Lucent Technologies
Bell Labs Innovations

Using Voice@Work

585-313-207
Comcode 108541103
Issue 4
July 1999

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

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

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

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

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

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

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

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

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

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

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

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

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

Trademarks
See the section titled "About This Book."

Ordering Information

Call:	Lucent Technologies Publications Center	
	Voice 1 800 457-1235	International Voice 317 322-6791
	Fax 1 800 457-1764	International Fax 317 322-6699
Write:	Lucent Technologies Publications Center	
	2855 N. Franklin Road	
	Indianapolis, IN 46219	
Order:	Document No.585-313-207	
	Comcode 108541103	
	Issue 4, July 1999	

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

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

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

EMC Directive	89/336/EEC
Low-Voltage Directive	73/23/EEC

The "CE" mark affixed to the equipment means that it conforms to the above directives.

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

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

Contents

About This Book

Purpose of This Book

The purpose of this book is to help the user find information about using Voice@Work quickly and easily.

Another purpose of this book is to provide the user with additional information about using Voice@Work more effectively and efficiently.

Intended Audiences

The primary audience for this book are those who use Voice@Work to develop caller-response applications for INTUITY™ CONVERSANT® systems.

Secondary audiences include personnel from the following Lucent organizations:

- Solutions Development Organization (SDO)
- Field support — Technical Service Center (TSC)
- Training

We assume that the primary users of Voice@Work are familiar with the operation of their target voice response systems.

We also assume that primary users of Voice@Work are familiar with their Windows operating systems and common procedures such as clicking, double-clicking, and dragging. If you are not familiar with these procedures, see the documentation for your operating system.

Finally, we assume that primary users of Voice@Work are familiar with basic application development and programming techniques and concepts.

Release History

The first issue of this book, for Voice@Work Release 2.0, was released in May, 1998.

The second issue of this book, for Voice@Work Release 2.1, was released in August, 1998.

The third issue of this book, for Voice@Work Release 3.0, was released in December, 1998.

This is the fourth issue of this book.

Trademarks

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

- CONVERSANT, DEFINITY, and FlexWord are registered trademarks, and INTUITY and @Work Studio are trademarks of Lucent Technologies, Inc.
- Microsoft, Windows NT, Windows 98, Windows 95, and WinHelp are registered trademarks of Microsoft Corporation. Notepad and WordPad are copyrighted products of Microsoft Corporation.
- UNIX is a registered trademark of the X/Open Company.
- ORACLE is a trademark of the Oracle Corporation.
- Sound Blaster is a registered trademark of Creative Technologies Ltd.
- InstallShield is a registered trademark of InstallShield Corporation.

How to Use This Book

Virtually all of the content of this book is also included in the online Help system. This means that you do not need to rely on this book to find the information you need to use Voice@Work effectively.

In general, this book is organized to reflect the architecture of Voice@Work itself. You may use the Table of Contents to view the major areas of the book and get a feel for its general organization.

This book also includes a glossary and a comprehensive index to assist you in finding the information you need quickly.

This book is divided into major sections which group two or more chapters according to their content. Each major section has its own sectional table of contents.

Within each chapter, the Purpose section at the beginning offers a list of the major topics within that chapter.

Lucent Technologies recommends that you read the first section (General Information, Chapters 1–4) in its entirety before using Voice@Work for the first time, especially if you have not used a product like Voice@Work before. This will give you a general understanding of how to use Voice@Work most effectively.

Specifically, the first four chapters have been written to help you with the following:

- Chapter 1, "Introduction to Voice@Work" describes system requirements for installing Voice@Work, some of the major features, and installation procedures.
- Chapter 2, "Getting Started with Voice@Work" helps you in getting started using Voice@Work for the first time and in configuring your workspace to your liking.
- Chapter 3, "Creating an Application Using Voice@Work" provides a model and the basic steps for creating a Voice@Work application. If you want an example and help in building a sample application, see Appendix G, "Building a Sample Application" as well.
- Chapter 4, "Using the Online Help" gives information about Voice@Work's extensive online help system, including tips for using the online help effectively.

The other sections of the book are intended for use primarily as reference chapters. They typically offer procedures for performing various tasks and using the tools and features of Voice@Work.

Terminology and Conventions Used

The following font, terminology, and keyboard conventions are used throughout this book. Familiarize yourself with these conventions before using this book.

The Use of Bold Type

Using Voice@Work uses bold type to indicate:

- Menu, option, field, or window title names, exactly as they appear on screen, when they are used as part of a procedure
- Text you should type exactly as given

Terminology Used

The following terminology has been used throughout this book:

- The term "type" means to press the key or sequence of keys exactly as specified.

  For example, an instruction to type the letter "y" is shown as

  Type **y** to continue.

- The term “enter” means to type a value, then press the ENTER or RETURN key on your keyboard.

  For example, an instruction to move the cursor to the **Length** field, then type the number "24" and press ENTER is shown as

  In the **Length** field, enter **24**

- The term "select" means to move the cursor to a desired item on the screen, then click it to highlight or activate it.

  For example, an instruction to move the cursor to the Print option on the File menu and click the left mouse button to activate it is shown as

  From the **File** menu, select **Print**.

- The term "click" means to move the cursor to a desired item on the screen, then press the primary (usually the left) mouse button once.

  For example, an instruction to move the cursor to the "OK" button in a window and press the primary mouse button is shown as:

  Click **OK**.

- The term "double-click" means to move the cursor to a desired item on the screen, then press the primary (usually the left) mouse button twice in rapid succession.

  For example, an instruction to move the cursor to the name of a table and select it by pressing the mouse button twice in rapid succession is shown as:

  Double-click the name of the table you want.

- The term “right-click” refers to clicking an object or menu item using the *secondary* (usually the right) mouse button.

  For example, an instruction to open a special popup menu is shown as:

  Right-click anywhere in the table.

- The terms “right-click menu” and “right-click popup menu” refer to the menus that appear in many Voice@Work windows when you click in a particular area using the *secondary* (usually the right) mouse button.

- The term "drag-and-drop" means to move the cursor to a desired item on the screen, and then press the primary (usually the left) mouse button and hold it while moving the cursor to a desired new location on the screen. Then release the mouse button to "drop" the selected item at the new location.

  For example, an instruction to move the cursor to a phrase (in a phrase table), press the mouse button, hold it down while moving the cursor to a prompt, and release the mouse button to "drop" the phrase in the prompt is shown as:

  Drag-and-drop the phrase into the prompt.

- The term "click-and-drag" is similar to "drag-and-drop", except that you do not release the mouse button right away when the item you are dragging reaches the desired destination. The "drop" in such cases is treated as a separate step.

- In addition the following terms are used:

  window — refers to a standard Windows OS window as part of the graphical user interface (GUI)

  dialog box — refers to a window in which you are asked to fill in information or make choices before closing it and returning to the main window

  menu — refers to a list of choices offered at the top of a window

  menu bar — refers to a set of menus displayed at the top of a window

  icon — refers to a pictorial graphic object that can be used to perform some action

  button — refers to a text-based graphic object that can be clicked to perform some action

  target voice response system — refers to the system on which the Voice@Work application is to be installed. For this release, the target voice response system must be either an INTUITY CONVERSANT V6 or V7 system.

  See the Glossary for additional terms and definitions.

Keyboard Conventions

Keyboard symbols and shortcuts are shown using the following conventions:

- ALT refers to the "Alt" (Alternate) key
- CONTROL refers to the "Ctrl" (Control) key
- F# refers to the "F" or "Function" keys at the top of the keyboard (# is replaced by a number, 1–12)
- DEL refers to the "Delete" key

- When you are to press two (or more) keys simultaneously, the instruction is similar to the following:

  To close the window, press ALT F4

- "Directional arrows" refers to the four arrow keys commonly found to the right of the alphanumeric keys, designated in this book as follows:

◄	Left arrow
►	Right arrow
▲	Up arrow
▼	Down arrow

- Underscores in item names indicate keyboard shortcuts as well according to standard Windows conventions (see your Windows documentation for more details).

  Examples: **File** tells you to press the ALT and F keys at the same time.

  Print (on the **File** menu) tells you to press the P key.

- Text that explains the subject or offers additional information is designated by the following symbol:

  NOTE:
  Note text goes here.

Technical Support

For additional help with Voice@Work, contact your Lucent Technologies sales and service representative.

For immediate help with Voice@Work problems, customers within the United States and Canada can call the Intuity CONVERSANT group Technical Service Center at:

1-800-242-2121

All other customers, call the local Customer Support Representative or Lucent Technologies Center of Excellence for your region.

Or, visit the Lucent Technologies Customer Self-Service Center on the World Wide Web at:

http://www.lucent.com/enterprise/selfservice/

How to Order Documentation

The documentation for Voice@Work comes complete on your Voice@Work CD-ROM. The documentation is available in two forms:

- The online Help system, which contains virtually the same content as this book.
- This book in the form of Adobe Acrobat .pdf files.You may print out paper copies of this book for your own use with Voice@Work. You may not, however, produce copies for any other purpose.

You may also order a paper copy of this book from the Lucent Technologies Publications Center, using the title of this book and the document number, 585-313-207.

Lucent Technologies Publications Center

Voice	1 800 457-1235	International Voice	317 361-5353
Fax	1 800 457-1764	International Fax	317 361-5355

P.O. Box 4100
Crawfordsville, IN 47933

How to Comment on This Document

This documentation was produced by Product Publications, Lucent Technologies, Columbus, OH.

We are interested in your suggestions for improving the Voice@Work documentation. Please feel free to print out, complete, and return the Users' Comments and Feedback form, which is located in the back of this book.

Please send your comments and suggestions to:

Lucent Technologies, Inc.
GLS Information Development Organization
Room 22-2H15
11900 North Pecos Street
Denver, CO 80234

You may also fax your comments to the attention of the Lucent Technologies GLS Information Development Organization at:

(303) 538-1741

Please mention the product name Voice@Work when sending your comments.

Introduction to Voice@Work

1

Overview

Lucent Technologies' Voice@Work allows you to create applications for voice response systems using a Windows NT 4.0, Windows 98, or Windows 95 environment. This graphically oriented tool make it easy for developers to create voice response system applications from a PC environment.

Voice@Work is part of the new @Work Studio™ suite of application development products from Lucent Technologies, Inc.

Purpose

The purpose of this chapter is to provide you with general introductory information about Voice@Work, including the following topics:

- Intended Users
- Features
- System Requirements
- Installing Voice@Work on Your PC
- Installing the SCI Tools on the INTUITY CONVERSANT System
- Language Support
- The Graphical User Interface

Intended Users

Voice@Work is designed to aid those who develop caller response applications for INTUITY™ CONVERSANT® Version 6 and Version 7 systems. This includes:

- Value added resellers (VARs) and independent software vendors (ISVs): These companies develop applications for customers with varied needs in terms of size and application. They produce both custom (used only by a single customer) and off-the-shelf (prepackaged) solutions.
- Software development departments within an end-customer company: Some firms, usually large in size, staff software development departments to write software for and maintain data and telecommunications systems. They are very focused on the needs of their company only. They are usually college-educated in computer science and have some professional development experience in computer languages required by their company; this could be anything from C and C++ to Cobol.

We assume that the primary users of Voice@Work are familiar with the operation of their target voice response systems.

We also assume that primary users of Voice@Work are familiar with their Windows operating systems and common procedures such as clicking, double-clicking, and dragging. If you are not familiar with these procedures, see the documentation for your operating system.

Finally, we assume that primary users of Voice@Work are familiar with basic application development and programming techniques and concepts.

Features

Major features of Voice@Work include:

- An easy-to-use, customizable graphical user interface (GUI)
- Standard Windows-type file operations
- A variety of tools to edit, diagnose, and debug applications
- A number of resource managers used to create and edit global resource elements
- Extensive use of right-click menus
- Cut/Copy/Paste capabilities across call flows and applications
- An Inline Code node that allows you to insert TAS code into your application “as is”
- The ability to share resources with other developers
- The ability to specify the use of more than one local or remote databases in applications

- Comprehensive language support
- Powerful Code Generation and Application Transfer tools
- External functions for Universal Call Identification (UCID), Nabnasset, Geotel, and Lucent Fax included as part of the standard external functions set
- An extensive online Help system

A Word about Y2K Compliance

Lucent Technologies, Inc., has determined that Voice@Work is Year 2000 compliant.

An Explanation of "Year 2000 Compliant"

By "Year 2000 compliant", Lucent means that, in the year 2000, Voice@Work will properly perform the functions described in its specifications and documentation related to the handling of calendar dates.

Specifically, Lucent has certified Voice@Work in the following particulars:

1. Voice@Work will continue to operate in the year 2000.
2. Voice@Work will provide valid and correct results, free of material errors, with reference to the input, processing, and output of date data which references a given century, provided that interfacing equipment or databases properly exchange date data with Voice@Work.

   NOTE:
   By "century", Lucent means the period 1900 through 1999 or the period 2000 through 2099.

3. Voice@Work will maintain existing date interfaces with other equipment, unless otherwise noted in the date usage remarks below.
4. Voice@Work, when processing date data, will include the following:
   a. Internal system date data century recognition
   b. Functions which perform calculations using same-century date values
   c. Recognition of the year 2000 as a leap year

Date Usage Remarks

The Simulation tool fails when selecting date records from database tables. This does not affect Intuity CONVERSANT-installed scripts.

Other Remarks

This statement of year 2000 compliance applies only to Voice@Work and does not apply to any other equipment that may interoperate with it. Lucent's certification of year 2000 compliance applies provided that it has been installed and operated in accordance with Lucent's specifications and that it has not been modified in any way.

System Requirements

The following section details the hardware and software requirements for Voice@Work.

Hardware Requirements

The following is the *minimum* hardware configuration for Voice@Work:

- Pentium 90-MHz processor or equivalent
- 64 Mbytes of RAM for Windows NT 4.0; 32 Mbytes of RAM for Windows 98 or Windows 95
- 30 Mbytes of free hard disk space

  NOTE:
  If you plan to record your own phrases, you may need considerably more hard disk space.

- CD-ROM drive, any speed
- 1.44-Mbyte floppy disk drive
- VGA video display
- Microsoft-compatible mouse
- Standard keyboard

The following is the *recommended* hardware configuration for Voice@Work to run most efficiently:

- Pentium Pro 200-MHz processor or equivalent for Windows NT 4.0; Pentium 166-MHz processor or equivalent for Windows 98 or Windows 95
- All other components as listed above.

Optional Hardware

The following hardware is *not required*, but can enhance your ability to use Voice@Work more effectively:

- Sound Blaster-compatible sound card capable of 16-bit monaural sampling at 8 KHz
- Microphone (see "Audio Recording Tips" in "The Phrase Editor" in Chapter 18, "Working with Phrases" for more information)
- Speakers (see "Audio Recording Tips" in "The Phrase Editor" in Chapter 18, "Working with Phrases" for more information)
- Microsoft-standard networking hardware for sharing resources and for transferring files to the target system

NOTE:
If you want to transfer files to the target system using networking, the target system must also be connected to the network.

For information about installing and using optional hardware see the documentation for the software you plan to use.

Software Requirements

The following are the *minimum* software requirements for Voice@Work:

- Voice@Work Version 2.1
- PC operating system (Windows NT 4.0, Windows 98, or Windows 95)
- Arial True Type font set, or other True Type sans-serif extended ASCII character set (needed to make the online Help display properly)

Optional Software

The following software is *not required* to use Voice@Work but can enhance your ability to use Voice@Work more effectively:

- Digital-audio editing software for editing audio files (see "Audio Recording Tips" in "The Phrase Editor" in Chapter 18, "Working with Phrases" for more information)

- Microsoft-standard networking software and TCP/IP stack for sharing resources and for transferring files to the target system

  NOTE:
  If you want to transfer files to the target system using networking, the target system must also have appropriate networking software installed.

  For information about installing and using optional software see the documentation for the software you plan to use.

System Settings

For optimal performance, Voice@Work is designed to work with the following settings on your computer:

- 256 (8-bit) color
- 1024 x 768 monitor display

NOTE:
These settings mean that Voice@Work may be difficult to use on some portable computers.

- Small fonts (system settings)

  NOTE:
  Do not use the “Large Fonts” setting in the Control Panel, Settings tab. Voice@Work does not display properly if large system fonts are used.

Installing Voice@Work on Your PC

To install Voice@Work on your PC, do the following:

1. Insert the Voice@Work CD-ROM into the CD-ROM drive.
   - If your computer is configured to allow autoplay of CDs, the system displays the Voice@Work browser window.
   - If the browser window does not appear within a few seconds, use Windows Explorer to locate the **setup.exe** on the CD. Double-click **setup.exe**. The system displays the Voice@Work browser window.
2. Select **Install Voice@Work** from the selection menu.

   The InstallShield Wizard runs and the system displays the Voice@Work **Welcome** window.
3. Click **Next.**

   The system displays the **Voice@Work Setup** window.

4. Click **Next**.

   The system displays the **Setup Type** window.

5. Select the type of setup you prefer from the three options described in Table 1.

Table 1. Voice@Work Installation Options

Option	Description/Comments
Typical	Installs all required Voice@Work files, along with all supported languages. Recommended for most users.
Compact	Installs basic Voice@Work files, but installs only the US English–female language package. Other languages must be installed separately.
Custom	Allows you to select which Voice@Work options you want to install. For example, you may choose to install only one or two languages from all those available. Or, if your PC already has the Borland Database Engine installed, you may choose not to install it.

6. If you want to select a destination other than the default (C:\Program Files\Lucent\@Work Studio\Voice@Work), click the **Browse** button in the **Destination Folder** area and select the desired destination path.

7. Click **Next**.

   If you chose the **Custom** setup option, the system displays the **Select Components** window. Continue with Step 8.

   If you did not choose the **Custom** setup option, skip to Step 9.

8. If you chose the **Custom** setup option, select the components you want to install.

9. If you have Lucent Technologies' Graphical Designer Version 1.6.1 installed on your PC, the system displays Graphical Designer Copy dialog box (Figure 1).

   If you do not have Graphical Designer installed on your PC, skip to Step 10.

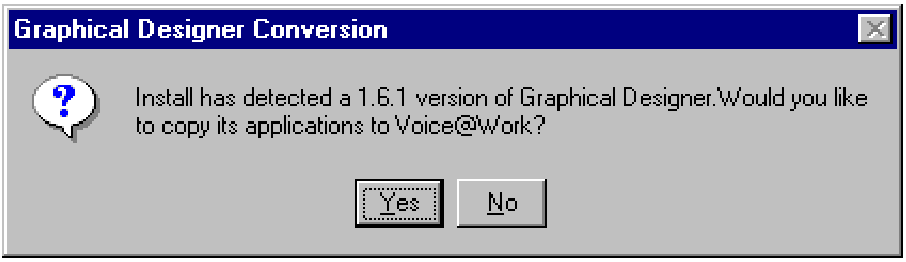


Figure 1. Graphical Designer Dialog Box

If you want to be able to convert Graphical Designer Version 1.6.1 applications to Voice@Work applications, click **Yes**.

NOTE:
The actual conversion of Graphical Designer applications does not take place at this point. The Graphical Designer applications are merely copied to a new directory. Conversion of the applications takes place when you attempt to open them (see Appendix E, "Converting Graphical Designer Applications" for more information on converting Graphical Designer applications).

10. The system displays the **Select Program Folder** window.

    Choose one of the following options:

    - Use the default (do nothing here, but proceed to Step 11).
    - Type in a new name for the folder where you want the program files to go.
    - Select from the list of **Existing Folders**.

11. Click **Next**.

    Voice@Work is installed on your PC according to the options you selected.

    When the installation process is nearly complete, the system displays **Default Language Selection** window.

12. Select the language you want to install as the default language.

    By default, the **US English - Female** box is checked. If you want **US English - Female** as the default language, do not check anything in this window.

    NOTE:
    Whatever you select here will be the default language for all new applications until you change it in the Preference Editor (see "The Application Parameters Tab" in Chapter 27, "The Preference Editor" for information on changing the default language).

13. Click **Next**.

    The system displays the Shared Resources Notice (Figure 2):

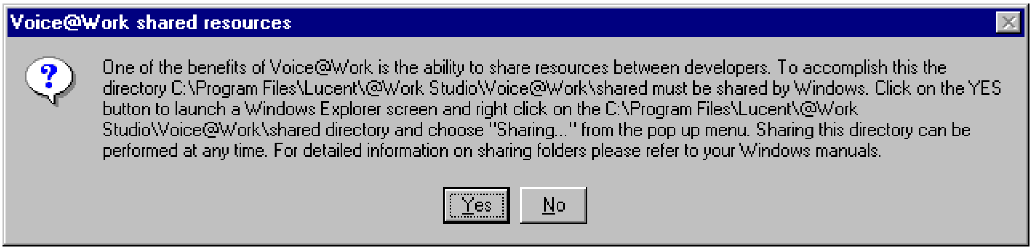


Figure 2. Voice@Work Shared Resources Notice

14. If you want to designate the **Shared** directory as a shared directory at this time, click **Yes** and follow the procedures for designating a directory as shared.

    If you want to designate the Shared directory as a shared directory later, click **No**.

    NOTE:
    For details about sharing directories, see the documentation for your Windows operating system.

    The InstallShield Wizard displays the "readme" file for Voice@Work.Read it for any changes that were made to the product after the documentation release.

15. When you have read the "readme" file, click **Next**.

    The system displays the Setup Complete window, notifying you that Voice@Work is successfully installed on your PC.

16. If you want to launch Voice@Work immediately, check the **I would like to launch Voice@Work** checkbox.

    If you want to launch Voice@Work later, uncheck the **I would like to launch Voice@Work** checkbox.

17. Click **Finish**.

Installing the SCI Tools on the INTUITY CONVERSANT System

The Service Creation Installation (SCI) tools are used to install Voice@Work applications on an INTUITY™ CONVERSANT® Version 6 or Version 7 system. For details about using the SCI tools, see Appendix A, "Using the SCI Tools" in *Using Voice@Work*, 585-313-207.

Checking for the SCI Tools

If the SCI tools are not present on the CONVERSANT, you must install them before you can install your Voice@Work applications.

To determine whether the SCI tools are already installed, do the following on the INTUITY CONVERSANT system:

1. At the Unix prompt, enter **pkginfo sc**

   If the SCI tools are installed on the INTUITY CONVERSANT, the system displays the following message:

   ```
   intuity sc  INTUITY Service Creation Integration
   Package
   ```

   If the SCI tools are *not* installed on the INTUITY CONVERSANT, the system displays the following message:

   ```
   UX:pkginfo: ERROR: information for sc was not found
   ```

Installing the SCI Tools

To install the SCI tools on the INTUITY CONVERSANT system, do the following:

1. At the Unix prompt, enter **pkgadd -d diskette1**

   The system displays the following message:

   ```
   Insert diskette into Floppy Drive 1.
   ```

   ```
   Type [go] when ready,
   or [q] to quit: (default: go)
   ```

2. Insert the Voice@Work floppy diskette labeled "INTUITY Service Creation Integration Package".

3. Press ENTER.

   The system displays the following message:

   ```
   Installation in progress.  Do not remove the diskette.

   The following packages are available:

     1 sc  INTUITY Service Creation Integration Package

   Select package(s) you wish to process (or 'all' to
   process all packages). (default: all) [?,??,q]:
   ```

4. Press ENTER.

   After a series of information messages, the system displays the following message:

   ```
   Installation of INTUITY Service Creation Integration
   Package (sc) was successful.

   Insert diskette into Floppy Drive 1.

   Type [go] when ready,
   or [q] to quit: (default: go) q
   ```

5. Enter **q**

6. Remove the floppy from the drive.

Language Support

Voice@Work provides support for all INTUITY CONVERSANT V6 and V7 languages, including all:

- Enhanced Basic Speech (EBS) languages and formats
- WholeWord speech recognition packages
- FlexWord speech recognition packages

You select the default language for Voice@Work at the time of installation. You can change the default language at any time using the Preference Editor.

You can only select one language per application.

NOTE:
Voice@Work has not yet been translated into other languages. Thus, regardless of the language you select, the windows and dialog boxes in Voice@Work remain in US English. The exception to this rule is that, if you have selected another language as the default or primary language within your Windows operating system, Windows-standard windows will be in the language for the operating system.

In addition, Voice@Work is designed to operate with keyboard input using the US English keyboard character map settings. If you have mapped your keyboard to use another language's keyboard character map, you may not get the proper input from your keyboard.

For more details on selecting or changing the default language for Voice@Work applications, see Chapter 27, "The Preference Editor".

The Graphical User Interface

The graphical user interface (GUI) is, for the most part, based upon standard Windows conventions and practices. This includes the use of clickable icons or buttons, many common keyboard shortcuts, and the extensive use of right-click menus.

Voice@Work's GUI also has many customizable features, which you will find described throughout this book. Of particular interest are the following:

- The ability to dock and undock toolbar palettes
- The capability to create new external functions groups/tabs
- The option to view encapsulated call flows using a Multiple Document Interface, which has multiple windows, or a Tabbed Interface, which uses tabs to access different call flow windows
- The ability to resize the Globals Manager and to dock/undock individual resource managers
- The choice to specify when the Node Inspector is displayed

When you first open Voice@Work, a window similar to the following appears (Figure 3).

