a single FlexWord language can support many applications. Applications can also share wordlists.

For each Prompt & Collect action, only one wordlist can be active (valid) for use by the recognizer. The words in each wordlist must sound different to avoid recognizer errors. Your application can use two words that sound similar, but they should be on separate wordlists, and used in a different menus.

What is a good word or phrase to use in a wordlist? The following sections discuss a number of attributes.

Length

Words (or phrases) with more than one syllable are easier to recognize than shorter words. For example, the phrase “add entry” would be better to use than the word “add.” You can include phrases of several words. If the phrases are too long, however, your callers may not remember the whole phrase when they respond.

Sound

Use words that do not sound similar to each other. To the recognizer, vowels are slightly more important than consonants. Avoid using words with similar vowel sounds. Words that are different in only one or two consonants are difficult for the recognizer to accurately distinguish between. For example, the words “connect” and “comment” sound similar to the recognizer.

Meaning

Make your wordlists meaningful and helpful to your callers. Choose words and phrases that would occur naturally to your callers. Ask some of your typical callers what words or phrases they would use for actions in your application. Do not rely only on your knowledge, and other people in your business.

Synonyms If your callers use more than one word for the same thing, add all the words to your wordlist. See “Modify a Touch-Tone-Only Application to Include Spoken Input,” above for an example of how to use more than one phrase for a single menu choice. Be sure that the synonyms you use are not confused by the recognizer with other words already in your wordlist. An exception would be when two words that sound alike also mean the same thing, and result in the same option for the caller.

Common Responses If your callers tend to respond with a word that is not on your wordlist, add that word to the wordlist. For example, if the wordlist contains the valid sizes “small,” “medium,” and “large,” you could add “petite” to the wordlist if several callers respond with this word. Then, if a caller says “petite,” the application can respond by saying “Sorry. We do not offer petite sizes. Please choose small, medium, or large.”

Digits If you want to recognize numbers like “one,” “two,” “three,” and so on, in the same application as your FlexWord vocabulary, you must have the WholeWord speech recognition software installed on your system. Do not attempt to build FlexWord vocabularies containing digits. WholeWord recognition models have been refined, and FlexWord recognition models for the same words will result in lower accuracy. Also, FlexWord speech recognition cannot recognize more than one word or phrase in a row (required to recognize connected digits).

“Yes” and “No” If you want to recognize “yes” and “no,” this is also best done with WholeWord speech recognition. If you do not plan to purchase WholeWord speech recognition, you can use a FlexWord wordlist including “yes” and “no,” but the accuracy for FlexWord speech recognition will be lower than for WholeWord speech recognition.

How to Segment Large Wordlists

Large wordlists are more likely to have words that sound alike. You can divide the list by grouping the items into smaller wordlists. Callers could be prompted in two steps:

1. Say the group word (such as a department).
2. Say the individual selection word (such as an employee name).

For example:

“Please say the department of the item you wish to purchase.”
<**Hardware**>
“Please say the name of the item you wish to purchase.”
<**Hammer**>

How to Word Prompts for FlexWord Speech Recognition

FlexWord speech recognition does not support barge-in. Therefore, write the prompts to discourage barge-in, as described above in, "How to Discourage Dial Through and Barge-in."

Bilingual, Multilingual, and Non-US English Applications

Your CONVERSANT system can support up to two different WholeWord speech recognition software in several available languages. Use the guidelines in this section to effectively use speech recognition for languages other than US English.

Bilingual Applications

The CONVERSANT system allows you to provide callers with bilingual service. The system can play prompts and announcements, and recognize speech, in several languages. In some cases you will be able to provide this type of service with a single application. In other cases, you will need to use a separate application for each language.

Selecting a Language

All bilingual services, whether simple or complex, should ask callers for their preferred language at the beginning of the application. The system can then select the application (or part of the application) that speaks and recognizes speech in the language the caller has chosen.

Use a Language Gate

The part of the application that asks the caller to pick one of two languages is called a *language gate*. Since the system can only recognize a single language for each Prompt & Collect statement, the language gate must be written carefully. The speech recognition language gate illustrated below includes two questions, one in each language. Each question asks the caller to say a vocabulary word in one of the languages. The application then branches to a different application depending on the caller's response.

You can also have a language gate for a touchtone only application. First, greet the callers in each language. Then say, for example, "For English, press one. Para Español, marque dos." The code executes a different application based on caller input.

See Figure 11 for an example of a successful language gate implementation. Figure 12 is an example of the Script Builder code to support a language gate. Note that Voice@Work restricts the application to one langauge, so a Spanish prompt would have to be performed in another application.