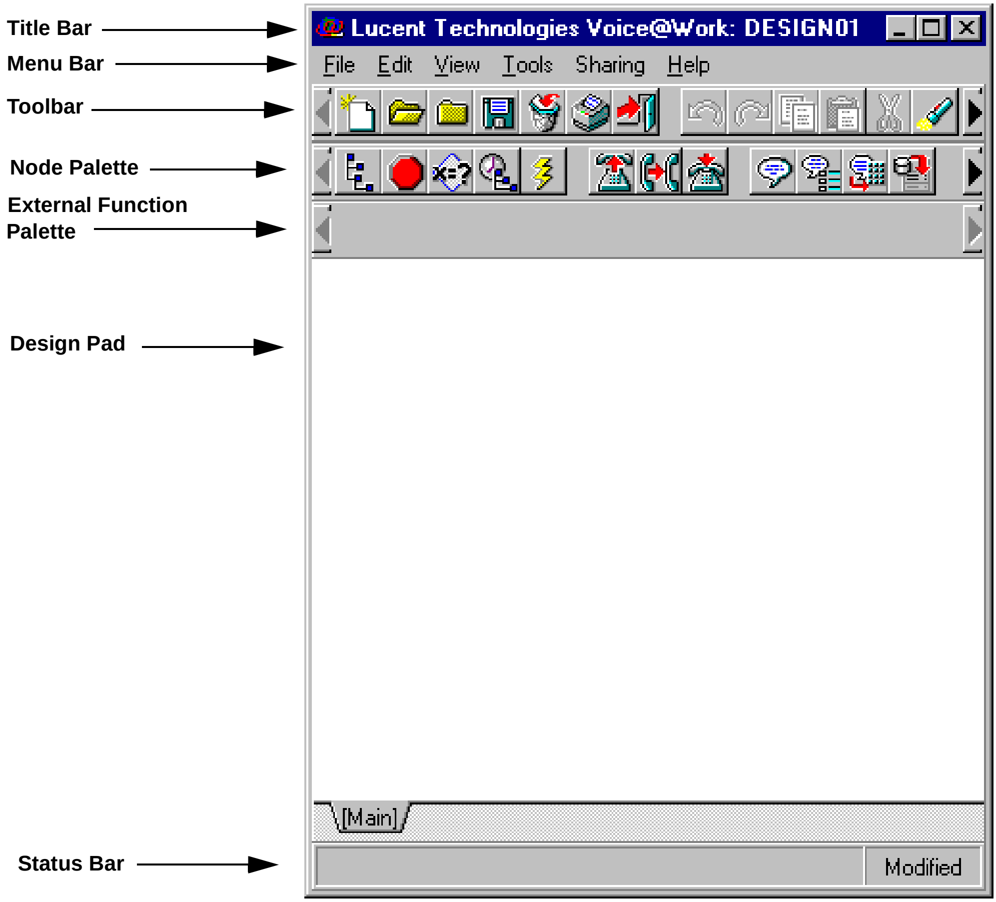


Figure 3. Voice@Work Main Window

At the top of the window is the title bar. This area includes the name of the current application and the minimize, restore, and close window buttons.

Just below the title bar is the menu bar. The menus on the menu bar have the most common commands used in creating an application.

Just below the menu bar are the palettes. There are three palettes, each with its own set of icons. These palettes are:

- The Toolbar
- The Node Palette
- The External Function Palette

These palettes can be docked to or undocked from the main window, configured separately from the others, and customized. See Chapter 7, "The View Menu" and Chapter 12, "Palettes" for details.

The main area, where application call flows are created, is called the design pad. This is where you do most of the work in creating an application.

At the bottom of the window is a status bar, where context-sensitive help or status messages display.

Getting Started with Voice@Work

2

Overview

Voice@Work offers powerful and flexible options for controlling your workspace. Some of these options include:

- Setting global preferences for using the workspace
- Using multiple nonmodal windows
- Customizing the palettes

Most of these options you will want to set once and leave them, but you can change any of them at any time.

Purpose

The purpose of this chapter is to orient you to Voice@Work workspaces and introduce you to some of its many customizable features. This chapter touches on topics of interest primarily to the first time user in a procedural overview format. The intent is *not* to cover each topic in detail, as these are described elsewhere in this book, but rather to provide you with a starting point after installing Voice@Work. Topics in this chapter include:

- Opening Voice@Work the First Time
- Setting Global Preferences
- Using Multiple Windows
- Configuring the Palettes
- Where to Go from Here

Opening Voice@Work the First Time

When you first start Voice@Work, the main window appears (Figure 4). Notice that, because you do not yet have an application open, the design pad and many of the Toolbar icons are inactive, and the External Functions Palette is empty.

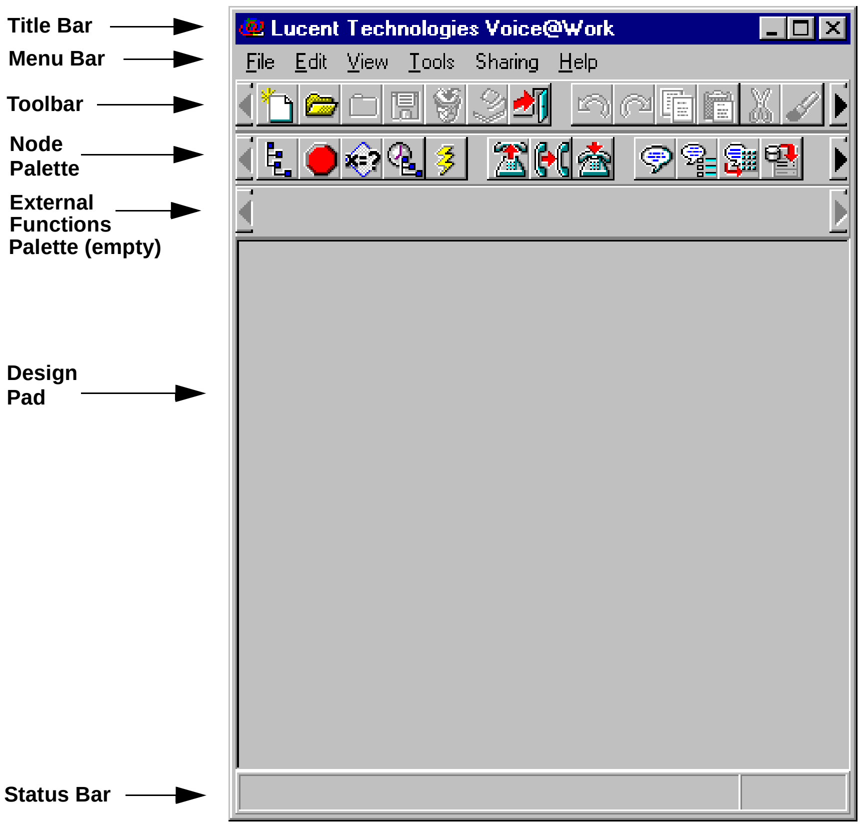


Figure 4. The Voice@Work Main Window

For a description of the components of the main window, see "The Graphical User Interface" in Chapter 1, "Introduction to Voice@Work".

Before you begin building your first Voice@Work application, it is a good idea to configure your workspace to your liking. The following sections will help you to do that.

Setting Global Preferences

Until you open or create an application, most tools and options are inactive. The Preference Editor is one exception to this rule. The Preference Editor allows you to set application and toolwide preferences for using Voice@Work without an application open.

This section introduces you to some of the preferences you can set using the Preference Editor. Specifically, you will:

- Turn on the Autosave feature
- Check to make sure the Autoname and Tabbed Interface options are on
- Modify the printing options to send printed reports to a file, rather than to a printer

While there are many other preferences you can set using the Preference Editor, they are not covered here. For more information about these preferences and using the Preference Editor, see Chapter 27, "The Preference Editor".

1. From the Tools menu, select **Preference Editor**.

   Voice@Work displays the Preference Editor window (Figure 5). Notice that there are three tabs at the top of the editor.

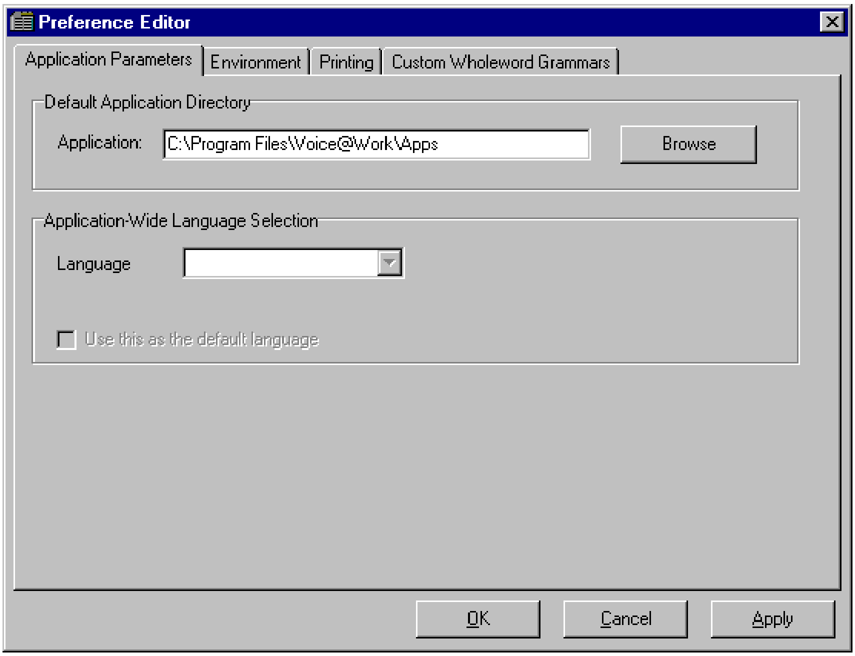


Figure 5. Preference Editor Window

The active tab is the Application Parameters tab.

2. Click the Environment tab.

   The Preference Editor window changes similar to Figure 6. This tab includes a variety of options for opening and saving applications, using the spell checker, and working within the design pad.

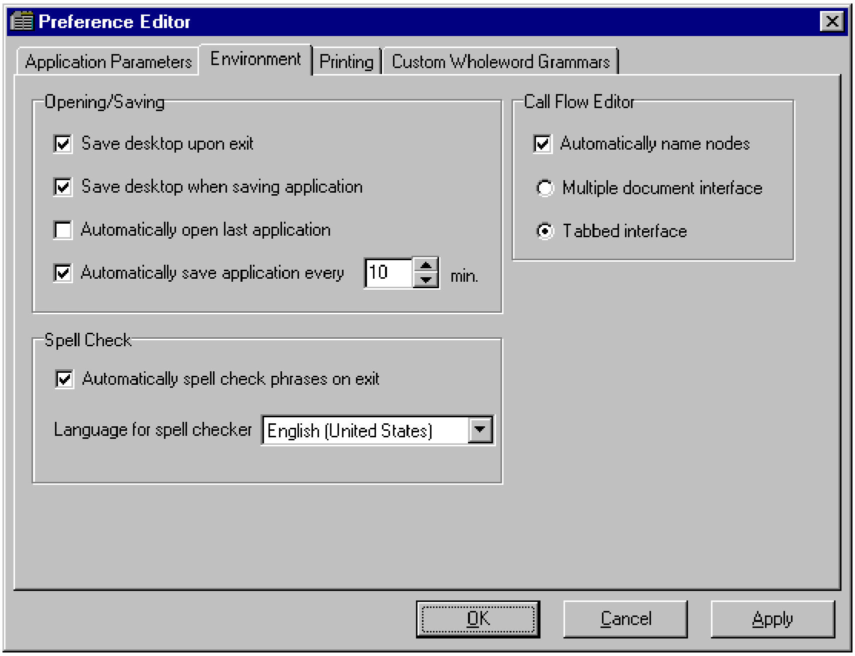


Figure 6. Preference Editor — Environment Tab

3. For now, make only three changes on this tab:

   - In the Opening/Saving box, place a check mark in the **Automatically save application every** checkbox, and enter **10** in the **min.** field.

     This option, sometimes called Autosave, causes Voice@Work to save your application automatically every 10 minutes, so that you never lose more than 10 minutes worth of work, in case of a power outage or system failure.

     NOTE:
     While selecting the Autosave option can help you in case of power outage or system failure, be aware that the Autosave option overwrites the last save action on your application. This is not always desirable.

For example, you may be experimenting with an application, but you do not necessarily want to keep the results of your experiments. If you select the Autosave option, your experimental version replaces the original version the first time the Autosave action is performed.

- In the **Call Flow Editor** box, select the **Automatically name nodes** checkbox, if it is not already selected.

  This option allows Voice@Work to name nodes automatically when you drag them onto the design pad. You will see more how this works in the "Opening the Node Inspector Window" section of "Using Multiple Windows" later in this chapter.

- In the **Call Flow Editor** box, make sure the **Tabbed Interface** radio button is selected, if it is not already.

  This option causes each call flow in your application to display using the entire design pad. Each call flow is identified with its own tab at the bottom of the design pad. You then select different call flows by clicking their tabs at the bottom of the design pad.

4. Click the **Printing** tab.

   The Preference Editor window changes similar to Figure 7. This tab contains options for printing reports in Voice@Work. The initial default is that all options on this tab are selected.

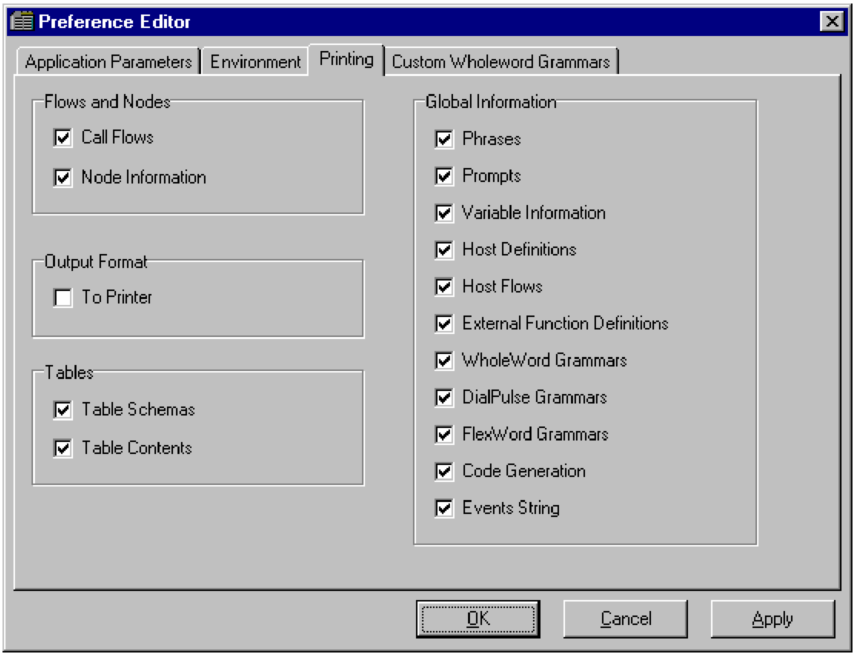


Figure 7. Preference Editor — Printing Tab

5. If there is a check mark in the **To Printer** checkbox, click in the checkbox to remove it.

   Removing the check mark from the **To Printer** checkbox causes Voice@Work to save all reports to a *.doc file by default when you select the **Print...** option on the File menu. You can then view and edit the contents of that file using a word processor before printing it out.

6. To confirm your choices and close the Preference Editor, click **OK**.

For more information about the Preference Editor and its options, see Chapter 27, “The Preference Editor”.

Using Multiple Windows

Voice@Work uses a multiple nonmodal window interface. This means that you can have several windows open at one time, and you can switch easily among them by clicking inside the window in which you want to work.

As you will see, the ability to open and use multiple windows is one of Voice@Work’s most powerful features. It gives you the ability to set up your workspace in whatever configuration works best for you.

Many windows, however, are not active or available until an application is open. To see the power and flexibility of multiple nonmodal windows, you must first create a new application.

Creating a New Application

To create a new application, do the following:

1. Click the  icon on the Toolbar.

   Voice@Work displays the New Voice@Work Application window (Figure 8). Notice that Voice@Work suggests a default name for the new application, in this case **design01**. You can change this name, but for this example, use the default.

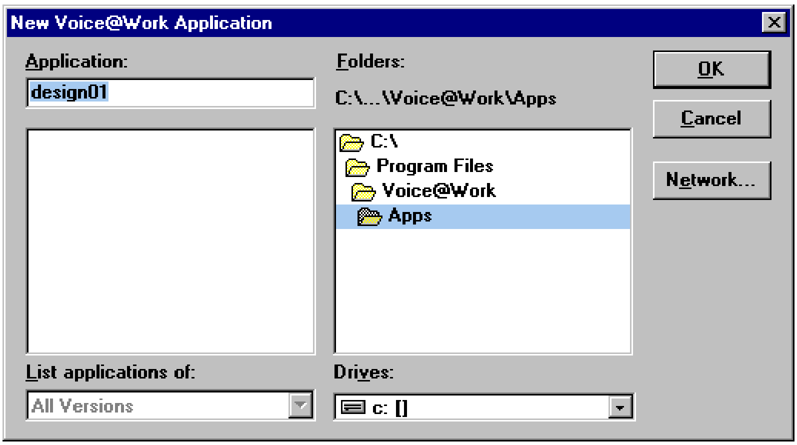


Figure 8. New Voice@Work Application Window

2. Click **OK**.

   The main window changes similar to Figure 9.

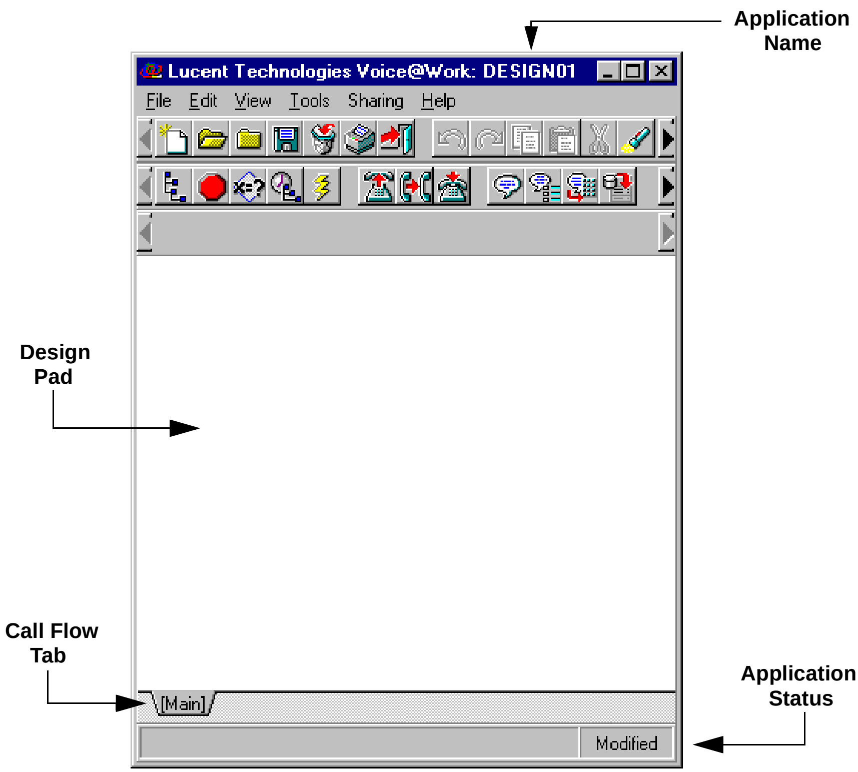


Figure 9. Voice@Work Main Window with Application Open

Notice that more of the Toolbar icons are now active, for example, the Save and Print icons. Notice also that the design pad is now white, and the application name appears in the title bar.

The application status box displays the word "Modified" until you save the application. Any time you make changes to your application, this word reappears in the application status box until you save the application again or until the Autosave option saves the application again.

Notice also that there is a tab at the bottom of the design pad with the word **Main**. This is because you selected the Tabbed Interface option in the Preference Editor (see "Setting Global Preferences" above). The tab display indicates that there is only one call flow — the default [Main] call flow — currently defined for the application. When other call flows are defined for an application, the name of each appears on its own tab.

The design pad (Figure 9) is where you do most of the work in creating call flows in Voice@Work. To create call flows, drag-and-drop nodes from the Nodes Palette onto this design pad. For more detailed instructions on dragging-and-dropping nodes onto the design pad see "Opening the Node Inspector Window" below.

NOTE:
Be aware that, because of the way applications compile and run on the target system, your applications must always be designed to start from the first node of the [Main] call flow.

Opening Other Windows

Once you open an application, you can open and customize the appearance of other windows you will use in creating this and other applications. Start by opening the Globals Manager window.

Opening the Globals Manager Window

To open the Globals Manager window, select **Globals** from the View menu. Voice@Work displays the Globals Manager window to the left of the main window (Figure 10).

NOTE:
As with most windows and tools in Voice@Work, there is more than one way to open the Globals Manager. Later chapters present all options for opening windows and tools, but this chapter presents only one for each action you are asked to perform.

To switch from one window to the other, click inside the window you want to be the active window.

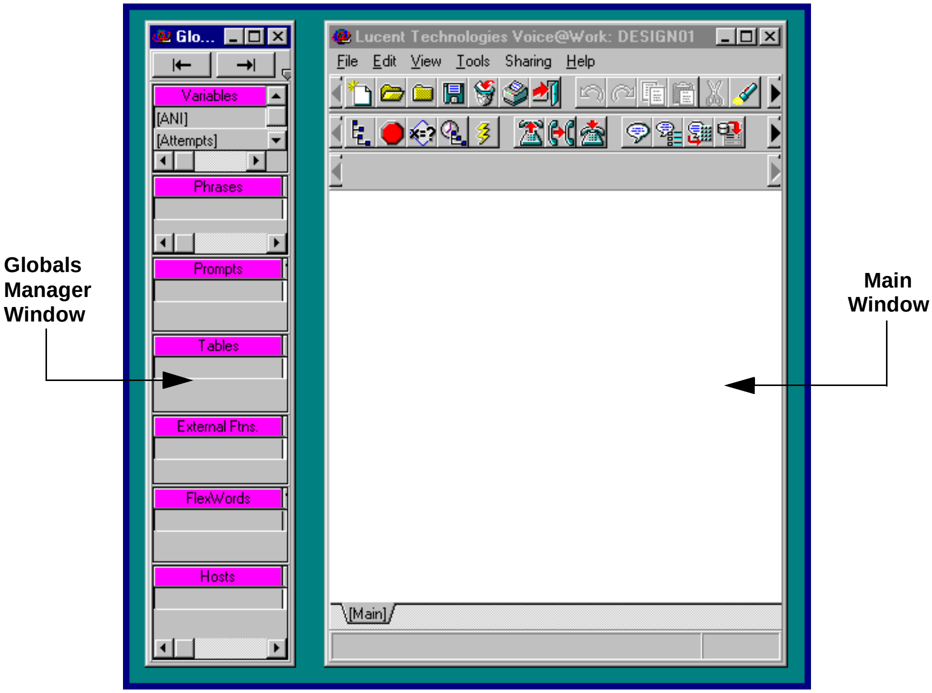


Figure 10. Main Window and Globals Manager Window Open

The Globals Manager is covered in detail in Chapter 13, "The Globals Manager", but there are a few aspects of it to note here:

- You can expand or collapse your view of the Globals Manager by clicking on the **Show More**  or **Show Less**  buttons near the top of the window.
- Notice that the Globals Manager has seven magenta-colored fields. These are the *name bars* for the seven global resource managers, each of which can be displayed in its own separate window by undocking it from the Globals Manager.
- You can resize the display space of each global resource manager by clicking and dragging its name bar up or down within the Globals Manager.
- You can undock a global resource manager by dragging the resource manager's *name bar* away from the Globals Manager. As you drag the resource manager away from the Globals Manager, a gray box with a solid black outline appears. Release the button wherever you want to place the resource manager. You can now treat the resource manager as a separate entity. You can open, close, move, and resize it, regardless of the status of the Globals Manager (Figure 11).

NOTE:
Once you undock a global resource manager from the Globals Manager, it remains undocked until you redock it. While it is undocked, it functions completely independently of the Globals Manager. It does not, for example, close or open automatically when you close or open the Globals Manager. If you close it, you must use the appropriate View menu entry or Toolbar icon to reopen it.

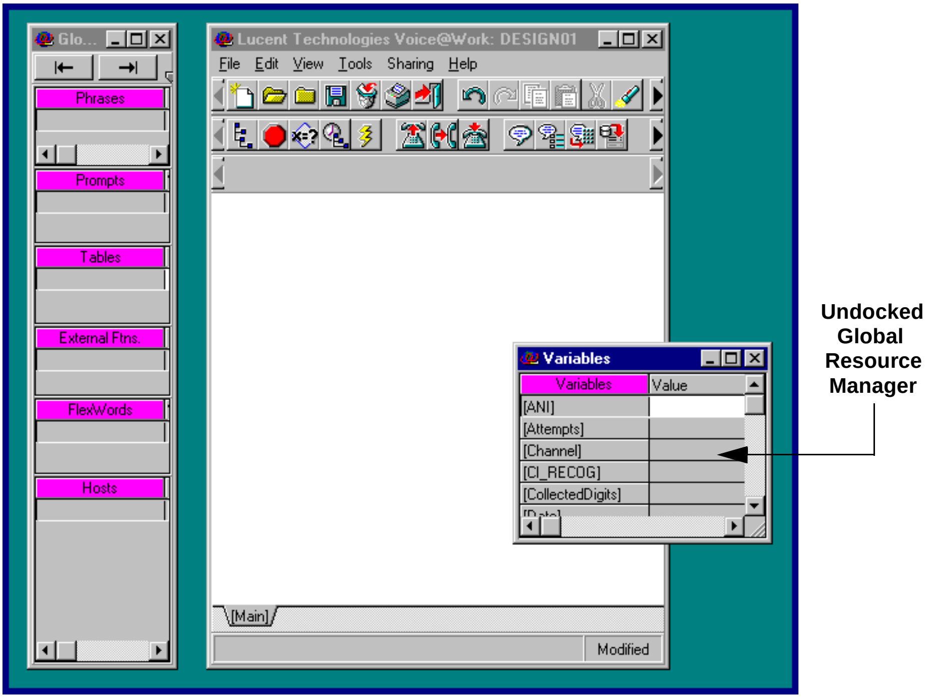


Figure 11. Example of Undocked Global Resource Manager

For more information about docking and undocking global resource managers, see "Undocking and Docking Resource Managers" in Chapter 13, "The Globals Manager".

Opening the Node Inspector Window

Another tool you will be working with extensively in Voice@Work is the Node Inspector. The Node Inspector window opens whenever you drag a new node onto the design pad or when you select **Node Inspector** from the Tools menu.

For this example, open the Node Inspector by dragging a node onto the design pad, using the following procedure:

1. Locate the **Answer Call** node icon  on the Node Palette (see Figure 4).
2. Drag-and-drop the **Answer Call** node icon into the design pad area.

   Voice@Work places the node in the design pad and displays the node naming dialog box (Figure 12). Notice that Voice@Work suggests a name for the node, because the **Automatically name nodes** checkbox was checked in the Preference Editor (see "Setting Global Preferences" above).

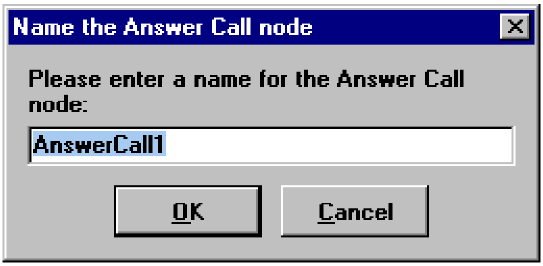


Figure 12. Node Naming Dialog Box

   At this point, you can override or accept the suggested name for the node. For now, accept the suggested name.

3. Click **OK**.

   Voice@Work displays the Node Inspector in its own nonmodal window (Figure 13). For more details on understanding and using the Node Inspector, see Chapter 26, "The Node Inspector".

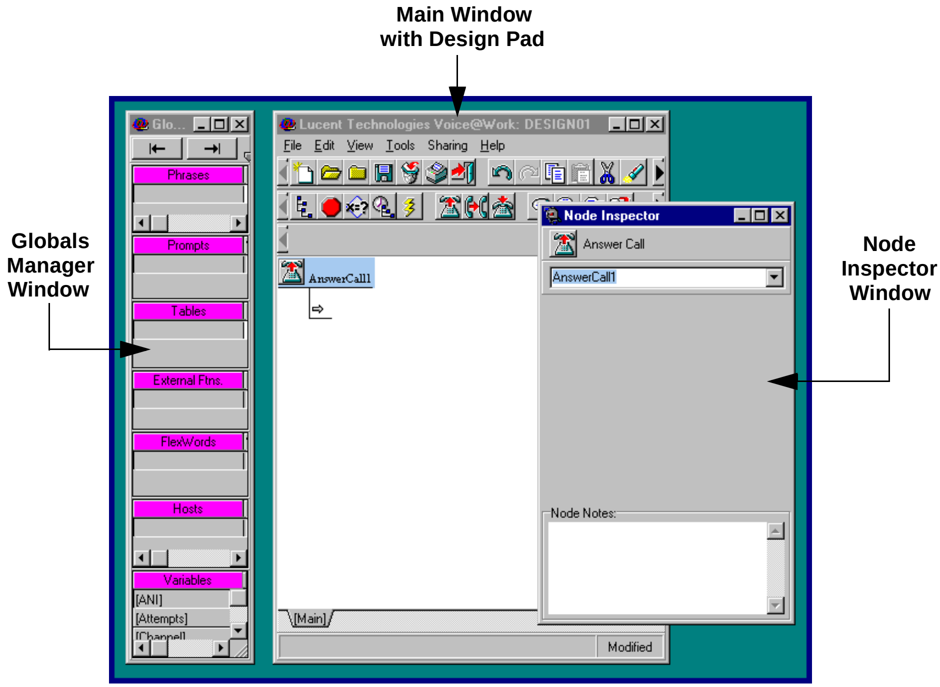


Figure 13. Multiple Windows Open

Configuring the Palettes

Another major feature of Voice@Work is the ability to configure each of the palettes — the Toolbar, the Node Palette, and the External Functions Palette (see Figure 4) — to maximize your effectiveness with Voice@Work. This section introduces you to a few of the configuration options for these palettes.

Selecting Viewing Options

Each of the palettes offers three viewing options, each of which you can select by right-clicking on any blank area of the palette:

NOTE:
Another of Voice@Work's powerful features is the extensive use of right-click "popup" menus. Most windows, nodes, and tools make use of them.

- The **Allow Scrolling** option, which is the initial default
- The **View All** option, which allows you to view all the icons for a given palette at once

  NOTE:
  To select this option, you must deselect both the **Allow Scrolling** and the **Use Tabs** options.

- The **Use Tabs** option, which groups icons and allows you to view each group by clicking its tab at the bottom of the palette

Figure 14 shows the three view options for the Toolbar.

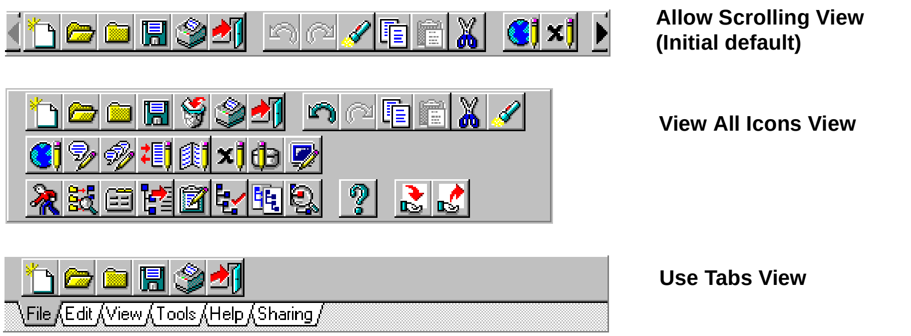


Figure 14. Three Views of the Toolbar

To see the three views of the Toolbar, do the following:

1. Right-click on the Toolbar anywhere other than on an icon, and uncheck the **Allow Scrolling** option from the popup menu. Notice that the Toolbar has expanded so that you can now see all its icons.
2. Right-click again on a blank area of the Toolbar, and check the **Use Tabs** option. Notice that the Toolbar again changes, this time showing each group, or category, of icons on a separate tab. To view categories of icons, click on the tabs.

Other Palette Options

Within each palette, you can also choose to show or hide each of the individual icons and the icon categories by selecting **Configure...** from the right-click popup menu for a palette.

As you can with the global resource managers, you can undock each of the palettes and view it in its own nonmodal window (see “Using Multiple Windows” above). Like other windows, these windows can then be opened, closed, moved, or resized independently of other Voice@Work windows.

For more information about configuring the palettes, see Chapter 12, “Palettes”.

Where to Go from Here

This chapter introduced you to some of the ways you can customize your workspace to work more effectively in Voice@Work. To see many of these examples demonstrated in an animated form, select **Demonstrations** on your Voice@Work CD-ROM disk.

This book provides two more resources to help you get started in developing your own Voice@Work applications:

- To see a procedural model for developing your own applications in Voice@Work, see Chapter 3, “Creating an Application Using Voice@Work”.
- For instructions on building a simple application step-by-step, see Appendix G, “Building a Sample Application”.

Before you go on, spend some time experimenting with the options presented in this chapter, to find the configuration that works most effectively for you with this powerful tool.

Creating an Application Using Voice@Work

3

Overview

Creating an application in Voice@Work is a five-step process:

1. Plan the application design.
2. Create the call flows and other application components.
3. Generate the source file code.
4. Transfer the application to the target system.
5. Install the application on the target system.

Purpose

The purpose of this chapter is to provide you with a model plan for creating an application using Voice@Work. Topics include:

- Plan the Application Design
- Create the Call Flows and Other Application Components
- Generating the Code
- Transferring the Application to the Target System
- Installing the Application on the Target System

NOTE:
Each of the topics discussed in this chapter are detailed further in later in the book. This chapter is an overview for creating application using Voice@Work.

Plan the Application Design

Before you create your application, you must have a clear idea of what you want to accomplish with it. This involves thorough planning to make sure you have thought of everything you want your application to do. You must plan for every contingency.

Creating a flow chart or diagram can be helpful when mapping out what you want your application to do. Writing an outline and a script is another helpful technique.

NOTE:
For more information about designing applications for INTUITY CONVERSANT systems, see the *INTUITY™ CONVERSANT® Application Design Guidelines* book for your system.

Create the Call Flows and Other Application Components

Once you have a clearly defined design, you can create the call flows and other components required to make the design work. Basically, you do this by dragging nodes onto the design pad (see Chapter 30, "Working with Nodes" for the procedure). You can set attributes for nodes and edit them using the Node Inspector tool.

Voice@Work Components

In addition to the nodes offered on the Node Palette, Voice@Work offers a variety of other components you can use in building a call flow. Table 2 describes the major components, called "global resources," offered in Voice@Work.

Table 2. Voice@Work Component Types

Type of Resource	Description/Comments
Phrases and phrase tables	Phrases are bits of spoken (or written) speech, often used to instruct customers or prompt them for pieces of information. Phrases are grouped together and contained in phrase tables. Phrase tables can be created within Voice@Work application or imported from other applications or from text files.

Continued on next page

Table 2. Voice@Work Component Types

Type of Resource	Description/Comments
Variables	Variables are application-specific information holders. Variables may be used to collect and hold data, act as counters, or make use of system performance information.
Prompts	Prompts are used to organize phrases, variables, and controls into logical units for use with the Interactive Voice Response (IVR) nodes. Prompts are used to instruct callers, to get callers to select choices, and to obtain information from callers.
Database Tables	Voice@Work applications can be designed to make use of database tables residing on the target system or on machines other than the target system. These tables be created and populated from within Voice@Work.
External functions	External functions are used when you need Voice@Work to do something that no currently existing node does.
Host definitions	Host definitions are used to specify an interface between the target system and a host computer remote to that machine.
FlexWord grammars	FlexWord grammars are based on a type of speech recognition that recognizes phonemes or parts of words in a specific language.
WholeWord grammars	WholeWord grammars are used to provide a highly accurate recognition of the digits 0–9, “oh”, “yes”, and “no”.

Voice@Work Tools

Voice@Work also offers a variety of development tools to help you edit, test, and debug your application. Table 3 describes the development tools offered in Voice@Work.

Table 3. Voice@Work Development Tools

Tool	Description/Comments
Application Resources	The Application Resources tool allows you to: ■ See at a glance what resources are in use with the current application ■ Import additional resources into your application as needed (as Private Copies or Shared) ■ View which resources have been shared in/out (Sharing tab)
Call Flow Inventory	The Call Flow Inventory tool allows you to see: ■ What call flows exist for your application. ■ Which call flows use other subordinate call flows (Uses tab) ■ Which call flows are used by or are subordinate to other call flows (Used By tab)
Code Generation / Application Transfer	The Code Generation tool generates the code necessary to transfer and install the application onto the target system. The Application Transfer tool transfers the application files to the target system so they can be installed.
Events String Editor	The Events String Editor displays both a list of all the variables defined for the application and a list of the variables selected to be treated as Call Data Events on the target system.
Node Inspector	The Node Inspector displays different information and editing options for each node, depending on the node's predetermined attributes. It allows you to edit node attributes easily and quickly.
Preference Editor	The Preference Editor allows you to set global preferences for using Voice@Work.
Simulation	The Simulation tool runs your application in a simulated environment. It is often helpful in finding problems within the application.

Table 3. Voice@Work Development Tools

Tool	Description/Comments
Verify Design	The Verify Design tool allows you to check your application design for possible errors or omissions. When you open the Verify Design tool, it automatically searches through each call flow of your application. The results are displayed in the Verification Results window.

Sharing Resources

Voice@Work also offers options for sharing resources with other developers. For details about sharing resources, see Chapter 10, "The Sharing Menu".

Generating the Code

Once you have created, tested, and debugged your application, you must convert it to a form that the target system can understand and use.

Use the Code Generation/Application Transfer tool to generate the code for the target system. For more information about generating the application source code, see Chapter 23, "The Code Generation Tool".

Transferring the Application to the Target System

When you have finished generating the code for the target system, you must transfer the generated source code files to the target system.

You have the option of transferring files using either a network or floppy disks.

Use the Code Generation/Application Transfer tool to transfer the files to the target system. For more information about transferring applications, see Chapter 24, "The Application Transfer Tool".

Installing the Application on the Target System

Before you can run the application you have transferred to the target system, you must install it on the target system. Do this from the target system, using the SCI tool. For more information about installing the application using the SCI tool, see Appendix A, ‘‘Using the SCI Tools’’.

Using the Online Help 4

Overview

The online Help system in Voice@Work is designed to make it as easy as possible for you to find and use the information you need when working with Voice@Work applications. Most of the information in this book is also included in the online Help.

The ability to link to related topics and to view both the online Help and the application you are working on in Voice@Work make the online Help in most cases superior to using printed documentation.

Purpose

The purpose of this chapter is to provide you with information about the online Help system and its features, including information on how you can get the most out of the online Help. Topics include:

- Purpose of the Online Help
- How the Online Help Is Organized
- Features of the Online Help
- Tips for Using the Online Help

Purpose of the Online Help

The purpose of the Voice@Work online Help system is to help you find information about using Voice@Work quickly and easily. It is designed to allow you to continue working in Voice@Work while accessing the help topic(s) needed.

How the Online Help Is Organized

In general, the online Help for Voice@Work is organized to reflect the architecture of Voice@Work itself. Major topics appear in main windows. Topics that provide additional information, including extra hints and tips for using Voice@Work more effectively, appear in Hints and Tips windows. Definitions and “sidebar” topics appear in “popup” windows (see “Appearance of the Online Help Screens” in “Features of the Online Help” below).

Main topics are typically organized with a brief overview of the topic followed by two or more subtopics. There is commonly a list of the subtopics near the beginning of the topic, with links to the particular subtopics.

Use the Contents tab to view the major areas of the online Help system and get a feel for the general organization.

Features of the Online Help

The Voice@Work online Help system has several features which should make it easier for you to get the most out of it. Among them are:

- Standard WinHelp features (such as links to topics and popup windows and navigational buttons)
- The use of bold type
- Interactive graphics
- “Extra Help” buttons
- Appearance of the online Help screens (color options)

Standard WinHelp Features

WinHelp systems typically offer two basic types of links in text, characterized by different colored text, usually green:

- Jumps – Jumps link to other topics. Jumps are indicated by a solid underline.
- Popups – Popups bring up a popup window that disappears when you click again. Popups are indicated by a broken underline.

WinHelp systems also offer a variety of standard navigational buttons. For details, see your Windows documentation.

The Use of Bold Type

The Voice@Work online Help system uses bold type for three purposes:

- To indicate another topic in Help where you can get more information
- To indicate menu, option, or window title names
- To emphasize important key words or concepts

Interactive Graphics

Nearly all of the graphics displays in Voice@Work are interactive. This means that they have "hotspots" embedded within them, and you can click on these hotspots to get more information about the selected item in the graphic.

When the cursor changes to a hand icon  , it designates a “hotspot” where you can click to get more information.

These are especially useful in the many screen captures displayed in the Voice@Work online Help system.

“Extra Help” Buttons

The online Help for Voice@Work uses three types of “extra help” buttons for those times when you need more information than the current topic offers:

See Also buttons offer ideas for using the current topic more effectively or provide extra (usually background) information.

How? buttons link to other Help topics that explain how to perform some procedure mentioned in the current topic.

Hints and Tips buttons bring up a “Topics Found” window that provides links to other related topics.

Appearance of the Online Help Screens

Voice@Work and its online Help is optimized to run on a system using 256 colors (8-bit) on a 1024 x 768 pixel display.

The online Help system is designed to display topic titles as white letters on a blue background. If your display does not look like it should, do one of the following:

- In the main Help window, check the **Options** menu. If the **Use System Colors** option is checked, uncheck it.
- Use the Control Panel to change the Display Appearance. (The problem is a bug with WinHelp – some desktop Display Appearance options do not work correctly.) For help in changing the Display Appearance of your system, see your Windows documentation.

Below are descriptions and samples of each of the types of windows used in this Help system.

- Main topic windows – Blue title banner with white letters and white background for topic text (Figure 15).

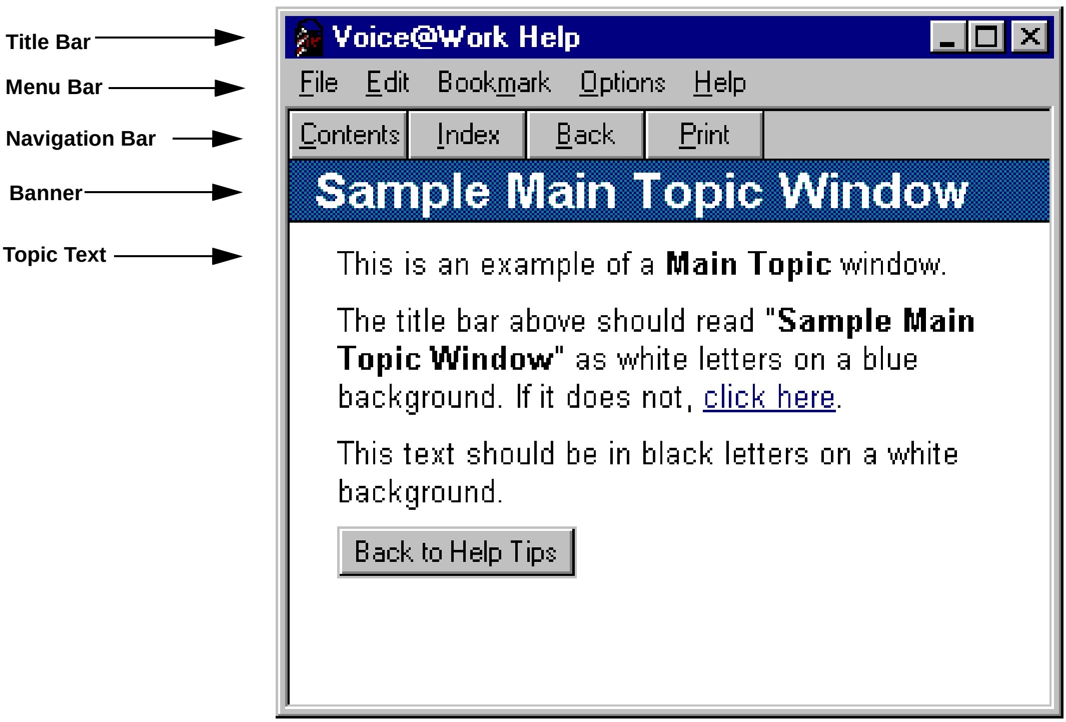


Figure 15. Sample Main Topic Window

- Hints and Tips windows – Blue title banner with white letters and yellow background for topic text (Figure 16).

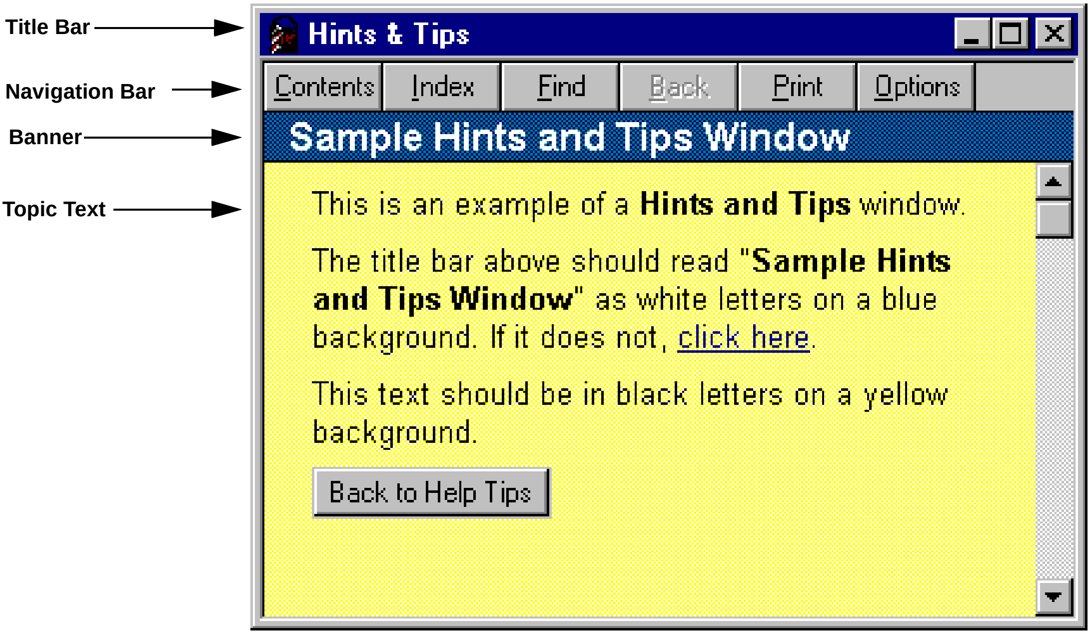


Figure 16. Sample Hints and Tips Window

- Popup topic windows – Yellow background for topic text. When titles are used, they appear in blue letters (Figure 17).

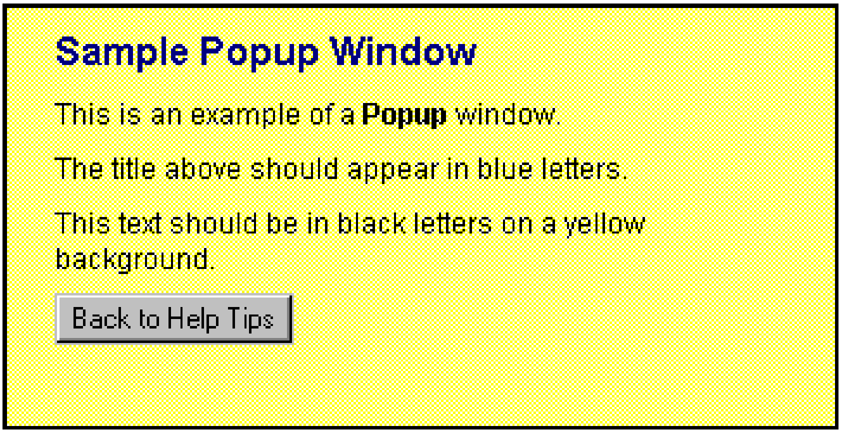


Figure 17. Sample Popup Window

Tips for Using the Online Help

Following are some tips that can aid you in using the online help more effectively. These tips include:

- Performing index and full text word searches
- Printing out selected topics
- Finding previously viewed topics
- Defining and using bookmarks
- Writing your own annotations for topics
- Using the carefully placed, movable, and resizable windows

Performing Index and Full Text Word Searches

Each Voice@Work online help window provides an **Index** button to allow you to access the index and full text word search (Find) tools. Many people are already familiar with the online index capabilities of WinHelp, which function much like their paper counterparts, with the exception that double-clicking an index entry automatically causes the help system to display that topic.

In many cases, the full text word search is even more helpful than the index. The full text word search tool is accessed by clicking the **Find** tab. This tool allows you to type in a word or a phrase, and then it automatically displays a list of all the topics in the online help that contain all the words you typed in. You can add more words to narrow the search, or you can select a topic from those displayed. This option can be especially helpful when you can remember a phrase or a key word in a topic you have previously looked at, but you cannot remember the topic title.

Printing Topics

At times, it can be helpful to print out one or more topics rather than trying to work with them online. WinHelp allows you to print selected topics, but only one at a time. The topic printed is the selected topic.

To print out a topic, simply open the topic you want and then select the **Print** button from the toolbar.