Figure 11.Example of a Language Gate Flowchart

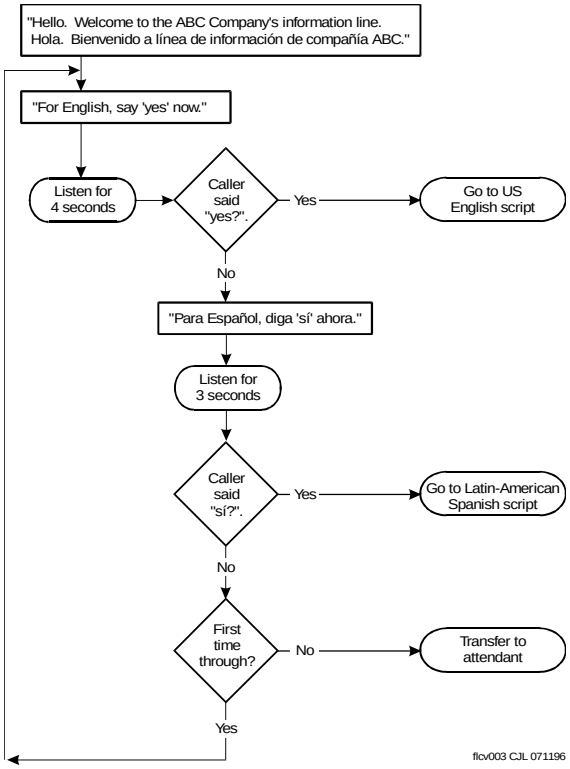


Figure 12. Example of the Script Builder Code to Support a Language Gate

```
start:
1.   Answer Phone
2.   Announce
        Speak With Interrupt
           Phrase: "Welcome / Bienvenido"
3.   Set Field Value
           Field: how_many_times = 0

     ENGLISH:
4.   Evaluate
     If how_many_times   > 1
5.      Announce
           Speak With Interrupt
              Phrase: "Transferring."
6.      Transfer To 1234 Type: Blind
7.      Quit
     End Evaluate
8.   Set Field Value
           Field: how_many_times = how_many_times + 1
9.   Prompt & Collect
        Prompt
           Speak With Interrupt
              Phrase: "For English, say yes now."
        Input
           Max Number Of Digits: 01
```

```
            RECOG: SR  US_YN
         Checklist
            Case: "Input Ok"
               Continue
            Case: "Initial Timeout"
               Goto SPANISH
            Case: "Too Few Digits"
               Goto SPANISH
            Case: "No More Tries"
               Goto SPANISH
      End Prompt & Collect
10.   Evaluate
      If $CI_VALUE   = "Y"
11.      External Action: Execute
            Application_Name:  "eng_appl"
            Write_Call_Data_Record:  "no"
            Return Field:  return_field
         End External Action
12.      Announce
            Speak With Interrupt
               Phrase: "We are experiencing technical
difficulties..."
13.      Quit
      End Evaluate

      SPANISH:
14.   Prompt & Collect
         Prompt
            Speak With Interrupt
               Phrase: "Para Espanola, Dagi si ahora.
        Input
            Max Number Of Digits: 01
            RECOG: SR  MS_YN
         Checklist
            Case: "Input Ok"
               Continue
            Case: "Initial Timeout"
               Goto ENGLISH
            Case: "Too Few Digits"
               Goto ENGLISH
            Case: "No More Tries"
               Goto ENGLISH
      End Prompt & Collect
15.   Evaluate
      If $CI_VALUE   = "Y"
16.      External Action: Execute
            Application_Name:  "span_appl"
            Write_Call_Data_Record:  "no"
            Return Field:  return_field
         End External Action
17.      Announce
            Speak With Interrupt
               Phrase: "We are experiencing technical
difficulties..."
18.      Quit
```

```
     End Evaluate
19.  Goto ENGLISH
```

Dates, Times, and Monetary Amounts

If your application *does not* need to speak dates, times, monetary amounts, or numbers to the caller, you can use a single Script Builder application to handle the interactions in both languages. Write the application in two different parts, with each part prompting and recognizing speech in a single language. When the caller chooses a preferred language in the beginning of the application, branch to the correct section.

If your application must speak dates, times, monetary amounts, or numbers to the caller, each language must be processed in a separate application. A single application can have only one set of ehanced basic speech phrases; the phrases that are used to speak numbers. Therefore, each application can only speak numbers in a single language.

Enhanced basic speech (previously called standard speech) allows the system to play commonly used words to callers from professional recordings made by Avaya. The best way to speak a numeric entry back to a caller is to use enhanced basic speech for the local language. The capability of speaking out dates, times, and monetary amounts is supported by the following enhanced basic speech languages (the voice is female unless separate male and female are indicated).

- Australian English
- Brazilian Portuguese
- Canadian French
- Cantonese Chinese
- Castilian Spanish
- Czech
- Dutch
- French
- German
- Hindi
- Hungarian
- Indonesian
- Italian
- Japanese
- Korean
- Latin-American Spanish
- Malay
- Mandarin Chinese
- Polish

- Slovak
- Thai
- United Kingdom English
- United States English (male and female)

You can install as many of these languages as you have space for on your system disks. One language will be available for each application, but your system can support several different applications.

Multilingual Applications

If you have at least two CONVERSANT systems, each with one or two WholeWord speech recognition software packages, you can process callers in up to four languages with what appears to callers as a single application. You could start with the above language gate. Instead of transferring to the attendant if the caller does not say “yes” or “sí” at one of the expected times, you could transfer the call to an application on another system. The second system could include a language gate in the other two languages, then branch to the correct application (or to the attendant) depending on the caller’s response.

Text-to-Speech in Applications

Text-to-Speech allows your application to convert ASCII text to spoken words. This ability is especially useful when it is not practical to prerecord the information to be read, for example if the information is lengthy, changes often, or comes from a database. Text-to-Speech is also useful when testing applications.

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.

Use the guidelines in this section when designing your applications with Text-to-Speech.

Use Text-to-Speech for Prompts and Announcements

Use the guidelines in this section when using Text-to-Speech for Prompts and Announcements.

Touch-Tone Input Prompts and Announcements

When Text-to-Speech is used with touchtone input, callers can interrupt, or dial through, the prompt or announcement just as they can when the prompt has been prerecorded.

If callers have problems dialing through a prompt or announcement with touchtone input, make sure that dial through is enabled. When using dial through, remember to flush all touchtone input from the buffer whenever the application plays an error message.

Spoken Input Prompts and Announcements

When Text-to-Speech is used with WholeWord speech recognition, callers barge-in during the prompt or announcement. However, when using Text-to-Speech with FlexWord speech recognition, remember that barge-in is not available. Callers cannot barge-in during Text-to-Speech prompts with FlexWord spoken input. Therefore, when accepting FlexWord speech recognition input, structure your Text-to-Speech application to discourage barge-in. See “How to Discourage Dial Through and Barge-in,” above for more information. If you have long announcements, you could prompt callers to press any touchtone key when they have heard all they want to hear of an announcement.

Use Both Text-to-Speech and Prerecorded Speech Prompts and Announcements

A single application can speak prompts and announcements in either prerecorded speech or Text-to-Speech, or a combination of the two. New callers find this more difficult to understand than prerecorded speech. Therefore, some application designers choose to speak out unchanging information (including most prompts) with prerecorded speech, unless the number of phrases makes pre-recording unmanageable. The best way to decide between these two alternatives is to test both with some of your callers. See, “How to Test Text-to-Speech Applications,” below for more information.

The following example shows how you might use both Text-to-Speech prompts and pre-recorded prompts in an application. With a wordlist including 350 names, instead of prerecording all of the names, you could use Text-to-Speech to speak the recognized name for verification. Then use prerecorded speech for the rest of the prompts. For example:

“Name, please.” (prerecorded speech)
<**John Smith**>
“John Smith” (Text-to-Speech)
“Yes or no?” (prerecorded speech)

How to Get the Most Out of Text-to-Speech

Use the guidelines in this section for better Text-to-Speech performance in your applications.

Help Callers Adjust to Text-to-Speech Output

Your callers may need to adjust to the computer voice of Text-to-Speech output to understand it well. To help them, you could speak out some less important information before speaking out the information callers need. This gives callers a chance to become familiar with the rhythm and intonation of Text-to-Speech. For example, your application could use Text-to-Speech to say “Thank you. Please wait while the information is located. The XYZ Technologies distributor closest to you is. . .” before giving the name and address of the store.

Because the sound of Text-to-Speech may be unfamiliar to your callers, consider giving them the option of having the information repeated, or spelled, if necessary. Spelling is especially useful with names, as Text-to-Speech may not pronounce names as the caller would.

Use Complete Sentences

Text-to-Speech works primarily as a “reading machine.” It operates under the assumption that the information it is reading is structured in standard English sentences (using punctuation, capitalization, subject, object, and verb). In order to make Text-to-Speech output sound most natural, use good grammar, complete sentences, and punctuation in the input text to be spoken.

What if the information you want to speak is not written in complete sentences? Since data fields cannot be punctuated, you may be able to control the output by changing the speaking rate and pauses between the information. Since Text-to-Speech cannot find standard English structure in unstructured input, the output will not be as understandable as the output of full sentences.

Note: US English 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.

Use Class Detectors

Some types of data require further clarification for the text to be spoken correctly. *Class detectors* help Text-to-Speech decide how to correctly speak out a proper name, address, telephone number, or fraction. Depending on the class detector used, Text-to-Speech speaks the data differently. Use class detectors within your application to specify the type of data to be spoken.

When using class detectors, turn them off as soon as you have finished speaking that class of information. Always leave one space before and after each class detector, and type them exactly like they appear in the documentation. Remember to use lowercase letters only.

The following is an example an application speaking out an address from database fields:

```
Announce
Speak with Interrupt
   Text: " \!nar The address of our nearest
          location is "
   Phrase: "sil.500"
   Field: store_address as A
   Phrase: "sil.500"
   Field: store_city as A
   Phrase: "sil.500"
   Field: store_state as A
   Phrase: "sil.500"
   Text: " \!nac "
```

Class detectors and their use are described in Appendix D, “Advanced Text-to-Speech Features,” in *CONVERSANT System Version 8.0 Speech Development, Processing, and Recognition*, 585-313-218.