NOTE:
When printing out topics from the online help, you must adjust your printer's grayscale settings to print the graphics properly.

Finding Previously Viewed Topics

There are times when you may want to return to a topic you have looked at earlier, to refresh your memory or to get more information from that topic. The online help system offers two ways of doing this:

- Using the **Back** button on the help window toolbar
- Selecting a topic from the **History Window**, which allows you to return to any topic you have already viewed since the online help was opened

NOTE:
The **History Window** shows the history only for the type of window you are in. If, for example, you want to view a topic in a Hints & Tips window, you must open the **History Window** for the Hints & Tips window type. You cannot see the history of Main Topic windows from a Hints & Tips window.

To select a topic from the **History Window**, do the following from any online help window:

1. From the **Options** menu, select the **Display History Window...** option.

   The system displays the **Windows Help History** window (Figure 18).

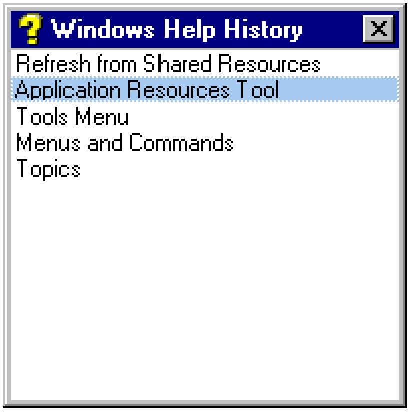


Figure 18. Windows Help History Window

2. Double-click the topic you want to view again.

   The system displays the selected topic.

Adding and Using Bookmarks

You can use bookmarks to mark topics that you know you want to return to or that you find yourself returning to often. Creating a set of bookmarks allows you to return to any topic at will.

NOTE:
This option is available only for Main Window topics.

To create a bookmark for a topic, do the following:

1. Open the topic you want to create the bookmark for.
2. From the **Bookmark** menu, select **Define...**

   The system displays the **Bookmark Define** window (Figure 19).

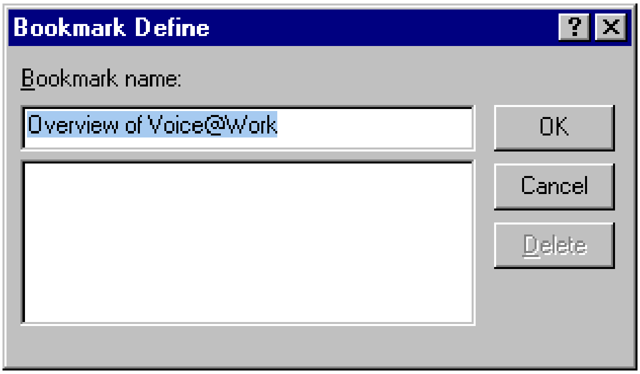


Figure 19. Bookmark Define Window

3. To give the bookmark a name different from the suggested topic name, type in the new name.
4. Click OK.

To use a bookmark after you have defined it, do the following:

1. From any Main Window topic, click the **Bookmark** menu.
2. From the dropdown menu, click the name of the bookmark you want.

   The system displays the selected topic.

Writing and Using Annotations

Annotations can be used much like electronic "post-it" notes to make notes for yourself about things you want to remember about selected topics. They are saved and kept with their selected topics until you delete them.

To write an annotation for a topic, do the following:

1. Open the topic you want to annotate.
2. Select the appropriate option from the following:
   - If you are in a Main Window topic, click the **Edit** menu.
   - If you are in a Hints & Tips topic, click the **Options** button.
3. Select the **Annotate...** option.

   The system displays the **Annotate** window (see graphic below).

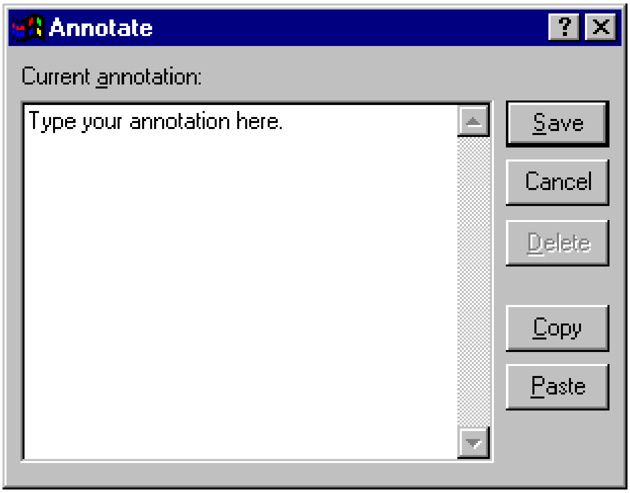


Figure 20. Annotate Window

4. Type your annotation in the text field.
5. Click **Save**.

   The system saves the annotation, attaches it to the topic, and places a paper clip icon at the beginning of the topic to indicate that the annotation exists.

To view an annotation after you have created it, click on the paper clip icon.

Using the Carefully Placed, Movable, and Resizable Windows

Voice@Work has been designed to allow you to open a help window on the right side of your screen and leave it open while you work in Voice@Work on the left side of your screen. This allows you to easily switch back and forth between help and your application.

You can, however, if you want, move and resize help windows to whatever configuration allows you to work most effectively. Moving and resizing of help windows works according to standard Windows procedures.

By default, Voice@Work online help — like most WinHelp systems — allows the help windows to move behind other windows when you make those other windows the active windows. In most cases, you can bring Voice@Work help to the top again by clicking in the help window (if it is visible) or by using ALT TAB to select the Voice@Work help window.

You can also set your help windows to always stay in view. To set the Voice@Work help windows to remain on top of any other window at all times, do the following:

1. Open the Voice@Work online help.
2. Click the **Options** menu on the menu bar.
3. From the **Keep Help on Top** submenu, select **On Top**.

The File Menu 5

Overview

The commands on the File menu deal with operations to applications.

In Voice@Work, only one application can be open at a time. Thus, where a command involves opening a new or existing application, Voice@Work automatically closes any application currently open. If there are any unsaved changes in that application, Voice@Work first asks whether you want to save changes before closing.

Purpose

The purpose of this chapter is to provide a summary of the File menu commands and their operations. Topics include:

- Opening the File Menu
- Summary of Commands
- The Print Reports Window
  - Print Options
  - Destination Options

Opening the File Menu

To open the File menu, do one of the following:

- Click **File** on the main window menu bar.
- Press ALT F.

Summary of Commands

File menu commands deal with various file operations, similar to other Windows-based applications.

Table 4 describes the commands found on the File menu.

Table 4. File Menu Options

Command	Keyboard Shortcut	Description/Action
New		Creates a new application.
Open...	CONTROL O	Opens an existing application. **NOTE:** If you are attempting to open or save an application on another (networked) computer, you must first make sure the appropriate read/write permissions are set for the application you want to use.
Delete		Deletes the current application.
Close		Closes the current application. If you made any changes to the application since the last Save action, Voice@Work asks whether you want to save the application before closing it.

Continued on next page

Table 4. File Menu Options — *Continued*

Command	Keyboard Shortcut	Description/Action
Save	CONTROL S	Saves the current application. NOTE: If you are attempting to open or save an application on another (networked) computer, you must first make sure the appropriate read/write permissions are set for the application you want to use.
Save As...		Saves the application with a different name.
Print...		Opens the **Print Reports** window (Figure 21), which allows you to select which reports you want to print; see "The Print Reports Window" below.
(Recent Applications)		Lists recently opened applications and can be used as shortcuts to reopen them.
Exit		Closes any open applications and quits Voice@Work. If you made any changes to the application since the last Save action, Voice@Work asks whether or not you want to save the application before closing it.

The Print Reports Window

The Print Reports window opens when you select the **Print...** option from the File menu. This window allows you to select the types of items you want to print. Each type of item generates a separate printout, which you can send directly to a printer or to a file. The date, the name of the application, type of item, and the page number appear in the header of the print output.

NOTE:
If you send the output to a file, Voice@Work saves it in *.doc format. You can print the file later using any program that recognizes and uses the *.doc format.

Print Options

The Print Options box allows you to select which reports to print. When the Print Reports window opens, the Print Options box displays all print options available, based upon the contents of the current application. Initially, all displayed options are checked, unless you have reset the Printing tab options in the Preference Editor. See Chapter 27, "The Preference Editor" for information on resetting the Printing tab options. You must uncheck any option for which you do *not* want to receive a printout.

NOTE:
Only those options for which items are actually available appear in the Print Options box. For example, if the Code Generation tool has not yet been used and no code has yet been generated, Code Generation does not appear as an option in this box.

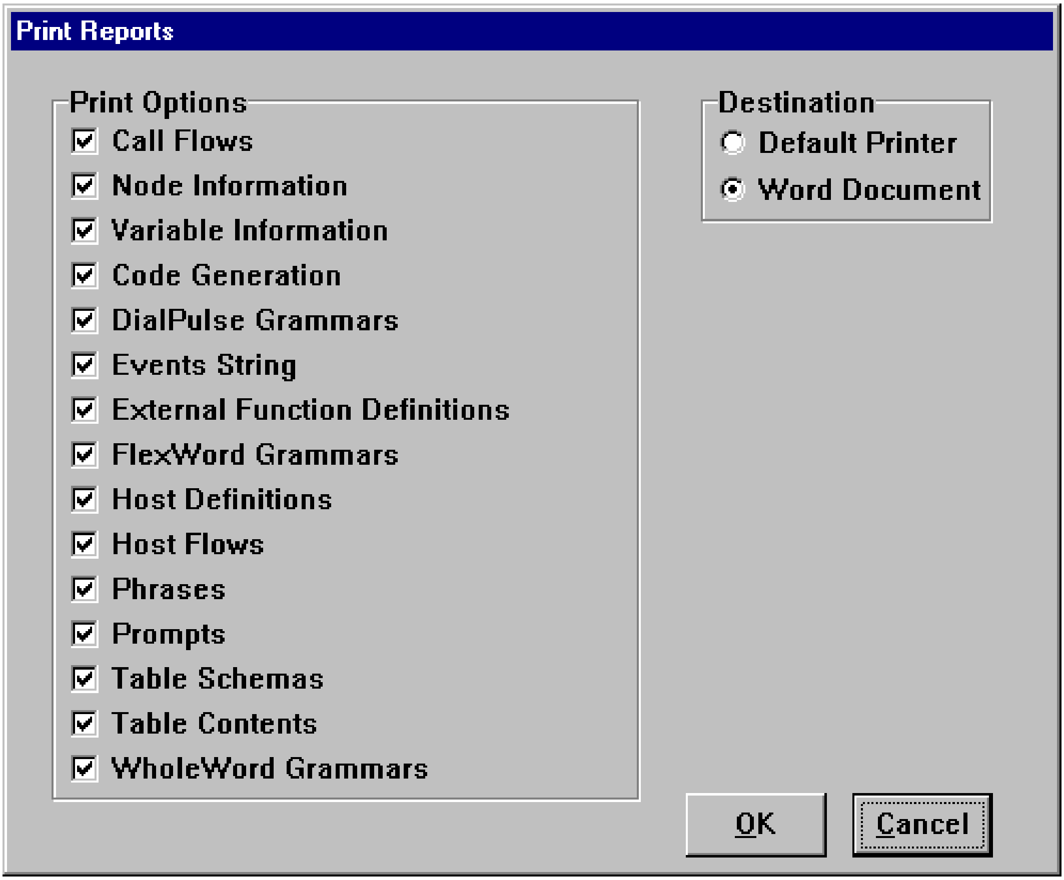


Figure 21. Print Reports Window

Table 5 describes the possible print options. Each type of item generates a separate printout. The date, the name of the application and type of item, and the page number appear in the header of each report.

If the print Destination option is set to Word Document,

Table 5. Print Options

Option	Name of Print File (If Saved as Word Document)	Print Output
Call Flows	flows.doc	Produces a printout of each call flow similar to the design pad display of the call flow.
Node Information	nodes.doc	Prints a report describing all nodes in the application, arranged by call flow, in the order in which they appear. Information in this report includes the following: ■ Node name ■ Type of node ■ Where the node is entered from (the parent node) ■ Node attributes (as they appear in the Node Inspector) ■ Any phrases played by the node ■ Information about where the caller is routed after leaving this node
Variable Information	Variable.doc	Summarizes the attributes of each variable in the application. The information included about each variable is the same as that found in the Variable Manager

Continued on next page

Table 5. Print Options — *Continued*

Option	Name of Print File (If Saved as Word Document)	Print Output
Code Generation	codegen.doc	Reports statistics about the code generated using the Code Generation tool. The printout includes information about the nodes, features, grammars, and database table provisioning used by the application.
DialPulse Grammars	DialPulse.doc	Lists and describes all Dial Pulse Recognition grammars available for the application.
Events String	events.doc	Summarizes the call data events defined in the Events String Editor. This report includes the names and lengths of all variables defined as call data events for the current application.
External Function Definitions	extftndf.doc	Lists information about the External Functions used in the application. This table includes the name, the description, the return description, and system information.
FlexWord Grammars	flexword.doc	Lists the names of all FlexWord grammars, along with the FlexWord lists.
Host Definitions	host.doc	Lists all hosts defined in the application, along with a list of defined parameters for each host. **NOTE:** If the Host Flows checkbox is also checked, Host Definitions and Host Flows are placed in the same file and printed out as part of the same report.

Continued on next page

Table 5. Print Options — *Continued*

Option	Name of Print File (If Saved as Word Document)	Print Output
Host Flows	host.doc	Produces a printout of each host flow similar to the Design Pad display of the host flow. **NOTE:** If the Host Definitions checkbox is also checked, Host Definitions and Host Flows are placed in the same file and printed out as part of the same report.
Phrases	Phrase.doc	Lists all phrases in the application in numerical order by phrase numbers. The text for each phrase follows the phrase numbers.
Prompts	Prompt.doc	Lists all prompts, in alphabetic order by prompt name. The text for each prompt follows the prompt name.

Continued on next page

Table 5. Print Options — *Continued*

Option	Name of Print File (If Saved as Word Document)	Print Output
Table Schemas	Table.doc	Lists the database table name, description, field names, classes, formats, length, and keys for each database table in the application. **NOTE:** If the Table Contents checkbox is also checked, Table Schemas and Table Contents are placed in the same file and printed out as part of the same report.
Table Contents	Table.doc	Lists the contents of each database table used in the application. If there is no data, prints a message to that effect. **NOTE:** If the Table Schemas checkbox is also checked, Table Schemas and Table Contents are placed in the same file and printed out as part of the same report.
WholeWord Grammars	Wholeword.doc	Lists and describes all WholeWord speech recognition grammars available for the application.

Destination Options

The Destination box options determine where the output of the reports will be sent. Table 6 describes the Destination options.

Table 6. Print Destination Options

Option	Description/Comments
Default Printer	Sends the output to the default printer for your system so it can be printed.
Word Document	Saves the output as a *.doc file, which allows you to edit it using a word processor before printing it. The output is saved to the print subdirectory of the application. For example, if your application were named **Test_app**, print output files would be saved to the following directory path: **C:\Program Files\Voice@Work\Apps/Test_ app\print*.doc** **NOTE:** The save path shown above assumes you have used the default installation paths for Voice@Work. If this is not the case, you must revise the path accordingly.

The Edit Menu

6

Overview

The commands on the Edit menu are concerned with basic editing operations performed on applications. Other commands deal with configuring the Toolbar, the Node Palette, and the External Functions Palette.

Purpose

The purpose of this chapter is to provide information about the commands on the Edit menu. Topics include:

- Opening the Edit Menu
- Summary of Commands
- More about Cut/Copy/Paste
- The Find... Feature

Opening the Edit Menu

To open the Edit menu, do one of the following:

- Click **Edit** on the main window menu bar.
- Press ALT E.

Summary of Commands

The Edit menu commands allow you to:

- Perform a variety of editing functions on your application
- Configure any of the three palettes (see Chapter 12, "Palettes")

Table 7 lists the Edit menu commands.

Table 7. Edit Menu Commands

Command	Keyboard Shortcut/Icon	Description/Action
Undo	CONTROL Z 	"Undoes" the last action or series of actions. Selecting **Undo** from the Edit Menu or pressing the Undo icon on the Toolbar allows you to undo (reverse) the last action or series of actions. The number of "undo levels" is limited only by the number of undoable actions performed since the last Save action. Actions are undone in reverse of the order in which they were done. **NOTE:** Some actions cannot be undone. If you place and leave the cursor over the Undo icon, a popup message appears telling you what action will be undone. The same message also appears in the status bar.

Continued on next page

Table 7. Edit Menu Commands — *Continued*

Command	Keyboard Shortcut/Icon	Description/Action
Redo	CONTROL Y 	"Redoes" any actions that were undone using the **Undo** command. Selecting **Redo** from the **Edit** menu or pressing the Redo icon on the Toolbar allows you to do redo the last action or series of actions that were undone using the Undo feature. Redo is limited only by the number of Undo actions that have been performed. Actions are redone in reverse of the order in which they were undone (or, in other words, in the same order they were originally done). Once you perform another action after performing an Undo or a Redo action, Redo is no longer active or available. If you place and leave the cursor over the Redo icon, a popup message appears telling you what action will be redone. The same message also appears in the status bar.
Cut	CONTROL X 	Copies all node attributes and application resources (including encapsulated call flows) necessary to make the selected node or nodes fully functional when it is pasted into a new location. It removes only the node itself, and not the associated resources, from the original application. When you cut a "single" node, that node and all subordinate nodes attached to it are cut. To cut a single node without cutting its subordinate nodes, you must first disconnect the subordinate nodes, then cut the desired node. For more information about the **Cut** command, see "More about Cut/Copy/Paste" in this chapter.

Continued on next page

Table 7. Edit Menu Commands — *Continued*

Command	Keyboard Shortcut/Icon	Description/Action
Copy	CONTROL C 	Copies all node attributes and application resources (including encapsulated call flows) necessary to make the selected node or nodes fully functional when pasted into a new location. When you copy a “single” node, that node and all subordinate nodes attached to it are copied. To copy a single node without copying its subordinate nodes, you must first disconnect the subordinate nodes, then copy the desired node. For more information about the **Copy** command, see “More about Cut/Copy/Paste” in this chapter.
Paste	CONTROL V 	Copies all nodes currently on the clipboard to a selected open branch, or set of open branches in the case of multiple-node paste operations For more information about the **Paste** command, see “More about Cut/Copy/Paste” in this chapter.
Configure Toolbar...	**Configure...** option on right-click menu	Opens the Configure Toolbar window which allows you to customize the Toolbar For more information about configuring the Toolbar, see Chapter 12, “Palettes”.
Configure Node Palette...	**Configure...** option on right-click menu	Opens the Configure Node Palette window which allows you to customize the Toolbar For more information about configuring the Node Palette, see Chapter 12, “Palettes”.
Configure External Function Palette...	**Configure...** option on right-click menu	Opens the Configure External Functions Palette window which allows you to customize the Toolbar For more information about configuring the External Functions Palette, see Chapter 12, “Palettes”.

Continued on next page

Table 7. Edit Menu Commands — *Continued*

Command	Keyboard Shortcut/Icon	Description/Action
Find...	CONTROL F 	Searches across all call flows in the application to find: ▪ Text strings used in component names ▪ Locations where specified items are used ▪ Unused items ▪ Text strings For more information about the Find feature, see "The Find... Feature" in this chapter.

More about Cut/Copy/Paste

The following sections provide rules and guidelines for using the cut/copy/paste features in Voice@Work.

General Rules

NOTE:
When you cut or copy a "single" node, that node and all subordinate nodes attached to it are cut or copied. To cut a single node without cutting its subordinate nodes, you must first disconnect the subordinate nodes, then cut the desired node.

If an open branch is not available or if you attempt to paste the nodes at a location that is not an open branch, the nodes are placed at the bottom, disconnected from the existing call flow. This is in contrast with many other Windows applications that allow you to paste a selection over another highlighted selection, thus replacing that selection.

Voice@Work does not allow you to paste nodes from one application into a different application if the selected language for the two applications is different.

The Paste action can be undone only if it is being done within a single application. If the Paste action is being done between two applications, it cannot be undone.

When attempting to paste, Voice@Work identifies and notifies you of any unresolved issues before either completing or blocking the paste operation. If Voice@Work is able to complete the paste action but finds an error involving a required attribute for a node, it places a message in the **Node Notes** field for the affected node.

Cut/Copy/Paste operations can be used both within an application or between different applications. When using these operations between applications, you must use the following procedure:

1. Cut or copy the desired nodes in the original application.
2. Save and close the original application.
3. Open the application you want to paste the desired nodes into.
4. Paste the nodes where you want them.

If a resource does not exist in the target application, Voice@Work creates or imports it during the paste operation. If one or more resources do exist, Voice@Work offers the option to overwrite them. If you elect to continue, *all* existing resources are overwritten. If you elect to cancel the overwrite action, the paste action is completed without overwriting existing resources.

During cut/copy/paste operations, all selected nodes are recreated in their entirety, including attributes and resources, in the target applications. Attributes include:

- Variables
- Phrases
- Prompts
- Database tables (includes both table schemas and table data)
- Host definitions
- WholeWord grammars
- FlexWord lists
- External functions
- Encapsulated call flows

Name Conflicts During Paste Actions

Node names must be unique within a call flow. When you attempt to paste selected nodes, Voice@Work checks for node name conflicts. If a node name matches one in the call flow where you are attempting to paste it, Voice@Work prompts you for a new name. If you select Cancel, the paste action is aborted.

Call Flow nodes are an exception to this name rule. A Call Flow node name is always appended with a number in parentheses indicating how many times it has been used in each call flow. Thus, when a Call Flow node is copied, the number is changed, but the name remains the same. For example, if a call flow containing a Call Flow node named **My_flow(1)** is copied to a different place in the same call flow, the new name is **My_flow(2)**.

NOTE:
The attributes of the copied nodes remain unchanged.

The paste operation uses the resource pathnames in pasting the new application. Therefore, if you are attempting to paste nodes with shared resources, use the Refresh From Shared Resources tool before pasting. This allows you to make sure that all resource pathnames are current before pasting.

NOTE:
Keep in mind during paste operations that Voice@Work allows you to overwrite a private copy of a shared resource by a shared version, or to overwrite a shared version with a private copy.

If the target application already has a resource with the same name as one in the nodes you are trying to paste, Voice@Work prompts you to specify whether you want to keep the original resource on the target system or overwrite it with the new one.

Selecting Multiple Nodes for Cut/Copy/Paste

Voice@Work also allows you to cut/copy/paste multiple nodes within a call flow, but they must be contiguous nodes. That is, they must have the same “parent” node and be adjacent to each other on the call flow. As with single nodes, all subordinate nodes are copied or moved with the parent nodes (Figure 22).

NOTE:
You cannot, for example, select only the Set and Test nodes **Move_Balance_to_Currency_Field** and **Bump_Unrecognized_Screen** in Figure 22 without also selecting the **Bump_Host_Timeout** node between them.

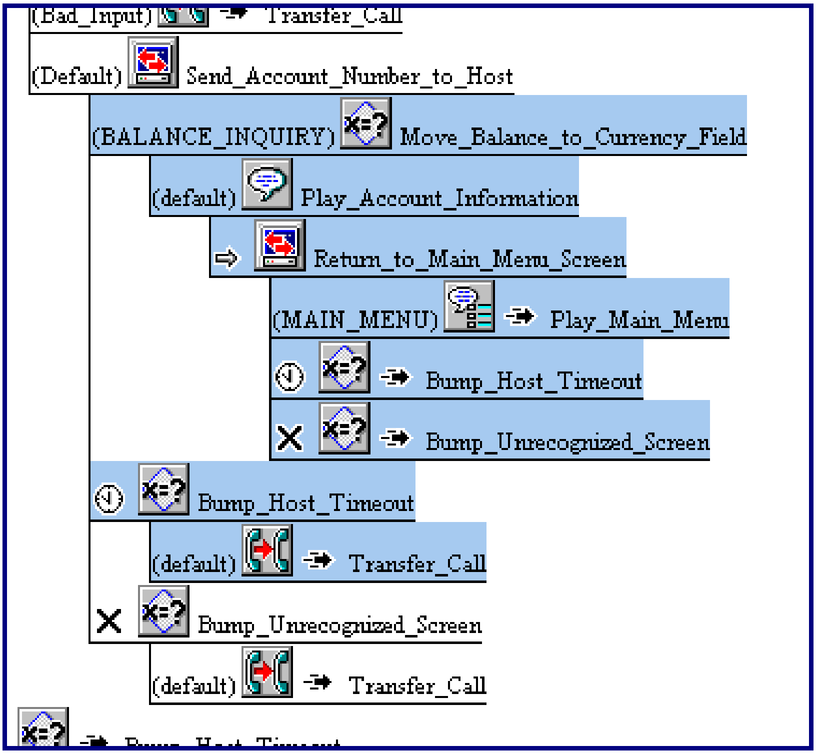


Figure 22. Multiple Contiguous Node Selection

To select multiple nodes for cut/copy/paste actions, do the following:

1. Click the first node you want to include in the action (Set and Test node **Move_Balance_to_Currency_Field** in Figure 22).
2. Shift-click the last node *at the same level* that you want to include in the action (Set and Test node **Bump_Host_Timeout** in Figure 22).