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 5.

Table 5.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. 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.

Consider Entire Words Instead of Abbreviations

Text-to-Speech can recognize and speak out most standard English abbreviations as full words. For example, “Blvd.” is spoken as “Boulevard.” You may use an abbreviation that Text-to-Speech does not know, and therefore cannot pronounce. For an abbreviation that Text-to-Speech does not know, substitute the entire word.

If you are reading text from a remote database that you do not control, you may only be able to modify abbreviations by using a data interface process (DIP). See Chapter 3, “Data Interface Processes,” and Chapter 5, “IRAPI Programming,” in *CONVERSANT System Version 8.0 Application Development with Advanced Methods*, 585-313-216, for more information on DIPs.

Control Speaking Rate

Escape sequences and inserting pauses are two ways to control the rate at which Text-to-Speech speaks.

Escape Sequences

To help callers understand Text-to-Speech output, you can change the Text-to-Speech speaking rate where necessary by using *escape sequences* within the application. For example, you might want to slow the speech when you read an address.

When using escape sequences, always leave one space before and after each escape sequence, and type them exactly as they appear in the documentation. Be sure to turn off speaking rate escape sequences as soon as you are finished speaking at that rate.

The following is an example of an application changing speaking rate:

```
Announce
   Speak with Interrupt
      Text: "Your confirmation number is \!rslowest 743211
            \!rnormal .
      Text: Please call us if you do not receive your
      Text: tickets \!rslowest in 7 days \!rnormal ."
```

Escape sequences and their use are described in Appendix D, “Advanced Text-to-Speech Features,” in *CONVERSANT System Version 8.0 Speech Development, Processing, and Recognition*, 585-313-218.

Pauses

You can also control the pace of the Text-to-Speech output by inserting pauses. If you are speaking text you control, the easiest way to do this is with punctuation within the words to be read. Remember to punctuate exactly like you would in a sentence (for example, do not leave a space before a period or a comma). You can also use the documented escape sequences to insert pauses.

When speaking out a large block of text, you may hear a pause where you do not want a pause to be. First, check to see if there is any stray punctuation causing the pause. If not, you can insert a short recorded silence phrase *before* the sentence during which you heard the pause. This should eliminate the misplaced pause. If the text block is from a remote database, however, this may not be possible. You may need to read in the text to different buffers to take care of this problem.

Eliminate Typographical Errors

Text-to-Speech pronounces exactly what is written, so typographical errors can cause mispronunciations. To make Text-to-Speech output as understandable as possible, look for and listen for typographical errors in the ASCII text, and remove them when you test the application.

Use Capital Letters To Spell Words

Any word written with all capital letters is spelled out. Therefore, when you want Text-to-Speech to spell something out, make sure the input is in capital letters. For example, you probably want the abbreviation for the Environmental Protection Agency spelled out as “E, P, A” instead of being pronounced as “eppah.”

Note: A state abbreviation that is spoken within an address or with the address class turned on will not be spelled. For example, OH within an address will be pronounced “Ohio”, not “O H”.

If you are speaking input from a remote database and the data is in all capital letters, remember that it will be spelled out. If you want it to be pronounced as words, you must use a DIP to convert the information to upper and lower case.

Remember Text-to-speech Pronunciation Differences

Text-to-Speech relies on built-in rules, but cannot account for all exceptions. Therefore, it may mispronounce words, especially some names. If Text-to-Speech mispronounces a word, use phonetic spelling to correct the pronunciation. For example, Text-to-Speech pronounces the name “Bagge” as “baggy,” but the correct pronunciation is with a silent *e*. You can change the spelling of the name to “bag” so that Text-to-Speech will pronounce it correctly.

Another way to overcome mispronunciation or misunderstandings is to spell some of the information, especially for names. Design the application to speak the name, then spell it out. Or, you can give callers the option of having a name spelled out.

How to Test Text-to-Speech Applications

Writing applications with understandable Text-to-Speech output is a process of trial and error. It is very important to plan the application, then test it with real data to ensure that the speech is understandable. Once you can understand the output well, test the application with some representative callers, refine, and test it again until you and the callers are satisfied with the result. See Chapter 5, “Testing and Using a Voice Response Application Design,” for more information on application testing.

Using FAX Actions in Applications

This section describes some of the uses for the FAX Actions in your applications.

Note: See Chapter 8, "Using Optional Features," in *CONVERSANT System Version 8.0 Application Development with Script Builder*, 585-313-217, for more information on FAX Actions. See *Using Voice@Work,* 585-313-207, for more information on using fax capabilities within Voice@Work.

Informing Callers of Available Information

If you have a long list of information available to callers, allow callers to request that the list be faxed to them. If each fax is tagged with a number, callers who have the list can simply call in and request a fax by number, without having to listen to a menu. For example:

"Welcome to State Hospital's Pharmaceutical Information Line. Please enter the four-digit number of the medicine you would like to know about. To receive a fax listing all medicines, press the pound key."

Managing Delays

This section describes some ways to manage delays in the call when using the FAX Actions.

Process Information After the Call

You can avoid delays by using the ON_HANGUP_EVENT label within your application. Define an external function to detect hang up, specify the ON_HANGUP_EVENT as the label, and specify that label to go to on hang-up detection. All steps that come after the label are completed after the caller hangs up. This saves the caller time; however, if the processing fails, the caller is no longer on the line and so cannot be informed. To avoid this, design the application to notify the administrator (possibly through electronic mail), so that the caller can be contacted.

Store Information in Fax Files

Information stored as a text file must first be converted to fax format before being sent out as a fax. Information stored as a fax file does not need to be converted if sent with the FAX_Send action. To speed up calls, store information in fax format whenever possible. Use text format mainly for information that changes often.

Inform Your Callers of a Wait

If you know the caller must wait, use an announcement such as, "Please wait while the information is located."

For more information on how to make an unavoidable wait more pleasant, see "Pace the Application," above.

Use Fewer Fax Channels

If you find long, unavoidable delays, speed up response times by running the application on fewer channels.

Use the Correct FAX Action

If you are combining a number of text files, use the UNIX command **cat** or the utilities described in Appendix C, “Format Conversion Tools for FAX Files,” in *CONVERSANT System Version 8.0 Administration*, 585-313-510.

If you do not need to customize your cover page for every different fax, you can just use FAX_Send, rather than FAX_CovrPage and FAX_Send. When using FAX_Send, send your cover page as the first file, and the desired fax as the second file.

Assuring Faxes Are Sent Successfully

Structure your application to detect when the system might have a problem with completing an instruction. Always check the return value from an action that attempts to send a fax. If the return value indicates a problem, inform the caller that there is a problem and the fax cannot be sent.

Speaking the fax identification number, or *job ID*, back to the caller is a good way to track the delivery of a fax. If the fax is not received and the caller calls to find out why, the system administrator can use the job ID that the caller provides to trace the problem.

Exec_UNIX in Non-FAX Applications

The Exec_UNIX FAX Action gives you the power to execute commands, programs, or shell scripts, and to access databases on host computers, whether or not fax messaging is part of your application. Exec_UNIX allows you to do in Script Builder and Voice@Work what you formerly could do only with a C-language program, either using a DIP or the Intuity Response Application Programming Interface (IRAPI). If you know the UNIX shell, you can use Exec_UNIX.

For example, you may want an application to read a caller’s electronic mail over the telephone. You could first ask the caller to enter a login ID and password for security. You might then use Exec_UNIX to issue a shell command to find the first mail message in the caller’s directory, then locate the line containing the word “Subject.” You could then use Text-to-Speech to read out the subject line, and then offer the caller the option of listening to the message. If the caller chooses to listen, you could have Text-to-Speech speak the contents of the file, or skip to the next message in the mailbox.

5 Testing and Using a Voice Response Application Design

Overview

This chapter provides guidelines to test and use a voice response application design.

Purpose

The purpose of this chapter is to provide the information needed to test and use your voice response application design. Testing can provide valuable information that can be used to improve the application design, and make it more useful to your callers.

Testing the Application

There are several steps to getting your application in service. No matter how carefully you plan your application, do not expect your first design to be perfect. The most successful application designers arrive at their designs through repeated testing. The key to repeated testing is to test your application a number of times, and use the data collected during a test to refine the application design before testing again. Repeated design and testing helps determine if your application follows the guidelines in Chapter 4, "Designing a Voice Response Application."

Repeated design testing involves a cycle of activities you do until your application design meets your usability goals. This section describes the steps in this cycle of activities.

Plan a Preliminary Application Design

Planning your application design is described in Chapter 3, "Planning a Voice Response Application."

Set Usability Goals

Usability goals allow you to measure whether or not your application will serve your callers well. Different services will require different usability goals. Remember to make your goals measurable, so that your testing will give data to help you decide if the application meets the goals. Here are some examples of usability goals:

- At least 95% of test callers are able to complete the transaction in less than 45 seconds with no errors.
- At least 90% of test callers are able to recover from all errors, thereby avoiding transfer to an attendant.
- At least 98% of test callers describe the application's ease of use as very good or excellent.

Test the Preliminary Application

When you have a functioning application, it is ready to be tested. Use a small group of representative callers to accomplish a set of typical tasks. It is important to make sure that the people tested are representative of the calling population. If your application is designed for customers, do not test it with your colleagues as they may have knowledge about the application that callers do not. If you know that a large proportion of your callers is over the age of 65, use test callers that are over the age of 65.