NOTE:
If you try to shift-click a node at a different level, only that node is selected. You cannot, for example, shift-click the **Transfer_Call** node to select the nodes highlighted in Figure 22.

Using Drag-and-Drop to Cut/Copy/Paste

Within a single call flow, you can use drag-and-drop techniques to cut/copy/paste nodes in Voice@Work. If you are in Multiple Document Interface mode (see Chapter 27, "The Preference Editor"), you *cannot* drag and drop nodes across call flows.

When dragging and dropping nodes within a call flow, the appearance of the cursor changes to let you know whether or not you can paste the nodes at the cursor's location. When you are at a legal drop location, the cursor appears as an arrow with a document attached (Figure 23).

Figure 23. Cursor Appearance for Legal Drop

When you are at an illegal drop location, the cursor appears as a circle with a slash mark (Figure 24). If you attempt to drop the selected nodes when the cursor has this appearance, the cut/copy/paste action is aborted and the nodes return to their original locations. This rule holds true for both single and multiple node cut/copy/paste actions.

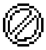


Figure 24. Cursor Appearance for Illegal Drop

To perform a cut/copy/paste action on multiple contiguous nodes, you must first make sure you have enough open branches to accommodate the paste action where you want the nodes to go.

NOTE:
If you do not have enough open branches—starting with the target branch and counting those immediately below it—to accommodate the number of nodes you want to paste, and you drop them anyway, the nodes are disconnected and pasted at the end of the call flow.

Select the nodes you want to cut/copy/paste (see "Selecting Multiple Nodes for Cut/Copy/Paste" above), then hold the SHIFT key down while you drag the selected nodes to the desired location.

The Find... Feature

Voice@Work has powerful and versatile search-and-find capabilities. You can use the Find... function on selected items or types of items to locate:

- Text strings
- Text strings in names
- Where items are being used
- Unused items

Accessing Find...

To use the Find... function:

1. Click the **Edit** menu, press ALT E, or click  on the Toolbar.
2. Click the **Find...** menu choice or press **F**.
3. Click the tab for the search you want to conduct. Tab choices include:
   - **Named** (ALT N)
   - **Uses Of** (ALT O)
   - **Unused** (ALT U)
   - **Find Text** (ALT F)

The Named Tab

The Named search option (Figure 25) searches for text strings only in the names of selected items.

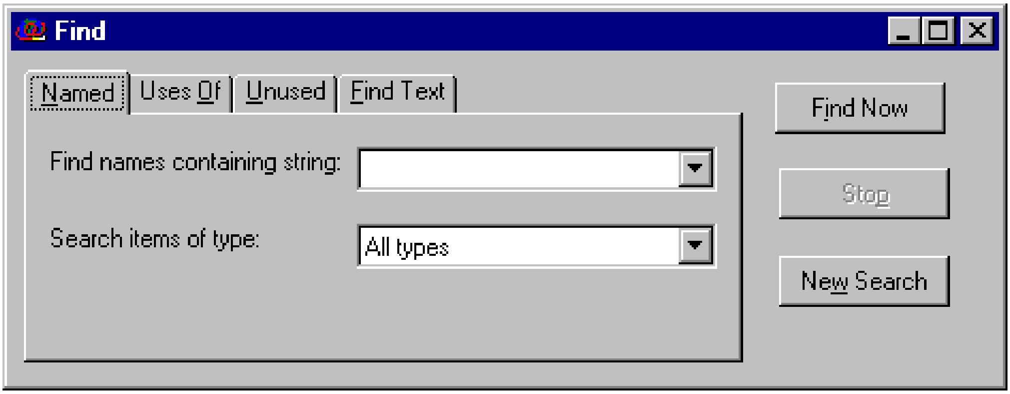


Figure 25. Find Window — Named Tab

To find a text string contained within the name of an item:

1. Type the text string in the **Find names containing string:** field.
2. Select the types of items you want to search in the current application from the **Search items of type:** pulldown menu.
3. Click **Find Now**.

   Voice@Work displays the results of the search in a box below the Find window (A in Figure 26) and highlights the item in the location where it finds the text string. If, for instance, the search finds the text string in the name of a node, it highlights the node in the call flow (B in Figure 26). If the search finds the text string in a phrase, it highlights the phrase in the appropriate phrase table.

   In many cases, the search results in multiple finds. To locate other search result items, click the **Item Name** in the results box (C in Figure 26). Voice@Work highlights the appropriate location of that item.

To clear the results of your search and start a new search, click the **New Search** button.

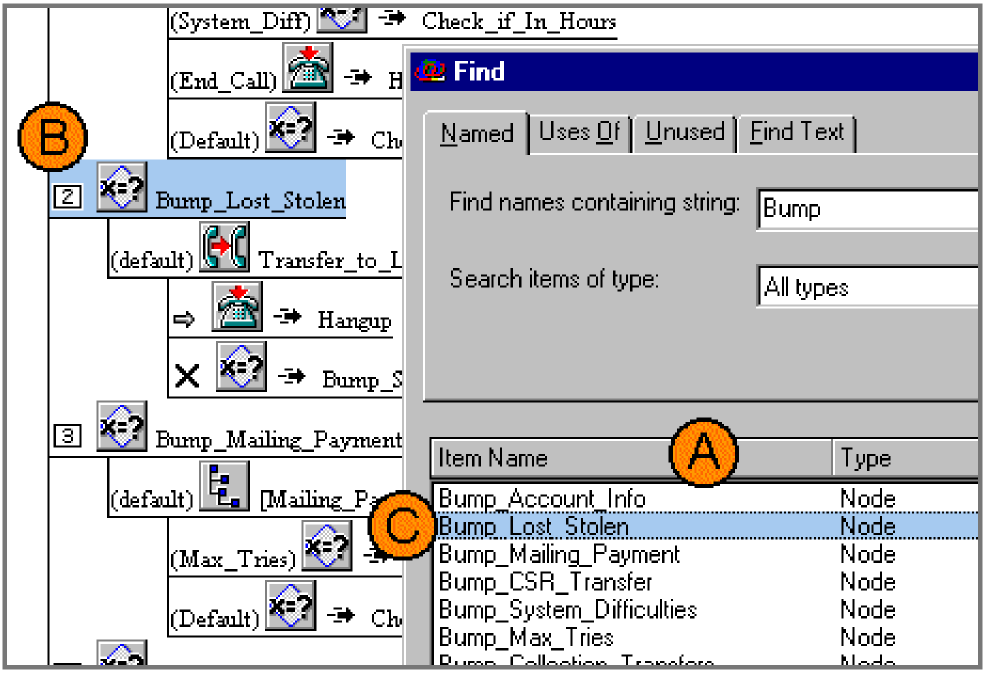


Figure 26. Named Tab Results Window

The Uses Of Tab

The Uses Of search option (Figure 27) searches for locations where items are used within the application. This can be useful in locating all the uses, for instance, of a certain call flow used as a subroutine within the application or locating where a particular external function is used.

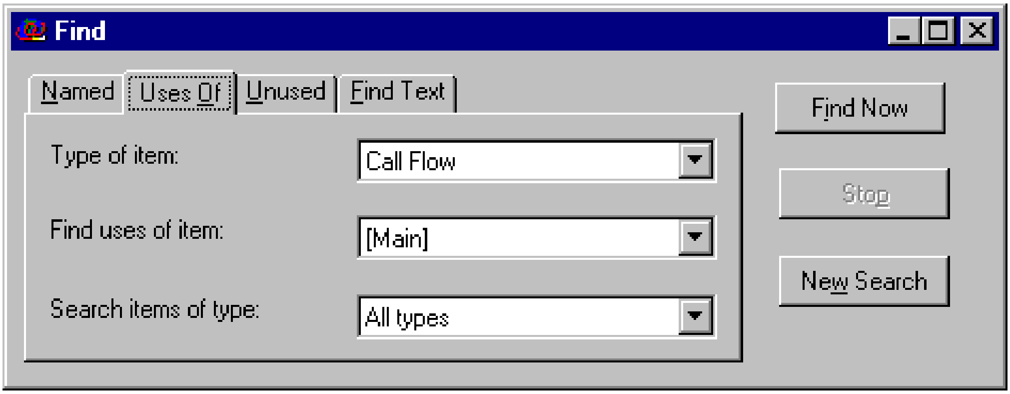


Figure 27. Find Window — Uses Of Tab

To find where selected items are used within the current application:

1. Select the kind of item you want to search for from the **Type of item:** pulldown menu.
2. Select which specific item of that type you want to find from the **Find uses of item:** pulldown menu.
3. Select which kinds of items you want to search through from the **Search items of type:** pulldown menu.

   In most cases, this will either be **All types** or the same as the **Type of item:** you selected in Step 1.
4. Click **Find Now**.

   Voice@Work displays the results of the search in a box below the Find window (A in Figure 28) and highlights the location of the item. If, for instance, you were searching for a particular phrase, it highlights the phrase in the appropriate phrase table (B in Figure 28).

   In many cases, the search results in multiple finds. To locate other search result items, click the **Item Name** in the results box (C in Figure 28). Voice@Work highlights the appropriate location of that item.

To clear the results of your search and start a new search, click the **New Search** button.

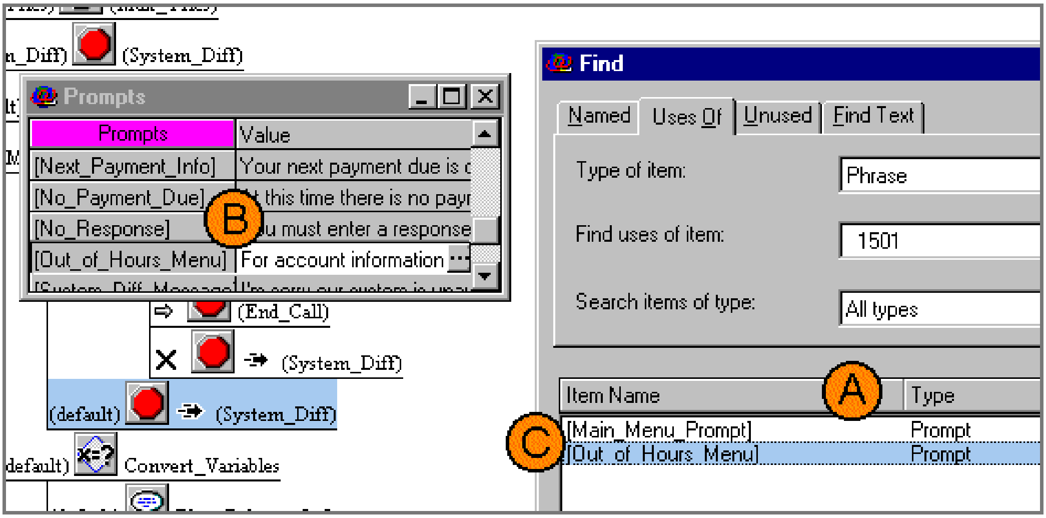


Figure 28. Uses of Tab Results Window

The Unused Tab

The Unused search option (Figure 29) searches for items that have been created but not used within the application. This can be useful in locating unused items that you may want to delete or items you have created but not yet used.

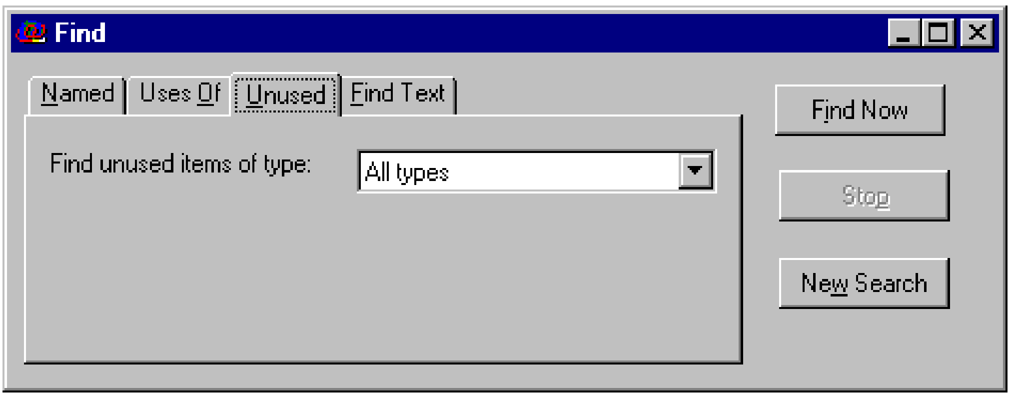


Figure 29. Find Window — Unused Tab

To find items that have been created but not used in the current application:

1. Select the types of items you want to search in the current application from the **Search items of type:** pulldown menu.
2. Click **Find Now**.

   Voice@Work displays the results of the search in a box below the Find window (A in Figure 30) and highlights the item. If, for instance, your search locates an unused call flow, Voice@Work makes that call flow the active call flow and displays it (B in Figure 30).

   To locate other find result entries, click that **Item Name** in the results box (C in Figure 30). Voice@Work displays the appropriate item.

To clear the results of your search and start a new search, click the **New Search** button.

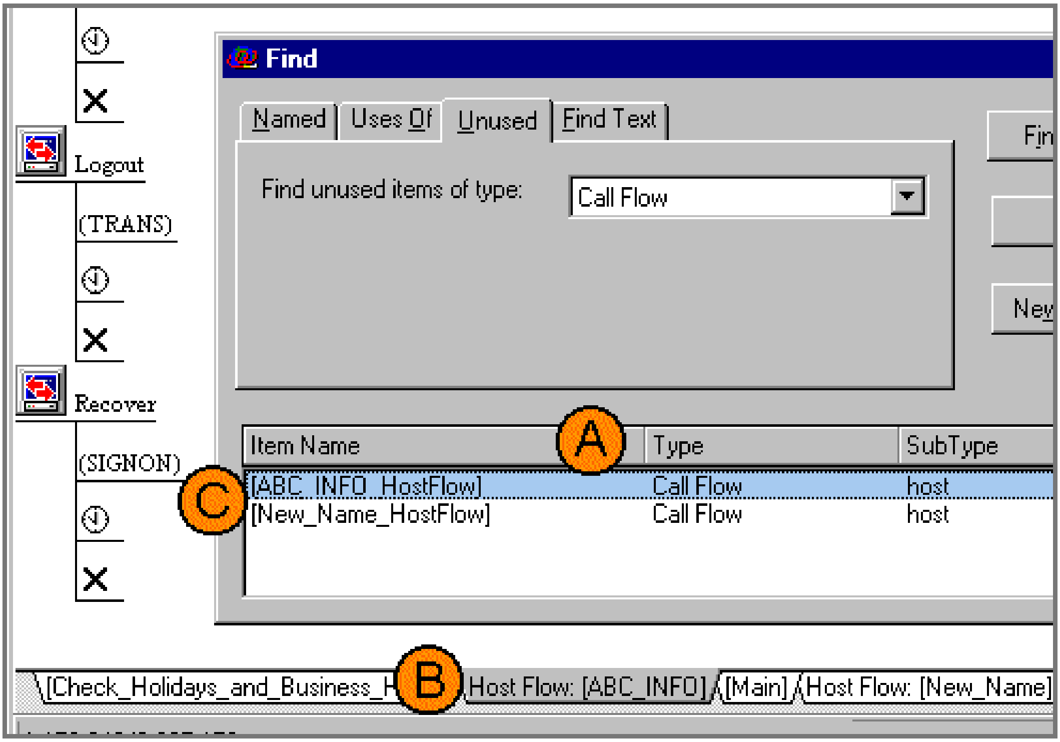


Figure 30. Unused Tab Results Window

The Find Text Tab

The Find Text search option (Figure 31) searches for text strings located anywhere within the application.

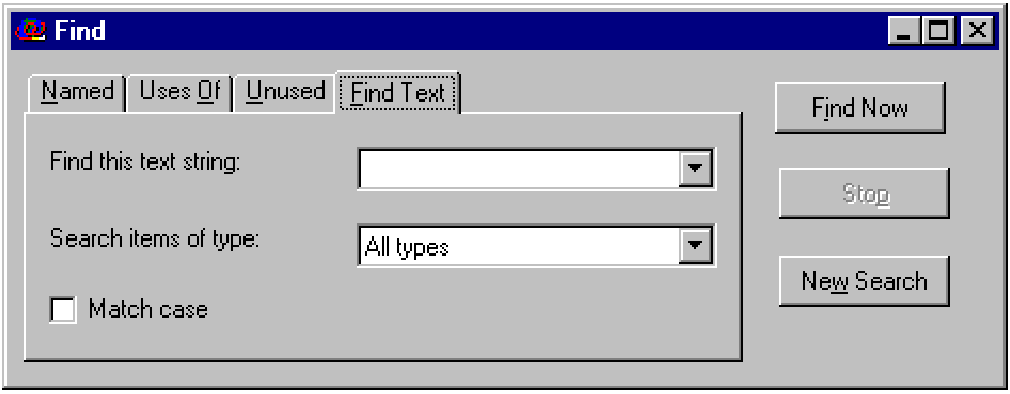


Figure 31. Find Window — Find Text Tab

To find a text string contained within an item:

1. Type the text string in the **Find this text string:** field.
2. Select the types of items you want to search in the current application from the **Search items of type:** pulldown menu.
3. If it is important to match cases (for instance, in searching proper names or titles), check the **Match case** checkbox.

   A checkmark appears to confirm that case matching is active.
4. Click **Find Now**.

   Voice@Work displays the results of the search in a box below the Find window (A in Figure 32) and locates and highlights the item where it finds the text string (B in Figure 32). If, for instance, the search finds the text string in a node, it highlights the node in the appropriate call flow. If the search finds the text string in a prompt, it highlights the prompt in the appropriate prompt table.

   In many cases, the search results in multiple finds. To locate other search result items, click the Item Name in the results box (C in Figure 32). Voice@Work highlights the appropriate location of that item.

To clear the results of your search and start a new search, click the **New Search** button.

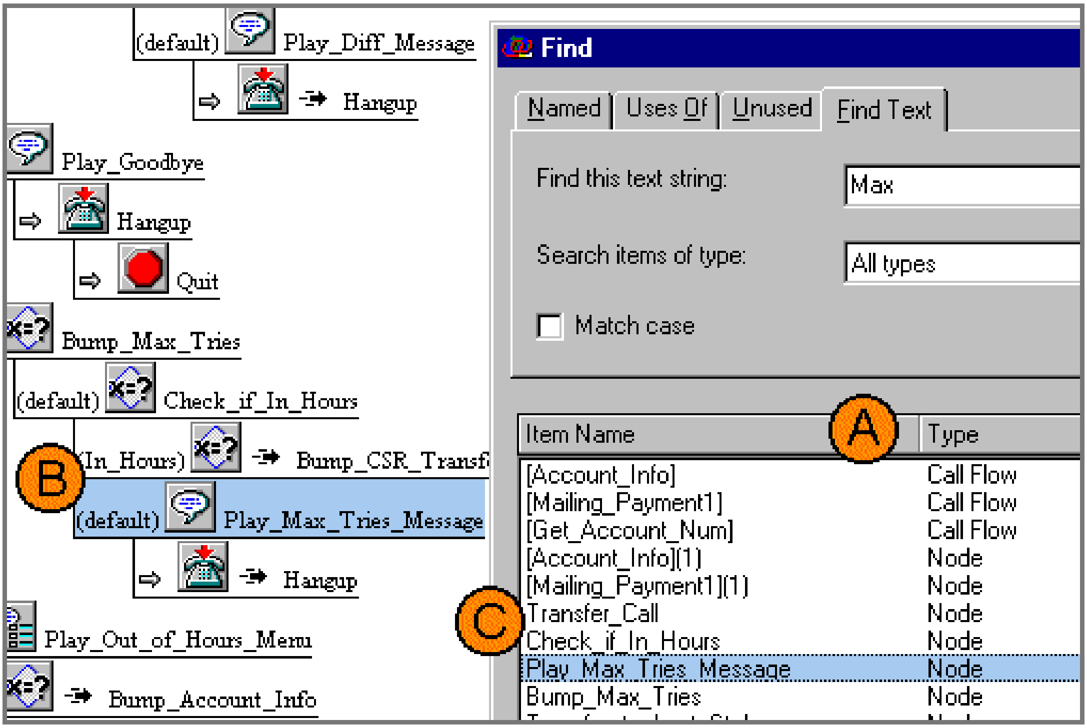


Figure 32. Find Text Results Window

The View Menu 7

Overview

Items on the View menu determine what is and is not displayed in the main window. The View menu consists of two groups of options:

- Palette-viewing options
- Globals-viewing options

Purpose

The purpose of this chapter is to summarize the options on the View menu. Topics include:

- Opening the View Menu
- Palette Options
- Globals Manager Options

Opening the View Menu

To open the View menu, do one of the following:

- Click **View** on the main window menu bar.
- Press ALT V.

Palette Options

Clicking on the name of a palette toggles it on and off.

NOTE:
If a palette has been hidden (unchecked) using the View menu options and then toggled on again, the palette displays at the bottom of whatever palettes are currently displayed.

Table 8 describes the palette options available on the View menu.

Table 8. Palette Options on the View Menu

Option	Description/Comments
Toolbar	Toggles to display and hide the Toolbar. When checked, the Toolbar is displayed.
Node Palette	Toggles to display and hide the Node Palette. When checked, the Node Palette is displayed.
External Functions Palette	Toggles to display and hide the External Function Palette. When checked, the External Function Palette is displayed.

Globals Manager Options

Clicking on the name of a global resource manager displays that resource manager wherever it was last placed.

NOTE:
If any of the global resource managers are docked to the Globals Manager, clicking on their names in the View menu (or on their icons on the Toolbar) displays the entire Globals Manager. To display only one resource manager (Prompts, for example), you must first undock the resource manager from the Globals Manager (see Chapter 13, "The Globals Manager" for the procedure).

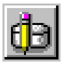
Table 9 describes the resource manager options available on the View menu.

Table 9. Globals Manager Options the View Menu

Option	Icon	Description/Comments
Globals		Opens the Globals Manager; see Chapter 13, "The Globals Manager" for the procedure).
Variables		Opens the Variables Manager; see Chapter 14, "Working with Variables" for the procedure. **NOTE:** If the Variables Manager is docked to the Globals Manager, the entire Globals Manager is displayed.
Phrases		Opens the Phrases Manager; see Chapter 18, "Working with Phrases" for the procedure. **NOTE:** If the Phrases Manager is docked to the Globals Manager, the entire Globals Manager is displayed.
Prompts		Opens the Prompts Manager; see Chapter 19, "Working with Prompts" for the procedure. **NOTE:** If the Prompts Manager is docked to the Globals Manager, the entire Globals Manager is displayed.

Continued on next page

Table 9. Globals Manager Options the View Menu — *Continued*

Option	Icon	Description/Comments
Tables		Opens the Tables Manager; see Chapter 20, ''Working with Tables'' for the procedure. **NOTE:** If the Tables Manager is docked to the Globals Manager, the entire Globals Manager is displayed.
External Functions		Opens the External Functions Manager; see Chapter 16, ''Working with External Functions'' for the procedure. **NOTE:** If the External Functions Manager is docked to the Globals Manager, the entire Globals Manager is displayed.
FlexWords		Opens the FlexWords Manager; see Chapter 15, ''Working with FlexWord Speech Recognition'' for the procedure. **NOTE:** If the FlexWords Manager is docked to the Globals Manager, the entire Globals Manager is displayed.
Hosts		Opens the Hosts Manager; see Chapter 17, ''Working with Hosts'' for the procedure. **NOTE:** If the Hosts Manager is docked to the Globals Manager, the entire Globals Manager is displayed.

The Window Menu

8

Overview

The Window menu provides standard Windows operating system commands for controlling the non-modal call flow windows in Voice@Work. Non-modal window capability means that you can have more than one call flow window open at one time and switch back and forth between them easily.

Purpose

The purpose of this chapter is to provide a description of the Window menu and its options. Topics include:

- Opening the Window Menu
- Window Menu Availability
- Window Menu Options

Opening the Window Menu

To open the Window menu, do one of the following:

- Click **Window** on the main window menu bar.
- Press ALT W.

Window Menu Availability

The Window menu is available only when the **Multiple Document Interface** option has been selected in the Preference Editor. See Chapter 27, "The Preference Editor" for information on selecting the Multiple Document Interface option.

When you select the **Multiple Document Interface** option, Voice@Work displays each call flow for an application in its own non-modal window. Non-modal window capability means that you can have more than one call flow window open at one time and switch back and forth between them easily. The Window menu offers Windows-standard options for arranging and displaying the call flow windows.

Window Menu Options

Table 10 describes the options available on the Window menu. For more information on these options and how to use them, see the documentation for your Windows operating system.