Collect Data During the Test

You want to collect two types of data in your testing:

- Performance measurements — Two important performance measures are time (how much time callers used to complete tasks) and errors (the number and types of errors made, and where in the application they occurred).
- Opinion feedback — Important opinion measures include caller judgments about how understandable the prompts are, the ease of each task, and the overall level of satisfaction with the application.

It is very important to collect *both* types of data, since each gives you different information about the application. The data will be used to refine and modify the application design to meet your usability goals.

Usability Testing

Testing the user interaction with the application using typical callers is called usability testing. This testing will help you determine how easily the intended audience can use the application to perform specific tasks. You can adapt usability testing methods to the time, resources, and facilities available.

Testing in a Controlled Environment

The more structured usability testing methods involve bringing representative callers into a controlled, laboratory environment. Give callers a list of tasks to perform by calling into the application. Observe callers interacting with the application, videotape or record them if possible, so you can refer to the tapes later and take performance measurements (such as counting the number of errors and measuring the time taken to complete the assigned tasks). After the callers have used the preliminary application, use a questionnaire to gather their preferences and reactions to the application. See Figure 13 for a sample questionnaire.

You may also want to gather feedback by speaking personally with each caller. Be sure to ask each an identical set of questions.

Figure 13. Sample Questionnaire

Name (optional): _____
Gender: ☐ Male ☐ Female

What is your native language? _____

Overall, I rate the voice response service as	difficult to use ☐1	☐2	☐3	☐4	easy to use ☐5
Recognition accuracy is	poor ☐1	☐2	☐3	☐4	excellent ☐5
The prompts present	too little information ☐1	☐2	☐3	☐4	too much information ☐5

Did you experience any problems using the voice response service? If yes, please describe. _____
_____
What do you like best about the voice response service? _____
_____ _____
What do like least about the voice response service? _____
_____ _____

Testing in an Uncontrolled Environment

The less structured usability testing methods involve similar tasks and measurements, but do not require that representative callers be brought into a controlled environment. You can give the callers a list of tasks to perform by calling into the application, but allow them to call from their homes or offices. Measure performance by listening in to the transactions as they happen, or record them for later measurement. Use a questionnaire to gather their feedback or call them and interview them instead.

When you observe callers interacting with the application during usability testing, pay attention to everything that occurs during the call. You can observe by sitting with callers as they call in to the application, or by listening on a telephone line by using a channel monitor application available from your Avaya representative.

Observe Callers During Testing

This section describes the types of observations to make with callers.

When Callers Become Confused

You can assume a caller is confused if you hear extraneous speech (such as "What?" or "I do not understand"), response delays, or silence instead of a response. Often, you will know callers are confused by observing their facial expressions or by listening to the tone of their voice.

Information you gather from observing callers interacting with the application can help you decide how to modify the application to increase its usability. For example, you might discover that callers sound unsure or delay responding at a prompt, "What is your customer number?" To remedy the prompt, you could prompt instead with, "Your eight-digit customer number is printed on the back of your catalog. Do you have a customer number?" Callers who say no can be transferred to an attendant, while callers who say yes can be prompted next with "What is your customer number?"

When Callers Say Non-Key Words

For example, you might observe that several callers gave non-key word responses to the question "What department do you want?" In this case, you might want to list the departments after you have prompted once and no key word was recognized. The application could say "What department do you want? Please say sales, service, credit, or office."

Recognition Errors

If your application uses speech recognition, listen to what the caller says, and what the recognizer determines that the caller says. Change your application, if necessary, so that callers give you the responses you want at the right time. For example, if you do not want callers to barge-in during the prompt but they tend to barge-in, you might announce at the beginning of the transaction that callers should wait until the end of the prompt before responding, or use the word "now" in your prompts when the system is ready to accept input.

If you are using FlexWord speech recognition, substitution errors (that is, where the recognizer mistakes one word for another) are extremely important. If many callers often mistake one option for another, change one of the menu options so the recognizer does not confuse the two options. Expect to change your vocabulary before you are finished with your preliminary application design.

System Performance

Pay attention to how the system operates. Is the response time too slow? If the server stops working or other system problems occur, does the application handle the call in a graceful manner?

Refine the Preliminary Application Based on Usability Goals

After collecting performance measurements and caller feedback, study the information to see if your usability goals are accomplished. If not, let the data show you how to refine your application. If you observe caller problems with the application, use both performance and feedback data to determine the problem areas in the application. Refine your application based on your findings.

Test the Application Again

Next, test your modified application with another group of representative callers. Use the same test and gather the same data. Compare the second test results to the first, to determine if the application refinements had a positive effect on the performance measurements.

Continue Testing Until the Usability Goals are Accomplished

If your usability goals are not yet accomplished, refine the application again, based on the second test results. If you make refinements, be sure to test the application again with representative callers to determine if the changes move you towards accomplishing your usability goals.

When you are confident in your application, test the entire application. This complete test may include accessing information from a database, host computer interfaces, in-service and out-of-service hours, and other aspects of the application. You could put your application in service for a subset of callers to use for specific transactions. Collect both performance and opinion data.

Use the Tested and Refined Application in Service

When your usability goals are accomplished, place the application in service for all of your intended callers. You may want to use a single channel at a time, to see how the system performs under load. It is important to observe the acceptance of your application, and how successful your callers are using it.

It is time consuming to listen to callers interacting with your application, but you will collect valuable data about your callers and the application. This information could influence your decisions about the present application, or ones that you will write later. For example, you may have assumed that only a few of your callers would have non-US English accents, when in fact many of them do. In this case, you probably do not want to write a speech-recognition-only application. Rather, you would encourage callers to use touchtone or dial pulse input.

Even when your application is available and serving your customers, it is a good idea to continue observing the application and system performance. Listen to random calls to determine how callers are interacting with the application. Ask your customers for their opinions on how to improve the voice response services you offer.

Index

A

accuracy
- dial pulse recognition 10, 63
- FlexWord speech recognition 16
- improvement
  - dial pulse recognition 10, 63
  - speech recognition 18
- measurement 18
- Text-to-Speech 20
- touch-tone recognition 10
- WholeWord speech recognition 13

Acrobat Reader
- adjusting the window size xx
- hiding and displaying bookmarks xx
- navigating xx
- printing from xxi
- searching xx
- setting the default magnification xx

announcements
- confirmation 36
- definition 5
- design guidelines 36
- feedback 36
- prerecorded speech 71
- Text-to-Speech 71

application
- bilingual 12, 66
  - design guidelines 66
  - enhanced basic speech 69
  - language gate 66
- definition 1, 5
- design 31
  - principles 31
  - testing 79
- diagrams 27
- errors 50, 82
- multilingual 70
- pace guidelines 48
- put in service 79
- testing 79, 83
- touch-tone and speech recognition
  - modify application 60
  - separate application 54
  - single application 56
- voice response applications 1

B

barge-in 61
- caller errors 50
- definition 5
- design guidelines 61
- dial pulse recognition 63
- discouraging 62
- encouraging 62
- FlexWord speech recognition 16
- prompt wording 62
- Text-to-Speech 20
- WholeWord speech recognition 13
- with error messages 61
- yes/no questions 45

bilingual applications 12, 66
- design guidelines 66
- enhanced basic speech 69
- language gate 66
- touch-tone recognition 66
- WholeWord speech recognition 12, 66

C

caller
- attributes 31
- confusion 82
- definition 5
- feedback 80, 81
- observations 82
- short-term memory 32

checksums 43

choosing words for wordlists, see word choice

class detectors
- definition 72
- example 73
- more information 73

collecting data, see data collection

Confirm action step 37
- see also confirmation

confirmation
announcements 36
digit input 44
digit sequences 44
single digits 44
when needed 37
when not needed 39
connected digit recognition
definition 5
entry 42
custom
grammars 43
speech
definition 6

D

data collection
caller opinion 80
performance 80
data interface process (DIP) 43, 75, 76
database
lookups 44
definitions
proxy server 17
design principles 31
consistency 33
dialogue 32
error prevention 33
exits 33
feedback 33
human memory 32
know your callers 31
research guidelines 31
shortcuts 33
dial 1 prompt 54
example 54
Script Builder code 55
dial ahead 10
definition 5
Text-to-Speech 19
touch-tone recognition 10
dial pulse recognition 5
dial pulse recognition (DPR) 10, 63
accuracy 63
improvement 10, 63
training 63
capabilities 10
design guidelines 63
uses 10
dial through 61
definition 5
design guidelines 61
discouraging 62
encouraging 62
prompt wording 62
with error messages 61
digit input 41
confirming 44
digit sequences 44
single digit 44
connected-digit entry 42
constant-length sequences 41
design guidelines 41
dial pulse recognition 63
entry 42
tone-paced entry 42
variable-length sequences 41
digit sequences
confirming 44
constant-length 41
entry 42
validation 43
checksums 43
custom grammar 43
database lookups 44
variable-length 41
DIP, see data interface process (DIP)
disabled callers 26
DPR, see dial pulse recognition (DPR)

E

electronic documentation, printing xxi
enhanced basic speech 24, 69
definition 7
languages 24, 69
errors 50
barge-in 50, 61
caller 50, 82
design guidelines 50
dial through 61
message wording 51
prevention 33
speech recognition 50, 82
substitution 7, 82
escape sequences 75
example 75
more information 75
Exec_UNIX 78

F

faults, see errors
fax
  see also Script Builder FAX Actions
FAX Actions
  see also Script Builder FAX Actions
FAX_CovrPage 78
FAX_Send 77, 78
feedback
  announcements 36
  caller opinion 80, 81
  design principles 33
FlexWord speech recognition 14, 64
  accuracy 16
    improvement 18, 64
  capabilities 14
    barge-in 16
    key word spotting 16
    languages 14
    vocabularies 14
  definition 5
  design guidelines 64
  prompt wording 66
  uses 14
  vocabulary 14, 64
FlexWord toolkit
  definition 5