Table 10. Window Menu Options

Option	Action/Description
Cascade	Overlaps all open windows in a cascading pattern starting in the upper left corner of the Design Pad with the most recently activated windows on top.
Tile	Arranges all active windows to fill the available space on the Design Pad. Windows are tiled top-to-bottom and left-to-right, and progressing in order from the most recently active. TIP: To arrange the active windows in a particular order, reactivate them by clicking in them one at a time *in reverse order* to the way you want them to appear on the Design Pad, and then click the **Tile** option.
Arrange Icons	Inactive in the current version.
Minimize All	Minimizes all open windows. Each window must be maximized to view the contents.
[Windows available]	Lists all currently open windows, even if minimized, in the order in which they were opened.

The Tools Menu

9

Overview

The Tools menu contains options for using a variety of development tools to help you edit, diagnose, and debug your applications.

Purpose

The purpose of this chapter is to briefly describe the development tools Voice@Work offers on the Tools menu. Topics include:

- Opening the Tools Menu
- Summary of Options

Opening the Tools Menu

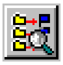
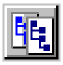
To open the Tools menu, do one of the following:

- Click **Tools** on the main window menu bar.
- Press ALT T.

Summary of Options

The development tools on the Tools include editing tools, diagnostic tools, and debugging tools. Table 11 describes the development tools on the Tools menu. For more information about each of these tools, see the appropriate chapter in this book.

Table 11. Tools Menu Options

Option/Tool	Icon	Description/Comments
Application Resources		The Application Resources tool allows you to: ■ See what resources are in use with the current application ■ Import additional resources into your application as needed, either as Private Copies or Shared ■ See what resources in your application are being shared
Call Flow Inventory		The Call Flow Inventory tool allows you to: ■ See what call flows exist for your application ■ See where each call flow is used within the application, and what other call flows (if any) are used within each call flow ■ Locate a specific call flow within an application ■ Provide an alternate means of dragging a call flow to the Design Pad

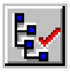
Continued on next page

Table 11. Tools Menu Options — *Continued*

Option/Tool	Icon	Description/Comments
Code Generation/ Application Transfer		The Code Generation tool generates the code necessary to transfer and install the application onto the target system. **NOTE:** Voice@Work currently generates code only for INTUITY CONVERSANT V6 and V7 systems. The Application Transfer tool transfers the application files to the target system.
Events String Editor		The Events String Editor tool allows you to: ■ Determine which variables will be treated as Call Data Events on the target system. ■ Display both a list of all the variables defined for the application and a list of the variables selected to be treated as Call Data Events.
Node Inspector		The Node Inspector tools allows you to: ■ Edit node attributes easily and quickly. ■ Display different information and editing options for each node, depending on the node’s predetermined attributes.
Preference Editor		The Preference Editor allows you to: ■ Set application and toolwide preferences for using Voice@Work. ■ Control the working environment for Voice@Work

Continued on next page

Table 11. Tools Menu Options — *Continued*

Option/Tool	**Icon**	**Description/Comments**
Simulation		The Simulation tool allows you to: ■ Run your application in a simulated environment. ■ Find problems within the application, thus reducing the amount of time required to test your application on the target system.
Verify Design		The Verify Design tool allows you to: ■ Check your application design for possible errors or omissions. ■ View each call flow's results in a separate display. ■ Easily locate nodes with possible errors or omissions.

The Sharing Menu

10

Overview

Many large applications require the work of more than one developer to write and assemble. Voice@Work accommodates this fact by allowing developers to share application resources (such as database tables, host definitions, and phrase tables) across applications and PCs.

The Sharing menu contains two options for updating shared resources:

- The Refresh from Shared Resources tool allows developers to update local copies of any application resources *shared-in* from another source.
- The Update Shared Resources tool allows developers to update any application resources they have designated as *shared-out*.

Purpose

The purpose of this chapter is to provide information about the Sharing menu and its options. Topics include:

- About Sharing Application Resources
- Guidelines for Sharing Resources
- Opening the Sharing Menu
- Sharing Menu Options
- Sharing-Out an Application Resource
- Updating Shared-Out Resources
- Refreshing Shared-In Resources

About Sharing Application Resources

Application resources can be shared between applications on the same PC or between different PCs using mapped network drives.

Application resources that can be shared in Voice@Work include:

- Host definition files
- Database tables
- Custom external functions
- Custom phrase table lists (not speech files)
- FlexWord lists

Defining Sharing Terminology

The developer who creates an application resource is the resource's *owner*.

The application resource can be shared only when the owner designates it as a Shared Resource. This is known as *sharing-out* the application resource (see "Sharing-Out an Application Resource" in this chapter).

When a shared-out resource is imported into and used by another application, it is said to be *shared-in*.

Setting Up for Sharing Resources

When Voice@Work is first installed, it creates a directory named "Shared", with subdirectories for each type of resource that can be shared. When a resource is designated as a shared-out resource, Voice@Work places it in the appropriate subdirectory.

Global resource managers for resources that can be shared-out have **Sharing...** menu options on their right-click popup menus.

The owner has the responsibility for deciding what read and write permissions to set on the shared resource directories and subdirectories. Only the owner of a shared resource may update it, using the Update Shared Resources tool (see "Updating Shared-Out Resources" below).

Any developer who wants to use the shared-out application resource must first map the shared resource's subdirectory as a network drive. It is only at this point that the shared resource appears on the developer's PC.

Once the directory is mapped, the developer can share-in the application using the Application Resources tool (see Chapter 21, "The Application Resources Tool"). To use the shared resource, the developer must import the resource either as shared or as a private copy.

Guidelines for Sharing Resources

When sharing application resources, keep the following guidelines in mind:

— Whether the developer imports a resource as shared or as a private copy, Voice@Work actually makes a local copy of the resource. This allows the developer to continue working on the application, even if the network is down.

The **Shared** option means that the developer cannot alter the original resource, but may only access it. The Refresh From Shared Resources tool allows the developer to update a shared-in resource to the current state of the shared-out resource at any time (see "Refreshing Shared-In Resources" below).

NOTE:
Before you can use the Refresh from Shared Resources tool, you must have already imported one or more resources as a shared resource. Standard application resources (such as external functions) that come with Voice@Work cannot be shared-in.

— Shared-in resources are not automatically updated as part of the Verify Design and Code Generation tools' operations. It is the developer's responsibility to update shared-in resources before using either of these tools.

— If the owner of a shared-out resource deletes it (using the Delete option from the right-click popup menu of that resource), Voice@Work warns the owner that it is a shared resource and might be in use by other developers. If the owner chooses to continue, Voice@Work deletes the resource from the Shared directory and corresponding subdirectory.

Any developers who may be using it as a shared-in resource are notified of that fact the next time they try to refresh the shared resource (using the Update From Shared Resources tool). At that point, the developer has the option to retain the local copy and become the new owner of the shared resource or delete it altogether.

If more than one developer is using the shared resource, each can become an owner of it independently of the others. This creates two resources of the same name on different PCs. If they later try to install the application on the same target system, they must then decide which one to use, whether to overwrite one with the other, or to rename one of them.

Opening the Sharing Menu

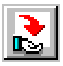
To open the Sharing menu, do one of the following:

- Click **Sharing** on the main window menu bar.
- Press ALT S.

Sharing Menu Options

The Sharing menu has two options related to updating shared resources. Table 12 describes these options.

Table 12. Sharing Menu Options

Option	Icon	Action/Description
Refresh From Shared Resources		Allows you to update local copies of resources imported to your application
Update Shared Resources		Allows the owner of a shared resource to update the resource

Sharing-Out an Application Resource

To designate an application resource you own as shared-out, making it available to other developers and applications, do the following:

1. Open the global resource manager for the type of resource you want to share (see the chapter for the type of global resource you want to share).
2. Right-click on the name of the resource you want to share.
3. Select **Sharing...** from the popup menu.

   The Application Sharing dialog box appears (Figure 33).

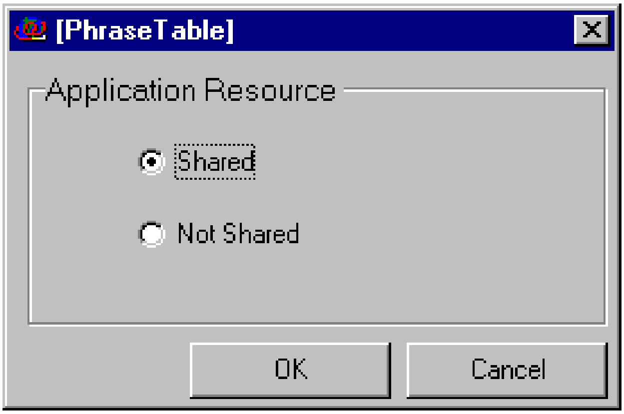


Figure 33. Application Sharing Dialog Box

4. Select the **Shared** radio button.
5. Click **OK**.

Updating Shared-Out Resources

When you designate an application resource as shared, Voice@Work makes it available to any other developer who has the ability to map a network drive to your PC.

For this reason, you should update your shared resources from time to time, to make the most recent version of the shared resource available to any developer who might be using it. To update these shared resources, use the Update Shared Resources tool.

NOTE:
Before you can use this tool, you must have already designated one or more resources as a shared resource (see "Sharing-Out an Application Resource" in this chapter).

To update your shared-out resources, do the following:

1. Do one of the following:
   - From the **Sharing** menu, select **Update Shared Resources**.
   - Click the  button on the Toolbar.

   The following message appears (Figure 34):

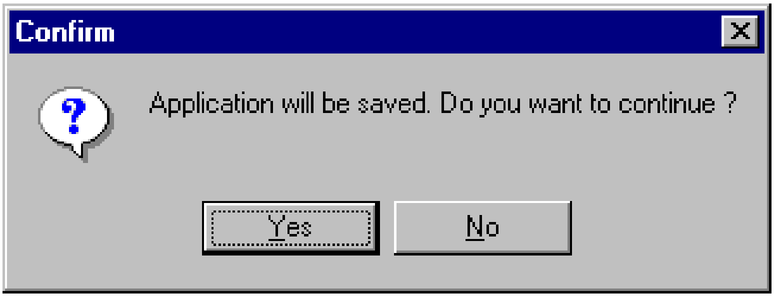


Figure 34. Confirm Update Shared Resources Dialog Box

2. Click **Yes**.

   Voice@Work updates all application resources you have designated as shared, and the following message appears (Figure 35):

Figure 35. Update Shared Resources Complete Message

3. Click **OK**.

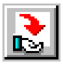
Refreshing Shared-In Resources

When you share-in (import as shared) an application resource, Voice@Work makes a local copy so you can continue to work with it even if the network is down.

For this reason, you should refresh your shared-in resources from time to time, to ensure that you are working with the most recent version of the shared resource. To refresh these shared-in resources, use the Refresh from Shared Resources tool.

NOTE:
Before you can use this tool, you must have already imported one or more resources as a shared resource. Standard application resources, such as external functions, that come with Voice@Work cannot be shared-in.

To refresh your shared-in resources, do the following:

1. Do one of the following:
   - From the **Sharing** menu, select **Refresh from Shared Resources**.
   - Click the  button on the Toolbar.

   The following warning message appears (Figure 36):

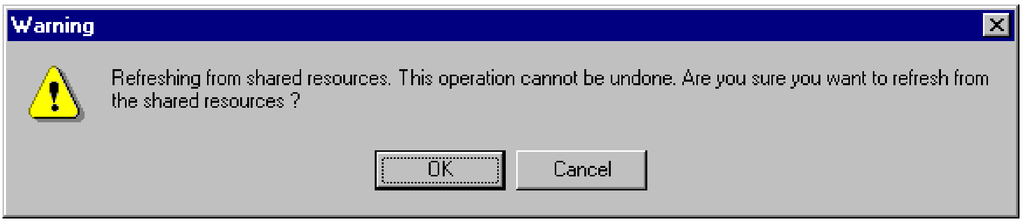


Figure 36. Refresh from Shared Resources Warning Message

2. Click **OK**.

   The Resource Refresh Results window appears summarizing the results of the Refresh from Shared Resources action (Figure 37).

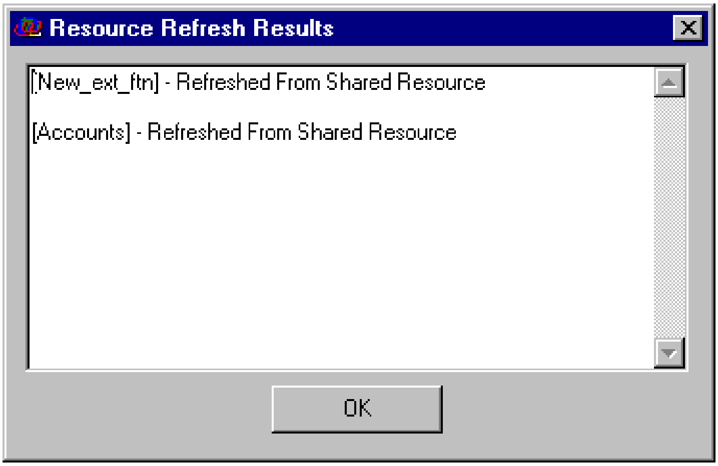


Figure 37. Resource Refresh Results Window

3. Click **OK**.

The Help Menu

11

Overview

The Help menu provides two means of accessing the online Help for Voice@Work.

Purpose

The purpose of this chapter is to provide information about the Help menu. Topics include:

- Opening the Help Menu
- Summary of Options

Opening the Help Menu

To open the Help menu, do one of the following:

- Click **Help** on the main window menu bar.
- Press ALT H.

Summary of Options

The Help menu has three options. Table 13 describes the Help menu options.

Table 13. Help Menu Options

Option	Keyboard Shortcut/ Icon	Description/Action
Contents...	F1 	Opens the online Help to the Voice@Work Help Topics. From there, you can link to topics of interest.
Search for Help on...		Opens the online Help index. Type in the first few letters of the topic you are looking for, then click **Display**.
About...		Displays the About Lucent Technologies Voice@Work information box (Figure 38).

Figure 38. Voice@Work About... Window

Palettes

12

Overview

Voice@Work has three palettes, each with its own function and set of icons. The methods for configuring these palettes are basically the same.

Each palette can be configured independently of the others:

- Icons can be rearranged and displayed or hidden.
- Groups of icons can be rearranged.
- Palettes can be undocked from or docked to the menu bar.

Icons for all three palettes are described in greater detail in Appendix B, “Voice@Work Icons”.

Purpose

The purpose of this chapter is to describe palette configuration options and to provide procedures for configuring the Voice@Work palettes. Topics include:

Opening a Palette Configuration Window

To open a window for configuring a palette, do one of the following:

- From the **Edit** menu, select the appropriate **Configure [Palette]...** option.
- Right-click on the appropriate palette (if it is displayed), and then select **Configure...** from the popup menu.

Voice@Work displays the selected palette configuration window (Figure 39).

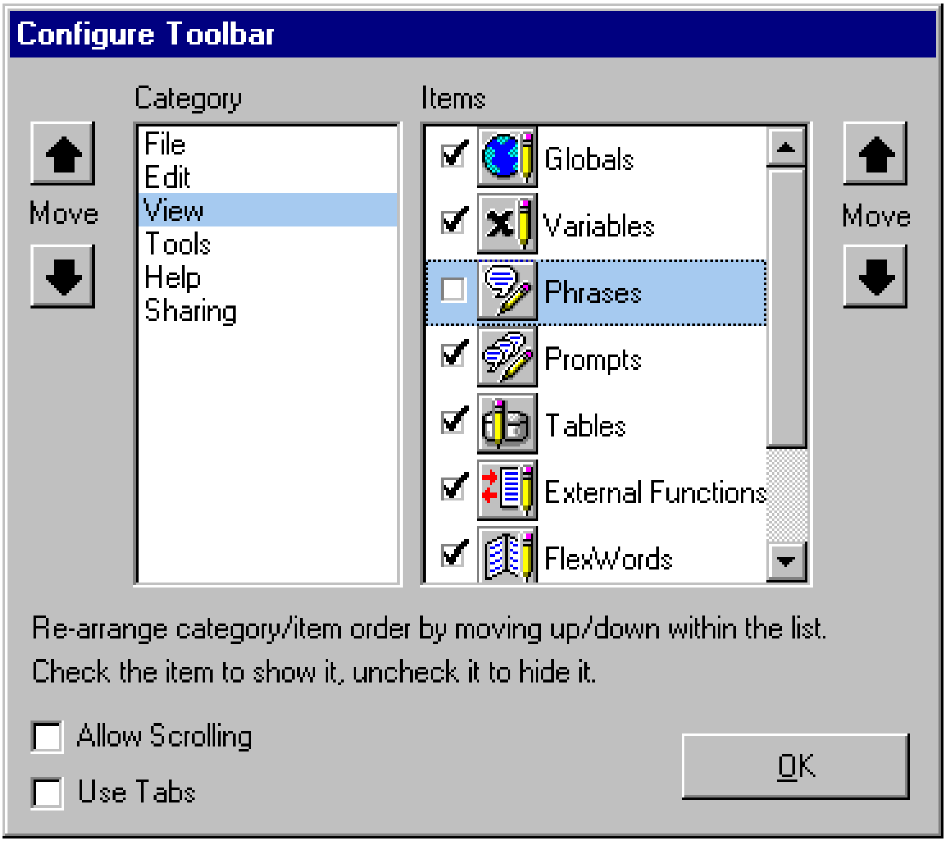


Figure 39. Configure Palette Window

Rearranging Icons on a Palette

To customize the placement of icons on any of the palettes, you can:

- Move whole groups (categories) of icons at once.
- Move individual icons within a group.
- Choose whether or not each icon should be displayed.
- Allow the horizontal scrolling of icons on the palette.
- Select a tabbed view of the categories of icons on a palette.

Moving Categories (Groups) of Icons

To move a category, or group, of icons, do the following:

1. Open the appropriate palette configuration window.
2. Highlight in the **Category** pane the category (group) of icons you want to move.
3. To move the category of icons to the left on the palette, click the up arrow.
4. To move the category of icons to the right on the palette, click the down arrow.
5. Click **OK**.

Moving Individual Items (Single Icons)

To move a single icon, do the following:

1. Open the appropriate palette configuration window.
2. Highlight in the **Items** pane the item (individual icon) you want to move.
3. To move the item to the left on the palette, click the up arrow.
4. To move the item to the right on the palette, click the down arrow.
5. Click **OK**.

Selecting Icons for Display

To select or remove an icon for display, do the following:

1. Open the appropriate palette configuration window.
2. Highlight the Category of the icon you want to select or remove.
3. To select (display) an icon on the palette, check its box.
4. To remove (hide) an icon from the palette, clear its checkbox.

Using the Allow Scrolling Option

The **Allow Scrolling** option causes the icons of a palette not to wrap if they do not fit in the width of the window. Instead, they move out of the window. They can still be accessed using the scrolling arrows, however.

Figure 40 is an example of a palette with the **Allow Scrolling** option turned on.

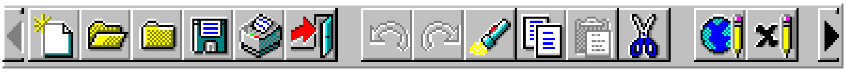


Figure 40. Allow Scrolling Option

To use the **Allow Scrolling** option, choose one of the following methods:

Method 1

1. Right-click anywhere on the appropriate palette.
2. Select **Allow Scrolling** from the popup menu.

Method 2

1. Open the appropriate palette configuration window.
2. Click the **Allow Scrolling** checkbox in the lower left corner of the window.
3. Click **OK**.

Using the Tabs Option

The **Use Tabs** option causes only one category (group) of icons at a time to display on the palette. To view another category of icons, you must click on the appropriate tab.

Figure 41 shows an example of a palette with the **Use Tabs** option turned on.

Figure 41. Use Tabs Option

To use the **Use Tabs** option, choose one of the following methods:

Method 1

1. Right-click anywhere on the appropriate palette.
2. Select **Use Tabs** from the popup menu.

Method 2

1. Open the appropriate palette configuration window.
2. Click the **Use Tabs** checkbox in the lower left corner of the window.
3. Click **OK**.

Docking/Undocking Palettes

You can undock (remove) a palette from the menu bar at the top of the design pad and place it wherever you want. You can also dock (attach) it again either to the top *or* the bottom of the design pad window.

Undocking a Palette

To undock a palette:

1. Click-and-drag anywhere in the palette area. Be careful *not* to click an icon.

   When you first click and hold the primary mouse button down in the palette area, the icons disappear and a dashed outline appears around the entire palette area. As you drag the palette away from the menu bar, the dashed outline changes to a solid black outline. At this point you can release the mouse button, and the palette remains where you release the button.

2. Place the palette where you want it.

   When you first release an undocked palette, it has the same configuration and general appearance as when it was docked.

   Once it is undocked, you can resize the new palette window and place it wherever you want. Limitations on the minimum width and height of the undocked palette window are determined by the number of icons within the groups (categories) and the number of categories.

   To resize the undocked palette, click and drag an edge or corner, the same as you would any other secondary window.

   To move the undocked palette, click-and-drag the window by the title bar to the new location.

Docking a Palette

To dock a palette:

1. Click-and-drag the window in the palette area to the top or to the bottom of the design pad. (Be careful not to click an icon.)

   When attempting to dock a palette, make sure you *do not* click-and-drag using the palette's title bar. If you do, you will not be able to dock the palette.

   When you first click and hold the primary mouse button down in the palette window, the icons disappear and a solid black outline appears around the entire window area. As you drag the window toward the menu bar or the bottom of the design pad, the solid black outline changes to a dashed outline.

2. Release the mouse button to attach the palette.

   NOTE:
   A newly docked palette always attaches itself at the top of the displayed palettes. Thus, if you want the palette at the top of the palettes, you must dock it last.

Right-Click Menu Options

The palette right-click menu is available by right-clicking anywhere on any of the palettes. It is not available within any of the **Configure [Palette]** windows.

Table 14 describes the options available on the right-click menu.

Table 14. Palette Right-Click Menu Options

Option	Description/Comments
Configure...	This option opens the palette configuration window for the selected palette.
Hide	This option hides the selected palette from view. **NOTE:** To view the palette again, you must select the appropriate palette from the **View** menu.

Continued on next page

Table 14. Palette Right-Click Menu Options — *Continued*

Option	Description/Comments
Show Hints	This option toggles on and off. When checked (toggled on), it displays “balloon”-type help for the palette elements.
Allow Scrolling	This option toggles on and off. When checked (toggled on), it turns on the **Allow Scrolling** option.
Use Tabs	This option toggles on and off. When checked (toggled on), it turns on the **Use Tabs** option.

The Globals Manager

13

Overview

Globals are alterable resources, such as prompts and database tables, that can be accessed from anywhere within an application.

By default, all the globals are organized together in a single display called the Globals Manager. Individual global resource managers can, however, either be docked (attached) to or undocked (separated) from the Globals Manager.

Purpose

The purpose of this chapter is to provide information about the Globals Manager and how to use it. Topics include:

- Opening the Globals Manager
- Maneuvering in the Globals Manager
- More about Tabstops
- Undocking and Docking Resource Managers
- Right-Click Menu Options
- List of Global Resources

Opening the Globals Manager

To open the Globals Manager, do one of the following:

- From the **View** menu, select **Globals**.
- Click the  button on the Toolbar.

Each type of global resource has its own window, where the resource elements and their attributes are displayed (Figure 42).

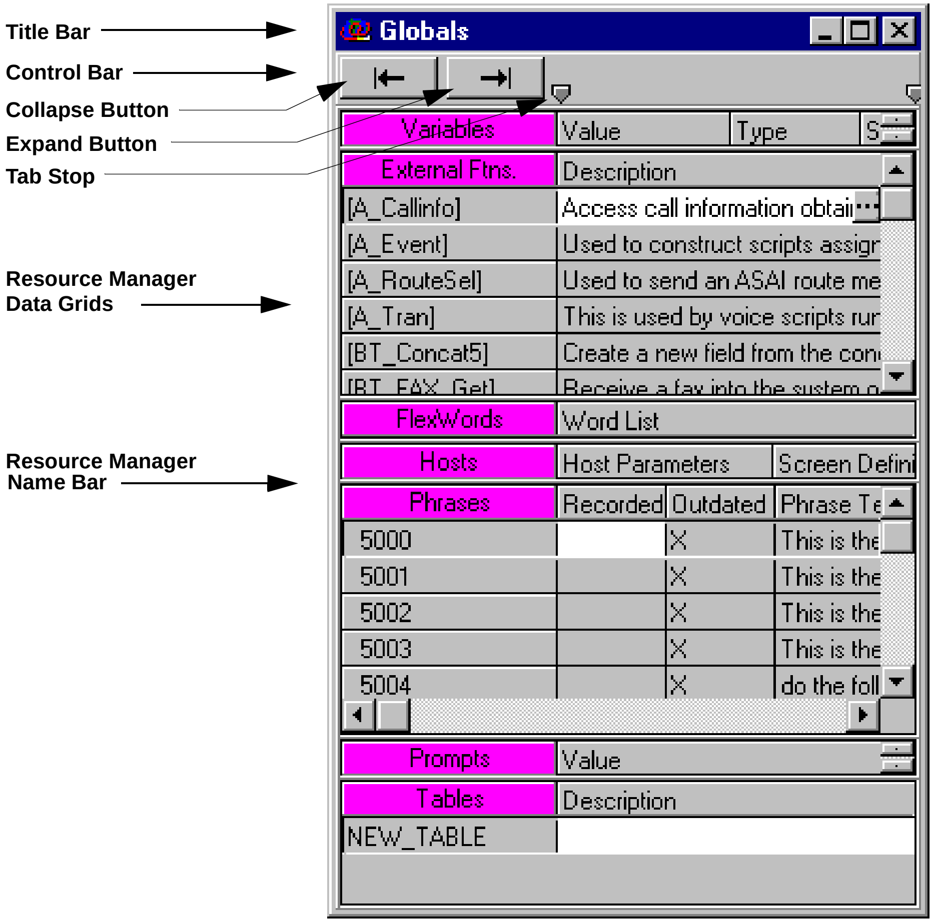
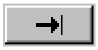
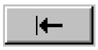


Figure 42. Globals Manager Window

Table 15 describes the elements of the Globals Manager window.