G

grammar, custom 43

K

key word spotting
  definition 8
  FlexWord speech recognition 16
  WholeWord speech recognition 13

L

language gate
  definition 66
  example 67
  spoken input 66
  touch-tone input 66
languages supported 20
  enhanced basic speech 24, 69
  FlexWord speech recognition 14
  Text-to-Speech 19
  WholeWord speech recognition 12

M

menu
  definition 6
  design guidelines 39
  number of choices 39
  numbered options 40
  sequence 40
  subdivided options 40
Microsoft Speech Application Programming Interface (SAPI) compliant languages 20
monetary amounts 69
multilingual applications 70

N

numerical input, see digit input

O

opinion feedback, questionnaire 81
outlines 29

P

performance
  data, collecting 80
  system 82
phrase screening
  definition 6
prerecorded speech
  definition 7
  enhanced basic speech 24, 69
  professional speaker 25
  prompts and announcements 71
Prompt & Collect action step
  definition 6

prompts 34
barge-in 62
definition 6
design guidelines 34
dial 1 54
dial through 62
digit entry 43
length 34
prerecorded speech 71
Text-to-Speech 71
wording 35, 62, 66
proxy server, defined 17
Proxy Text-to-Speech 70

R

recognition
dial pulse, see dial pulse recognition (DPR)
speech, see speech recognition
touch-tone, see touch-tone recognition
recognition type
definition 6
research, design principles 31

S

SAPI, *see* Speech Application Programming Interface (SAPI) compliant languages
screening, phrase 6
Script Builder
code example
class detector 73
dial 1 prompt 55
escape sequence 75
touch-tone with spoken input 58
yes/no question with barge-in 46
confirm an entry 37
custom grammar 43
definition 6
FAX Actions 21, 77
design guidelines 77
Script Builder FAX Actions
capabilities 22
definition 5
Exec_UNIX 78
FAX_CovrPage 78
FAX_Send 77, 78
more information 77
uses 21
selecting words for wordlists, see word choice
speech
custom 6
prerecorded
definition 7
enhanced basic speech 7, 24
professional speaker 25
prompts and announcements 71
recognition, see speech recognition
Text-to-Speech, see Text-to-Speech (TTS)
Speech Application Programming Interface (SAPI) compliant languages 20
speech recognition 11, 52
accuracy
improvement 18
measurement 18
application differences 53
definition 6
design guidelines 54, 64
FlexWord speech recognition 14, 64
recognition differences 52
used with touch-tone recognition 54
modify application 60
separate application 54
single application 56
WholeWord speech recognition 11, 54
yes/no questions 45
spell words, Text-to-Speech 76
substitution error
definition 7
testing 82
system performance 82
systems
online help support xvii

T

test the application 79
controlled environment 80
data collection
caller opinion 80
performance data 80
uncontrolled environment 81

Text-to-Speech (TTS) 19, 70
accuracy 20
capabilities 19, 76
abbreviations 19, 75
barge-in 20
language 19
output 72
pronunciation 76
speaking rate 75
speech recognition 19
spelling words 76
touch-tone recognition 19
class detectors
definition 72
example 73
definition 7
design guidelines 70
escape sequences 75
example 75
information 75
gender of speaking voice 73
performance improvement guidelines 72
prompts and announcements
spoken input 71
touch-tone input 71
rules for sentences 72
testing applications 76
uses 19
TFLUSH 48
tone-paced digit entry 42
touch-tone
recognition 9, 52
accuracy 10
application differences 53
capabilities 10
dial ahead 10
dial through 10
definition 7
used with speech recognition 54
modify application 60
separate application 54
single application 56
yes/no questions 45
touch-tone recognition differences 52
training
for dial pulse recognition accuracy 63
transaction
definition 7
troubles, see errors
TTS, see Text-to-Speech (TTS)

U

usability
definition 7
goals 80, 83
testing 80, 81
user interface
definition 7

V

vocabulary, see also wordlist 14, 64
Avaya service 15
creation 15
definition 7
example 15
FlexWord speech recognition 14, 64
size 15
word 8, 14
voice response system 2
advanced technologies 9
application design 31
application uses 2
definition 1

W

WholeWord speech recognition 11, 54, 66
accuracy 13
bilingual applications 66
enhanced basic speech 69
language gate 66
capabilities
barge-in 13
bilingual applications 12
key word spotting 13
languages 12
definition 7
design guidelines 54, 66
multilingual applications 70
uses 11
word
definition 8, 14
vocabulary, see wordlist
word choice
common responses 65
digits 65
length 64
meaning 64
sound 64
synonyms 65
yes and no 65

word spotting, see key word spotting

wordlist 15, 65
- creation 15
- definition 8, 15
- example 15
- large lists 65
- size 15

Y

yes/no questions 45
- barge-in 45
- design guidelines 45
- spoken input 45
- touch-tone input 45