Table 15. Globals Manager Elements

Element	Example	Description/Comments
Title bar		Located at the top of the window. Contains the buttons for minimizing, maximizing, or restoring, and closing the Globals Manager window.
Control bar		Located immediately below the title bar. Contains the tabstops, Show More button, and Show Less button (see the next three elements).
Show More icon		Expands your view of the Globals Manager to the next tabstop (see below). If the last tabstop has been reached or exceeded, clicking this icon has no effect.
Show Less icon		Collapses your view of the Globals Manager to the next tabstop (see below). If the last tabstop has been reached or exceeded, clicking this icon has no effect.
Tabstop		Controls how far the Globals Manager window expands or collapses when the appropriate button is clicked. Can be dragged to any position in the control bar.
Resource manager name bar		Used to identify the individual global resource managers. Is distinguishable from other fields in a globals window by its magenta color.
Resource manager header bar		Located adjacent to the name bar (see above). Identifies the data columns within each global resource manager.
Data displays		Fields located underneath the name bar and header bars. Used to display the appropriate data for each column.

Maneuvering in the Globals Manager

You can expand or collapse the entire Globals Manager by clicking on the Show More or Show Less buttons at the top of the window (Figure 42 and Table 15).

There are also tabstops at the top of the Globals Manager window. You can set these tabstops to control how far the Globals Manager expands or collapses. Move these tabstops where you want them by clicking-and-dragging them horizontally within the control bar (Figure 42 and Table 15).

NOTE:
For more information about using tabstops, see "More about Tabstops" below.

You can open or close a particular global resource manager by double-clicking on the resource manager's name bar. You can also expand or collapse a resource manager by clicking-and-dragging the resource manager's name bar up or down. (In Figure 42, the Variables, FlexWords, Hosts, and Prompts managers have been closed. The External Ftns., Phrases, and Tables managers have been opened.)

NOTE:
If a global resource manager is between two other resource managers, you are limited in how far you can move that resource manager's name bar up or down. You cannot go past the other two resource managers' name bars or move them in tandem. In Figure 42, for instance, the Prompts Manager name bar can only be moved within the space between the Phrases Manager and the Tables Manager name bars.

More about Tabstops

With Voice@Work, you can create or delete tabstops and place them where you want them on the Globals Manager control bar. These tabstops control how far the Globals Manager expands or collapses when the appropriate buttons are clicked in the control bar (Figure 42 and Table 15).

To Create a Tabstop

1. Right-click in the control bar somewhere other than on the Show More or Show Less icons (Figure 42 and Table 15).
2. Click the **New Tabstop** option.

   Voice@Work displays a new tabstop in the control bar.
3. Drag the tabstop to place it where you want it in the control bar.

To Delete a Tabstop

1. Right-click the tabstop you want to delete.
2. Select the **Delete Tabstop** option.

   The tabstop disappears.

Using Tabstops

The Show More icon (Figure 42 and Table 15) does not expand the Globals Manager window beyond the last tabstop currently in place. To expand the Globals Manager window beyond the last tabstop, do the following:

1. Resize the window by clicking-and-dragging the window edge (as you would any other secondary window).
2. Move an existing tabstop to the desired new location or create another tabstop and place it further out.

All tabstops will not be visible until you have expanded the Globals Manager to the last one. Thus, to move or set all tabstops, you must do one of the following:

- Expand the Globals Manager as far as you can using the Show More icon (Figure 42 and Table 15).
- Maximize the Globals Manager window.

TIP: You might want to set:

- A first tabstop so that the Globals Manager displays only the first column with the resource manager name visible
- Other tabstops so that you can see the desired amount of identifying information for each resource manager
- The last tabstop so that the Globals Manager expands to the entire width of your screen

Undocking and Docking Resource Managers

To undock a resource manager, drag the resource manager's name bar (Table 15) away from the Globals Manager. As you drag the resource manager away from the Globals Manager, a gray box with a solid black outline appears. Release the button wherever you want to place the resource manager. You can now treat the resource manager as a separate entity. You can open, close, move, and resize it, regardless of the status of the Globals Manager.

To dock a resource manager, drag the resource manager's name bar (Table 15) until the name bar is inside the Globals Manager. As you drag the resource manager into the Globals Manager area, the cursor changes to an arrow with a small white box. Release the button. The resource manager automatically reattaches itself to the *bottom* of the Globals Manager.

NOTE:
Do not attempt to dock a resource manager by dragging it by its *title bar*. It will not dock to the Globals Manager unless you drag it using the name bar.

Right-Click Menu Options

Right-click menu options that appear in the Globals Manager depend on where exactly you right-click.

Table 16. Globals Manager Right-Click Menu Options

Location	Menu Option	Description
In the control bar	New Tabstop	Creates a new tabstop and places it on the control bar (Table 15 and Figure 42).
	Stay On Top (toggle option)	When checked, causes the Globals Manager to stay on top of all other Voice@Work windows. When unchecked, allows the Globals Manager to move to the background when another window is selected as the active window.
On an existing tabstop	New Tabstop	Creates a new tabstop and places it on the control bar.
	Delete Tabstop	Deletes the tabstop under the cursor.

Continued on next page

Table 16. Globals Manager Right-Click Menu Options — *Continued*

Location	Menu Option	Description
On an individual resource manager's name bar	Open/Close (toggle option)	Opens or closes the individual resource manager. **NOTE:** This action is inactive if the resource manager has been undocked from the Globals Manager.
On other resource managers' fields:		Calls up a choice menu. The choices available depend on which resource manager you are in. For more details on the specific menu choices available, see the chapter for the resource manager you are interested in.

List of Global Resources

Below is a list of the resource managers available through the Globals Manager and the chapters in which information about each is located.

- Variables — Chapter 14, "Working with Variables"
- FlexWord speech recognition — Chapter 15, "Working with FlexWord Speech Recognition"
- External functions — Chapter 16, "Working with External Functions"
- Remote hosts — Chapter 17, "Working with Hosts"
- Phrases — Chapter 18, "Working with Phrases"
- Prompts — Chapter 19, "Working with Prompts"
- Database tables — Chapter 20, "Working with Tables"

Working with Variables

14

Overview

Variables in Voice@Work refer to application-specific information holders. They may be used to collect and hold data, act as counters, or make use of system information. Variables are grouped together and accessed through the Variables Manager.

Purpose

The purpose of this chapter is to provide information about using variables in Voice@Work applications. Topics include:

- The Variables Manager
- Creating New (Custom) Variables
- Renaming Custom Variables
- Setting or Editing a Variable's Attributes
- System Variables
- Right-Click Menu Options for the Variables Manager

The Variables Manager

The Variables Manager displays all variables defined for the current application, along with their attributes.

From the Variables Manager, you can:

- Make use of system variables that come as part of the Voice@Work package (see "System Variables" below).
- Create new variables.
- Rename custom variables.
- Set or Edit a variable's attributes.
- Use the right-click menu to perform a variety of other operations on variables.

Opening the Variables Manager

To open the Variables Manager, do one of the following:

- From the **View** menu, select **Variables**.
- Click the  button on the Toolbar.
- Open the Globals Manager and locate the Variables Manager (if it is docked there).

Voice@Work opens the Variables Manager window (Figure 43).

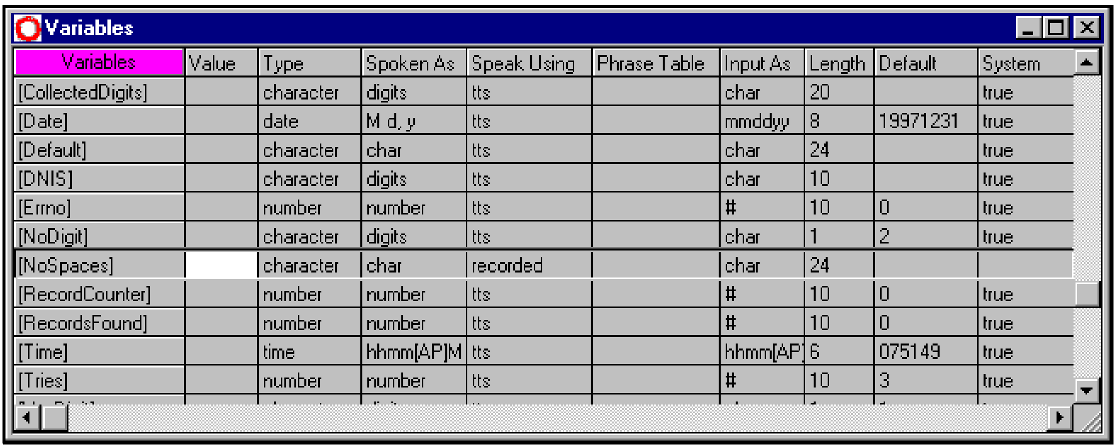


Figure 43. Variables Manager Window

The following sections describe the attributes (properties) of each field in the Variables Manager.

Name Column

The name column is located underneath the Variables Manager name bar. It displays the names of all variables defined for the current application in alphabetic order. Each variable name is enclosed in square brackets [ ].

Value Column

The value displayed in this field is the current value of the variable. This field remains blank until you run the Simulation tool, when it becomes populated with values generated during the simulation.

Type Column

The Type chosen for a variable influences several of its other attributes. The six basic variable types, with their default values and accepted input, are listed in Table 17.

NOTE:
The following options are for US English only. For information about options specific to other EBS language packages, see the “Enhanced Basic Speech Formats” appendix in *INTUITY™ CONVERSANT® System Speech Development, Processing, and Recognition*, for your system.

Table 17. Variable Types and Their Attributes

Type	Spoken As	Input As	Length	Default	Notes
Character	A (tts) C (recorded)	char	24		Accepts as input all standard ASCII characters, including special characters and punctuation (The extended ASCII character set is not supported.)
Currency	AN$D0 (tts) N$ (recorded)	#*00	15	0.00	Accepts as input numerals (0–9), a dollar sign, a single leading minus sign, and a decimal point
Date	ADMDYY (tts) D (recorded)	mmddyy	8	19980101	Accepts as input numerals (0–9), hyphens between numerals, slashes, and spaces
Number	AN (tts) N (recorded)	#	10	0	Accepts as input numerals (0–9) and a leading minus sign
Real	AND1 (tts) ND0 (recorded)	#*#	10	0.0	Accepts as input numerals (0–9), a leading minus sign, and a decimal point
Time	ATHMAM (tts) T (recorded)	hhmm(AP)M	6	000000	Accepts as input numerals (0–9) and the letters A, P, and M

For more information about the **Spoken As**, **Input As**, **Length**, and **Default** fields, see the appropriate sections below.

Spoken As Column

The value that appears in the Spoken As column of the Variables Manager depends on what type of variable it is and what value you set for it.

When you select a new Type for the variable, Voice@Work automatically assigns it a default value in the Spoken As column. If you change the value in the Speak Using column from **tts** to **recorded** or from **recorded** to **tts**, the value in this field automatically changes to the default value for the new **Speak Using** option.

To set or reset the values in the **Spoken As** field, click in the field and select the option you want from the pulldown menu.

NOTE:
When you open the pulldown menu, a context-sensitive popup help message appears when you place the cursor over an option on the menu. This popup message gives you a brief description of what that option does.

Character Spoken As Options

For recorded speech, individual characters in the string are spoken in succession, with the specified inflection, if any. For Text-To-Speech (TTS), the string may either be treated as individual characters or read as words.

For both types, a null character field is spoken as nothing. If a character within a field is not valid for that type, it is ignored.

Table 18 summarizes the Spoken As options available when the variable is of type character.

NOTE:
The following options are for US English only. For information about options specific to other EBS language packages, see the “Enhanced Basic Speech Formats” appendix in *INTUITY™ CONVERSANT® System Speech Development, Processing, and Recognition*, for your system.

Table 18. Character Spoken As Values

Speech Type	Option	Notes
Recorded	Crmf	Speaks the first character with rising inflection, the last character with falling inflection, and all other characters with medial inflection.
	Crmm	Speaks the first character with rising inflection and all other characters with medial inflection.
	Cmmf	Speaks the last character with falling inflection and all other characters with medial inflection.
	Cmmm	Speaks all characters with medial inflection.
	C (default)	Speaks all characters with medial inflection (same as Cmmm).
TTS	AC	Spells out the characters in the string; for example, the character string “Lucent” is spoken as “L -U- C -E -N -T”.
	A (default)	Speaks the character string as words, if possible; for example, the character string “Lucent” is spoken as “Lucent”.

Currency Spoken As Options

Currency variables are actually specialized forms of number variables. All currency variable Spoken As formats begin with N$ (for recorded speech) or AN$ (for TTS). In most cases, this is followed by a D and a number. This combination represents the number of decimal places to assign to the number.

Table 19 shows how the currency input **12397** is spoken using each option.

Table 19. Currency Spoken As Values

Speech Type	Option	Notes
Recorded	N$ (default)	Speaks the number as whole dollars (same as N$D0).
	N$D0	Speaks the number as whole dollars; for example, “Twelve thousand three-hundred ninety-seven dollars”.
	N$D2	Speaks the number as dollars and cents (treating the last two digits as decimal places); for example, “One hundred twenty-three dollars and ninety-seven cents”.
Recorded (cont.)	N$D1	Speaks the number as dollars and tenths of dollars; for example, “One thousand two-hundred thirty-nine point seven dollars”.
	N$D3	Speaks the number as dollars and thousandths of dollars; for example, “Twelve point three nine seven dollars”.
	N$D4	Speaks the number as dollars and ten-thousandths of dollars; for example, “One point two three nine seven dollars”.
	N$D5	Speaks the number as dollars and hundred-thousandths of dollars; for example, “Point one two three nine seven dollars”.
	N$D6	Speaks the number as dollars and millionths of dollars; for example, “Point zero one two three nine seven dollars”.
	N$D7	Speaks the number as dollars and ten-millionths of dollars; for example, “Point zero zero one two three nine seven dollars”.
	N$D8	Speaks the number as dollars and hundred-millionths of dollars; for example, “Point zero zero zero one two three nine seven dollars”.

Continued on next page

Table 19. Currency Spoken As Values — *Continued*

Speech Type	Option	Notes
	N$D9	Speaks the number as dollars and billionths of dollars; for example, “Point zero zero zero zero one two three nine seven dollars”.
TTS	AN$D2	Speaks the number as dollars and cents (treating the last two digits as decimal places); for example, “One hundred twenty-three dollars and ninety-seven cents”.
	AN$D0 (default)	Speaks the number as rounded to the nearest whole dollar; for example, “Twelve thousand three-hundred ninety-seven dollars”.
TTS (cont.)	AN$D1	Speaks the number as dollars and tenths of dollars; for example, “Twelve hundred thirty-nine point seven dollars”.
	AN$D3	Speaks the number as dollars and thousandths of dollars; for example, “Twelve point three nine seven dollars”.
	AN$D4	Speaks the number as dollars and ten-thousandths of dollars; for example, “One point two three nine seven dollars”.
	AN$D5	Speaks the number as dollars and hundred-thousandths of dollars; for example, “Point one two three nine seven dollars”.
	AN$D6	Speaks the number as dollars and millionths of dollars; for example, “Point zero one two three nine seven dollars”.

Continued on next page

Table 19. Currency Spoken As Values — *Continued*

Speech Type	Option	Notes
	AN$D7	Speaks the number as dollars and ten-millionths of dollars; for example, “Point zero zero one two three nine seven dollars”.
	AN$D8	Speaks the number as dollars and hundred-millionths of dollars; for example, “Point zero zero zero one two three nine seven dollars”.
	AN$D9	Speaks the number as dollars and billionths of dollars; for example, “Point zero zero zero zero one two three nine seven dollars”.

Date Spoken As Options

Date formats always begin with a D (for recorded speech) or AD (for TTS). These are appended by a combination of the following:

- M – The month spoken as a number
- MSP – The month spoken by name
- D – The day (number)
- Y – The year expressed as a two-digit number
- YY – The year expressed as a four-digit number

Table 20 shows how the date input **112397** is spoken using each option.

Table 20. Date Spoken As Values

Speech Type	Option	Notes
Recorded	DMDYY	“Eleven (pause) twenty-three (pause) nineteen ninety-seven”
	DMDY	“Eleven (pause) twenty-three (pause) ninety-seven”
	DMD	“Eleven (pause) twenty-three”

Continued on next page

Table 20. Date Spoken As Values — *Continued*

Speech Type	Option	Notes
	DMSPDYY	“November twenty-third (pause) nineteen ninety-seven”
	DMSPDY	“November twenty-third (pause) ninety-seven”
	DMSPD	“November twenty-third”
	D (default)	Same as DMSPDY
TTS	ADMDYY (default)	“Eleven (pause) twenty-three (pause) nineteen ninety-seven”
	ADMSPDYY	“November twenty-third (pause) nineteen ninety-seven”
	ADMDY	“Eleven (pause) twenty-three (pause) ninety-seven”
	ADMSPDY	“November twenty-third (pause) ninety-seven”
	ADMD	“Eleven (pause) twenty-three”
	ADMSPD	“November twenty-third”

Number Spoken As Options

Number variables are spoken as numbers, with the specified inflection, if any. A negative number is spoken with the word “minus” in front of the number.

Voice@Work does not support speaking numbers in the billions or trillions, because those numbers are generally outside the range of a number variable (numbers greater than 2,147,483,647 and less than -2,147,483,648).

The phrases “billion” and “trillion”, however, are included in the Enhanced Basic Speech (EBS) packages.

If your application requires speaking numbers in the billions or trillions, you can:

1. Write an external function that accepts a larger number in the form of an ASCII string.
2. Parse the string (using a Set and Test node) getting the amounts of billions and trillions as substrings.
3. Convert the resulting substrings to integer values.

4. Speak the resulting integer values, inserting the phrases for “billion” or “trillion” where appropriate.

Table 21 shows how the number input **1522** is spoken using each option.

Table 21. Number Spoken As Values

Speech Type	Option	Notes
Recorded	Nrmf	“One thousand five hundred twenty-two” (The “One” is spoken with rising inflection, the “two” with falling inflection, and all other digits with medial inflection.)
	Nrmm	“One thousand five hundred twenty-two” (The “One” is spoken with rising inflection and all other digits with medial inflection.)
	Nmmf	“One thousand five hundred twenty-two” (The “two” is spoken with falling inflection and all other digits with medial inflection.)
	Nmmm	“One thousand five hundred twenty-two” (All digits are spoken with medial inflection.)
	N (default)	Same as Nmmm.
	NX	Speaks the phrase having this number. (For example, for the number 1522, phrase number 1522 from the designated phrase table is spoken, not the number.) **NOTE:** If you select the NX option, you must also designate a phrase table in the Phrase Table column.
TTS	AN (default)	Speaks the number as a number (“One thousand five hundred twenty-two”)
	AC	Speaks the number as individual digits (“One five two two”)

Real Spoken As Options

Real variables are actually specialized forms of number variables. For this reason, all real variable Spoken As formats begin with N, followed by a D and a number. This combination represents the number of decimal places to assign the number. The decimal point is spoken as “point”.

Table 22 shows how a literal real input of **123.97** is spoken using each option.

Table 22. Real Spoken As Values

Speech Type	Option	Notes
Recorded	ND	Speaks the number as rounded to the nearest whole number; for example, “One hundred twenty-four”
	ND1 (default)	Speaks the number rounded to the nearest tenth; for example, “One hundred twenty-four point zero”
	ND2	Speaks the number rounded to the nearest hundredth; for example, “One hundred twenty-three point nine seven”
	ND3	Speaks the number rounded to the nearest thousandth; for example, “One hundred twenty-three point nine seven zero”
	ND4	Speaks the number rounded to the nearest four decimal places; for example, “One hundred twenty-three point nine seven zero zero”
	ND5	Speaks the number rounded to the nearest five decimal places; for example, “One hundred twenty-three point nine seven zero zero zero”
	ND6	Speaks the number rounded to the nearest six decimal places; for example, “One hundred twenty-three point nine seven zero zero zero zero”
	ND7	Speaks the number rounded to the nearest seven decimal places; for example, “One hundred twenty-three point nine seven zero zero zero zero zero”

Continued on next page

Table 22. Real Spoken As Values — *Continued*

Speech Type	Option	Notes
	ND8	Speaks the number rounded to the nearest eight decimal places; for example, “One hundred twenty-three point nine seven zero zero zero zero zero zero”
	ND9	Speaks the number rounded to the nearest decimal places; for example, “One hundred twenty-three point nine seven zero zero zero zero zero zero zero”
TTS	AND1 (default)	Speaks the number rounded to the nearest tenth; for example, “One hundred twenty-four point zero”
	AND2	Speaks the number rounded to the nearest hundredth; for example, “One hundred twenty-three point nine seven”
	AND3	Speaks the number rounded to the nearest thousandth; for example, “One hundred twenty-three point nine seven zero”
	AND4	Speaks the number rounded to the nearest four decimal places; for example, “One hundred twenty-three point nine seven zero zero”
	AND5	Speaks the number rounded to the nearest five decimal places; for example, “One hundred twenty-three point nine seven zero zero zero”
	AND6	Speaks the number rounded to the nearest six decimal places; for example, “One hundred twenty-three point nine seven zero zero zero zero”

Continued on next page

Table 22. Real Spoken As Values — *Continued*

Speech Type	Option	Notes
TTS (cont.)	AND7	Speaks the number rounded to the nearest seven decimal places; for example, “One hundred twenty-three point nine seven zero zero zero zero zero”
	AND8	Speaks the number rounded to the nearest eight decimal places; for example, “One hundred twenty-three point nine seven zero zero zero zero zero zero”
	AND9	Speaks the number rounded to the nearest nine decimal places; for example, “One hundred twenty-three point nine seven zero zero zero zero zero zero zero”

Time Spoken As Options

There is only one Spoken As format for time variables. This format speaks the variable using a 12-hour clock and the abbreviations “AM” or “PM”.

For example, a military time of 14:30 is spoken as “Two-thirty (pause) P M”.

Table 23 describes the Time Spoken As options.

Table 23. Time Spoken As Options

Speech Type	Option	Notes
Recorded	THMAM	Speaks the variable using a 12-hour clock and the abbreviations “AM” or “PM”
	T (default)	Same as THMAM
TTS	ATHMAM (default)	Speaks the variable using a 12-hour clock and the abbreviations “AM” or “PM”

Speak Using Column

The displayed value tells what method will be used by the target voice response system to speak the variable. Table 24 describes the types of speech available.

NOTE:
Text-to-Speech (TTS) options are available and appear only for languages that support TTS. Note that you can only use TTS in Voice@Work if you have purchased it as an option for your INTUITY CONVERSANT system. In most cases, you will probably find it preferable to use Recorded Speech.

Table 24. Speech Types Available from the Speak Using Column

Speech Type	Comments
Recorded	Digitally recorded spoken speech
tts (Text-to-Speech)	Synthesized speech produced according to predetermined algorithms

To change a Speak Using attribute to another value, click in the field for the variable you want to change and select the type of speech you want from the pulldown menu.

Phrase Table Column

A value only appears in the Phrase Table column of the Variables Manager if the Type of variable is number, the Speak Using speech type is “recorded,” and the Spoken As value is NX. Otherwise, this field is empty.

When you select number as the Type, “recorded” as the Speak Using speech type, and NX as the Spoken As value, Voice@Work populates this field with the Phrase Table that is currently active.

To set or reset the values in the **Phrase Table** field, click in the field and select the Phrase Table you want from the pulldown menu.

Input As Column

The value that appears in the Input As column of the Variables Manager depends on what type of variable it is and what value you set for it.

When data is received into the variable, the format you specify for the field must match the way the data is input.

When you select a new Type for the variable, Voice@Work automatically assigns it a default value in the Input As column (Table 17).

Table 25 describes the options available for each variable type.

Table 25. Input As Column Options

Variable Type	Default	Other Options	Comments
Character	char		Accepts all standard ASCII characters as input. Stores input as a string without making any changes to it.
Currency	#*00		Accepts numerals, dollar sign ($), minus sign (-), decimal point (.), and commas (,). If no characters are input after a decimal point (or no decimal point exists), Voice@Work automatically assigns zeroes.
Date	mmddyy	yyyymmdd	Accepts numerals, hyphens (-), slashes (/), and spaces. The option you select here determines how incoming data is interpreted by the target system.

Continued on next page

Table 25. Input As Column Options — *Continued*

Variable Type	Default	Other Options	Comments
Number	#		Accepts only numerical input and disregards any nonnumeric characters, such as commas or hyphens. The only exception is a leading minus sign (-), which indicates a negative number. All input is treated as integer values. **NOTE:** If you want to preserve hyphens or other nonnumeric characters within a number–for example, social security numbers or phone numbers–use the Character variable type.
Real	#*#		Accepts numerals, decimal points (.), and minus signs (-). May be input as any number of digits before and after the decimal point.
Time	hhmm(AP)M	hhmm hh24mm hhmmss	Accepts numerals, colons (:), decimal points (.), and the letters “A”, “P”, and “M”. Strips out characters other than numerals and the letters “A”, “P”, and “M” when storing values.

To set or reset the values in the **Input As** field, click in the field and select the option you want from the pulldown menu.

Length Column

The purpose of the Length column is to set the maximum length that the selected variable can have. If the length of the variable input is greater than this value, it is truncated to fit within the allocated space.

The value that appears in the Length column depends on what type of variable it is and what value you set.

When you select a new Type for the variable, Voice@Work automatically assigns it a default value in the Length column (Table 17).

You can enter any integer value you want in the **Length** field. (It may not always make good sense to do so, however.) Table 26 describes the default values for each Type of variable.

Table 26. Length Column Default Values

Variable Type	Default Value	Comments
Character	24	
Currency	15	When setting this value, you must include the decimal point (.), the dollar sign ($), and the minus sign (-).
Date	8	This value cannot be reset by the user.
Number	10	Values may range from -2,147,483,648 to 2,147,483,647.
Real	10	When setting this value, you must include the decimal point (.) and the minus sign (-).
Time	6	This value cannot be reset by the user.

To set or reset the values in the **Length** field, click in the field and type in the new value you want.

NOTE:
You cannot edit or reset the Length values for system variables.

Default Column

The value that appears in the Default column depends on what type of variable it is and what value you set for it.

When you select a new Type for the variable, Voice@Work automatically assigns it a default value in the Default column (Table 17).

Table 27 describes the default values for the Default column for each variable type.

Table 27. Default Column Default Values

Variable Type	Default Value	Comments
Character		All character values in the standard ASCII character set are accepted, including spaces.
Currency	0.00	Only numerical values are accepted. All decimal values are rounded to the nearest cent. The value is always displayed as dollars and cents. Any characters other than numerals (including spaces) are ignored and removed.
Date	19980101	Any new value must be a valid date using the *yyyymmdd* format. If you attempt to enter an invalid date value, Voice@Work rejects it and returns the field value to the last valid value that was entered into the field.
Number	0	Only numerical values and leading minus symbols are accepted. A decimal point is ignored in any value containing one (such as 234.56).

Continued on next page

Table 27. Default Column Default Values — *Continued*

Variable Type	Default Value	Comments
Real	0.0	Only numerical values and the decimal point and minus sign are accepted. All values are rounded to the number of places specified in the Spoken As column and displayed with the appropriate number of decimal places. If you attempt to enter an invalid (non-numeric) value, Voice@Work rejects it and returns the field value to the last valid value that was entered into the field.
Time	000000	Any new value must be a valid time using the *hhmmss* format. If you attempt to enter an invalid time value, Voice@Work rejects it and returns the field value to the last valid value that was entered into the field.

To set or reset the values in the Default column, click in the field for the desired variable and type in the new value you want.

System Column

If the word “true” appears in this column, the selected variable is defined as a “system variable.” For more information about system variables, see “System Variables” below.

Creating New (Custom) Variables

To create a new (custom) variable, do the following:

1. Open the Variables Manager.
2. Right-click in the Variables Manager somewhere *other* than on the name bar.
3. Select **New...** from the popup menu.

   Voice@Work displays a dialog box and prompts you for a name for the new variable (Figure 44).

Figure 44. The Name New Variable Window

4. Enter the name you want for the variable in the space provided.

   When naming variables, keep the following in mind:

   - Names can be any length. As a matter of practicality, however, you should try to keep variable names as short as possible.
   - Do not use spaces in names. Voice@Work ignores and deletes spaces when creating the variable name.
   - Variable names are case sensitive. Voice@Work preserves cases as you assign them when naming the variable. The one exception is that the first letter of a name is always converted to upper case, regardless of how you type it in. For example, if you typed in **new_ Var** as the variable name, Voice@Work converts it to **New_Var** when creating the variable.

     NOTE:
     Although Voice@Work preserves cases when saving names and transfers them to the target system with cases preserved, you cannot have two names with the same letters. If, for example, you have a variable called **New_Var**, you cannot also create a variable called **NEW_VAR**.

   - Variable names should begin with a letter only. Voice@Work ignores and deletes all other initial characters when creating the variable name.

     NOTE:
     You may use numbers or the underscore (_) symbol after the initial character of a variable name.

5. Click **OK**.

   The new variable appears in the Variables Manager.

   When you create a new variable, the Variables Manager does the following:

   - Places the new variable (along with all other variables) in the grid in alphabetical order by name.
   - Gives the new variable the following default values:

     Value (blank)
     Type character
     Spoken As char
     Speak Using (see Note below)
     Phrase Table (blank)
     Input As char
     Length 24
     Default (blank)
     System (blank)

     NOTE:
     The Speak Using default value depends on the language you have selected: If the application language supports **tts**, **tts** is selected as the default. If the application language does *not* support TTS, **recorded** is selected as the default.

6. Set the attributes for the new variable as desired. See "Setting or Editing a Variable's Attributes" below for the procedure.

Renaming Custom Variables

NOTE:
You can only rename custom variables. System variables cannot be renamed (see "System Variables" below).

To rename a custom variable, do the following:

1. Open the Variables Manager.
2. Right-click the name of the variable whose name you want to change.
3. Select **Rename...** from the popup menu.

   Voice@Work displays a dialog box and prompts you for a new name for the variable (Figure 45).

Figure 45. The Rename Variable Window

4. Enter the new name you want for the variable in the space provided. For guidelines on naming variables, see Step 4 in "Creating New (Custom) Variables" above.
5. Click **OK**.

   The name of the variable changes in the Variables Manager and anywhere it is referenced within the application.

Setting or Editing a Variable's Attributes

NOTE:
Some system variable attributes (such as Type) cannot be edited.

To set or edit a variable's attributes, do the following:

1. Open the Variables Manager.
2. Click in the space of the attribute you want to set or edit.
3. Do one of the following:
   - Enter the value you want for the attribute.
   - Select the value you want for the attribute from the pulldown menu, where available.

System Variables

System variables are variables that are predefined and included as part of the Voice@Work package.

Using System Variables

System variables are displayed whenever you open the Variables Manager. They are identified by the word “True” in the System column. They can be used in much the same way as any custom variable.

System variables cannot be renamed, and some of their attributes (such as Type) cannot be edited by the user.

System variables are used mostly behind-the-scenes or as part of prompts. Behind-the-scenes use, for example, includes using a Time Branch node to direct a call through the call flow. In prompts, use of variables is typically defined using the **Prompts** tabs.

System Variable Descriptions

Table 28 describes the system variables included with Voice@Work.

Table 28. System Variables

Variable Name	Description
ANI (Automatic Number Identification)	Stores the calling party ID number, if available.
Attempts	Stores the number of times the caller tries to respond to a prompt.
Channel	Channel number on which the current call is being handled.
CI_RECOG	Number, set by the system, representing the recognizer used to identify a caller response.
CollectedDigits	Value of the digit or digits pressed or spoken by the caller in responding to a prompt.
Date	Current date in *yyyymmdd* format.
Default	24-character string variable; available for general use by application developers.

Continued on next page

Table 28. System Variables — *Continued*

Variable Name	Description
DNIS (Dialed Number Identification Service)	Stores the calling party ID number dialed, if available.
Errno	Number variable used for return values.
FindBest	Character variable used to help improve recognition accuracy in applications that use DPR or speech recognition.
NoDigit	Value of the number for a caller to press or speak signifying a “no” response.
PivotYear	Number used to help convert character variables to date variables; used to assign the date to the correct century.
RecordCounter	Used as a counter to track the number of records that match selected criteria. Each number in the final set represents a different record of all those found.
RecordsFound	Increments by 1 each time a record is found. The final value of the number is the total number of records found.
Time	Current time in *hh24mmss* format.
Tries	Represents the number of times to allow a caller to attempt a response to a prompt (Default = 3).
YesDigit	Value of the number for a caller to press or speak signifying a “yes” response.

Right-Click Menu Options for the Variables Manager

Table 29 describes the right-click menu options available in the Variables Manager.

Table 29. Variables Manager Right-Click Menu Options

Option	**Description/Action**
New...	Allows you to create a new (custom) variable; see "Creating New (Custom) Variables" above.
Delete	Deletes the selected variable from the Variables Manager.
Rename...	Allows you to rename the selected custom variable without altering its attributes; see "Renaming Custom Variables" above.
Find Uses of	Opens the Find window (see "The Find... Feature" in Chapter 6, "The Edit Menu"), searches for Uses Of the selected variable, and presents the results of the search in the **Find window results** field.
Sharing...	Inactive option in the Variables Manager.
Stay on Top	(Available only when the Variables Manager is undocked) Toggles on and off. When checked (toggled on), causes the Variables Manager window to remain on top of other open Voice@Work windows. **NOTE:** If you have more than one resource manager window toggled to "Stay on Top", the window actually on top will be the last window you clicked in. All windows toggled to "Stay on Top", however, will stay on top of any other Voice@Work windows that are open.
Use TTS only	When checked, changes all values in the Speak Using column to "tts" (Text-to-Speech). **NOTE:** Checking this box also changes all Spoken As formats to the appropriate **tts** defaults.
Use Recorded only	When checked, changes all values in the Speak Using column to "recorded" (Recorded Speech). **NOTE:** Checking this box also changes all Spoken As formats to the appropriate **recorded** defaults.

Working with FlexWord Speech Recognition 15

Overview

FlexWord™ speech recognition recognizes words by matching phonemes (parts of words). Individual words are grouped together in word lists, sometimes called grammars. FlexWord lists are grouped together and accessed through the FlexWords Manager.

NOTE:
FlexWord options can only be used if FlexWord speech recognition is available for the application language selected. FlexWord options can only be implemented on the target voice response system if you have purchased the FlexWord speech recognition package for your system.

Purpose

The purpose of this chapter is to provide information about the FlexWords Manager and using FlexWord lists in Voice@Work applications. Topics include:

- The FlexWords Manager
- Importing FlexWord Grammars (Word Lists)
- Creating a Word List
- Renaming a Word List
- Editing a Word List
- Right-Click Menu Options for the FlexWords Manager

The FlexWords Manager

The FlexWords Manager displays the contents of all FlexWords grammars or word lists that are defined for the application.

Opening the FlexWords Manager

To open the FlexWords Manager, do one of the following:

- From the **View** menu, select **FlexWords**.
- Click the  button on the Toolbar.
- Open the Globals Manager and locate the FlexWords Manager (if it is docked there).

Voice@Work displays the FlexWords Manager window (Figure 46).

Figure 46. FlexWords Manager Window

The Name column, on the left, lists all FlexWord grammars defined for the current application, in alphabetic order.

The Word List column, on the right, displays in order all the FlexWord entries in each word list.

Importing FlexWord Grammars (Word Lists)

It is often desirable to use FlexWord lists that you or other developers have created for other applications. Voice@Work allows you to import such FlexWord lists from outside the current application. This saves you the trouble of having to create new FlexWord lists each time you want to use one in your application.

To import a FlexWord lsit into your application, do the following:

1. Open the Application Resources tool. See Chapter 21, "The Application Resources Tool" for the procedure.
2. Click the **Import** tab at the bottom of the window.
3. (Optional) From the pulldown **Drive** menu, select the mapped network drive containing available resources.
4. Select **FlexWord Grammar** from the Application Resources pulldown menu at the top of the window. See Chapter 21, "The Application Resources Tool" for the procedure.

   Voice@Work displays a list of available FlexWord lists and their locations.

   NOTE:
   The FlexWords option is only available if it exists for the language you have selected.

5. Select whether you want to import the resource as **Shared** or as a **Private Copy** by clicking the appropriate radio button.
6. Do one of the following:
   - Highlight the line entry for the FlexWord list you want, then click the **Import** button.
   - Double-click the line entry for the word list you want.

   Voice@Work imports the selected FlexWord list and displays it as an entry in the FlexWords Manager.

Creating a Word List

Voice@Work makes it easy for you to create new FlexWord lists to use in your applications. Once created, these word lists become part of the application.

NOTE:
The FlexWord option is only available if it exists for the language you have selected.

To create a new word list, do the following:

1. Open the Flexwords Manager. See “Opening the FlexWords Manager” above for the procedure.
2. Right-click in the Flexwords Manager somewhere *other* than on the name bar.
3. Select **New...** from the popup menu.

   Voice@Work displays a dialog box and prompts you for a name for the new word list (Figure 47).

Figure 47. Name New FlexWord Grammar Window

4. Enter the name you want for the FlexWord list in the space provided.

   NOTE:
   When naming FlexWord lists, keep in mind that the name should be unique in the first 14 characters. If not, any FlexWord list on the target system with a name that matches the first 14 characters will be overwritten during installation of the application.

5. Click **OK**.

   Voice@Work displays the FlexWord Editor window (Figure 48).

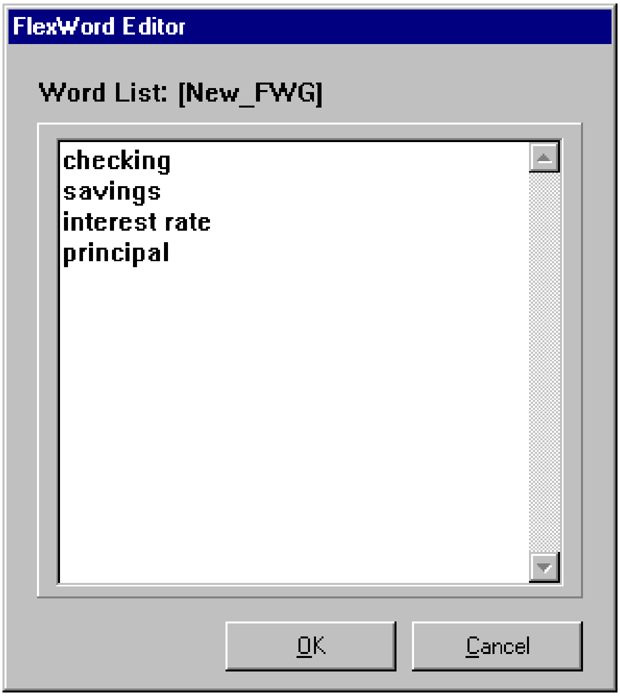


Figure 48. FlexWord Editor Window

6. Enter your FlexWord list in the **Word List** field (Figure 48).
7. When you are finished adding words, click **OK**.

   Voice@Work displays the new FlexWord list in the FlexWords Manager.

Renaming a Word List

To rename a word list, do the following:

1. Open the Flexwords Manager. See "Opening the FlexWords Manager" above for the procedure.
2. Right-click on the name of the FlexWord list you want to rename.
3. Select **Rename...** from the popup menu.

   Voice@Work displays a dialog box and prompts you for a new name for the word list (Figure 49).

Figure 49. Rename Word List Window

4. Enter the new name you want for the word list in the space provided.
5. Click **OK**.

   The name of the FlexWord list changes in the FlexWords Manager window.

Editing a Word List

To edit a word list, do the following:

1. Open the Flexwords Manager. See "Opening the FlexWords Manager" above for the procedure.
2. Right-click on the name of the FlexWord list you want to edit.
3. Select **Edit FlexWord Grammar...** from the popup menu.

   Voice@Work displays the FlexWord Editor window (Figure 48).
4. Edit your FlexWord list in the **Word List** field (Figure 48).
5. When you are finished editing the word list, click **OK**.

   The FlexWord Editor disappears, and the new FlexWord list appears in the FlexWords Manager window.

Right-Click Menu Options for the FlexWords Manager

Table 30 describes the options available from the FlexWords Manager right-click popup menu.

Table 30. FlexWords Menu Options

Menu Option	Description/Action
New...	Allows you to create a new FlexWord list; see “Creating a Word List” above.
Delete	Deletes the FlexWord list from the FlexWords Manager, if it is not in use in the application. **NOTE:** If you have used the FlexWord list in your application, you must first remove its use from the application before you can delete it.
Rename...	Allows you to rename the selected FlexWord list; see “Renaming a Word List” above.
Find Uses of	Opens the Find window (see “The Find... Feature” in Chapter 6, “The Edit Menu”), searches for Uses Of the selected FlexWord list, and presents the results of the search in the **Find window results** field.
Sharing...	Allows you to designate a FlexWord list as “Shared”, so that you or other developers can use it in other applications. When a FlexWord list is marked as “Shared”, it can be imported into other applications using the Application Resources tool (see Chapter 21, “The Application Resources Tool” for additional information).
Stay on Top	(Available only when the FlexWords Manager is undocked) Toggles on and off. When checked (toggled on), it causes the FlexWords Manager window to remain on top of other open Voice@Work windows. **NOTE:** If you have more than one resource manager window toggled to “Stay on Top”, the window actually on top will be the last window you clicked in. All windows toggled to “Stay on Top”, however, will stay on top of any other Voice@Work windows that are open.
Edit FlexWord Grammar...	Allows you to edit the FlexWord list; see “Editing a Word List” above.

Working with External Functions

16

Overview

External Functions provide an interface for cases requiring special handling in an application. In such cases, you can either import a predefined (standard) external function node provided with Voice@Work using the Application Resources tool, or create your own external function node to do what you need.

External functions are grouped together and accessed through the External Functions Manager.

Purpose

The purpose of this chapter is to provide information about using external functions in Voice@Work. Topics include:

The External Functions Manager

Most of the actions you can perform with external functions are done from the External Functions Manager.

The External Functions Manager displays the names of and brief descriptions for all external functions defined for the current application.

NOTE:
The actual external function resides on the target voice response system — and not in Voice@Work. To make use of an external function residing on the target system, you must create (or import) an “external function node” to act as an interface with the external function itself. In this book, however, the external function nodes are referred to simply as “external functions.”

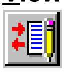
Lucent Technologies has made available, as part of the INTUITY™ CONVERSANT® system and Voice@Work packages, a set of commonly used standard external functions for your use. These in effect, are “shared-out” by Lucent so that you can import them into (“share them in”) your application.

If you want to create your own custom external functions, you can use the External Function Source Editor to create them (see “The External Function Source Editor” section in this chapter), or you can create them directly on the target system. For information about creating external functions on the target system, see the *INTUITY CONVERSANT System Application Development with Script Builder* book for your system.

Once the external function resides on the target system, you must know the name of that external function before you can create the external function node in Voice@Work to call it.

Opening the External Functions Manager

To open the External Functions Manager, do one of the following:

- From the **View** menu, select **External Functions**.
- Click the  button on the Toolbar.
- Open the Globals Manager and locate the External Functions Manager (if it is docked there).

Voice@Work displays the External Functions Manager window (Figure 50).

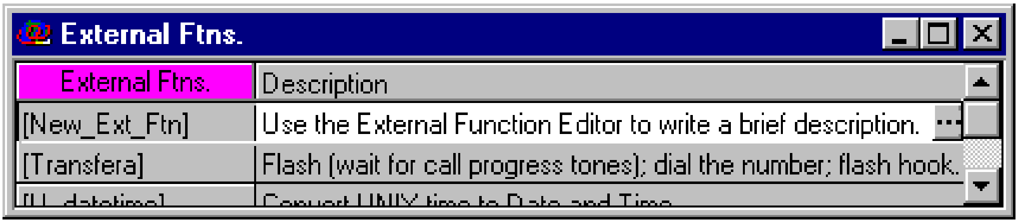


Figure 50. External Functions Manager Window

The left column displays the names of all external functions defined for the current application, listed in alphabetic order.

The right column displays a brief description of the external function.

NOTE:
This description can only be typed in using the External Function Editor (see "The External Function Editor" below).

Importing External Functions

The Voice@Work package includes a variety of standard external functions. It is often desirable to use these external functions or others that you or other developers have created. Voice@Work allows you to import such external functions from outside the current application. This can save you the trouble of having to create new external functions each time you want to use one in your application.

To import an external function, do the following:

1. Open the Application Resources tool. See Chapter 21, "The Application Resources Tool" for the procedure.
2. Select **External Functions** from the Application Resources **Options** menu See Chapter 21, "The Application Resources Tool" for the procedure.

   Voice@Work displays a list of available external functions and their locations.
3. Do one of the following:
   - Highlight the line entry for the external function you want and click the **Import** button.
   - Double-click the line entry for the external function you want.

   Voice@Work imports the selected external function and displays it as an entry in the External Functions Manager.

Creating New External Functions

When no external functions exist that perform the action you need, you can use Voice@Work to create new external functions for use in applications.

To create a custom external function, do the following:

1. Open the External Functions Manager. See "Opening the External Functions Manager" above for the procedure.
2. Right-click in the External Functions Manager somewhere *other* than on the name bar.
3. Select **New...** from the popup menu.

   Voice@Work displays a dialog box and prompts you for a name for the new external function (Figure 51).

Figure 51. The Name New External Function Window

4. Enter the name you want for the external function in the space provided.

   When naming external functions, keep the following in mind:

   - There is no limit on the length for external function names, but as a matter of practicality, you should keep them as short as possible.
   - Do not use spaces in names. Voice@Work ignores and deletes spaces when creating the external function name.

- External function names are case sensitive. Voice@Work preserves cases as you assign them when naming the external function. The one exception is that the first letter of a name is always converted to upper case, regardless of how you type it in. For example, if you typed in **new_External_Function** as the external function name, Voice@Work converts it to **New_External_Function** when creating the external function.

  NOTE:
  Although Voice@Work preserves cases when saving names and transfers them to the target system with cases preserved, you cannot have two names with the same letters. If, for example, you have an external function called **New_External_Function**, you cannot also create an external function called **NEW_EXTERNAL_FUNCTION**.

- External function names must begin with an alphanumeric character. Voice@Work ignores and deletes all other initial characters when creating the external function name.

  NOTE:
  You may use the underscore (_) symbol after the initial character of a external function name.

5. Click **OK**.

   Voice@Work displays the External Function Editor window (Figure 52).

6. Use the External Function Editor to define the attributes for the new external function. See "The External Function Editor" below.

7. Click **OK**.

   The External Function Editor closes and the new external function name appears as an entry in the External Functions Manager.

Renaming an External Function

To rename an external function, do the following:

NOTE:
If you want to rename an external function defined as a "system function" (such as the standard external functions that come packaged with Voice@Work), you must first uncheck the "System Function" checkbox in the External Function Editor. For details about defining external functions as "system functions," see "The External Function Editor" below.

1. Open the External Functions Manager. See "Opening the External Functions Manager" for the procedure.

2. Right-click on the name of the external function you want to rename.
3. Select **Rename...** from the popup menu.

   Voice@Work displays a dialog box and prompts you for a new name for the external function (Figure 51).
4. Enter the new name you want for the external function in the space provided. Keep in mind the external function name conventions. See Step 4 in "Creating New External Functions" above.
5. Click **OK**.

   The name of the external function changes in the External Functions Manager window.

Editing an External Function

To edit an external function, do the following:

1. Open the External Functions Manager. See "Opening the External Functions Manager" above for the procedure.
2. Right-click on the name of the external function you want to edit.
3. Select **Edit External Function** from the popup menu.

   Voice@Work displays the External Function Editor window (Figure 52).
4. Edit the External Function Definition as needed. See "The External Function Editor" below.
5. Click **OK**.

Right-Click Menu Options for the External Functions Manager

Table 31 describes the actions performed by each item on the External Functions Manager right-click popup menu.

Table 31. External Functions Right-Click Menu Options

Menu Option	Description
New...	Allows you to create a new external function; see "Creating New External Functions" above.
Delete	Deletes the external function definition from the External Functions Manager, if it is not in use in the application.
Rename...	Allows you to rename the selected external function without altering its definition; see "Renaming an External Function" above.
Find Uses of	Opens the Find window, searches for Uses Of the selected external function definition, and presents the results of the search in the **Find window results** field; see "The Find... Feature" in Chapter 6, "The Edit Menu".
Sharing...	Allows you to designate a custom external function as Shared, so that you or other developers can use it in other applications. When an external function is marked as Shared, it can be imported into other applications using the Application Resources tool. See Chapter 21, "The Application Resources Tool" for additional information.
Stay on Top	(Available only when the External Functions Manager is undocked) Toggles on and off. When checked (toggled on), it causes the External Functions Manager window to remain on top of other open Voice@Work windows. **NOTE:** If you have more than one resource manager window toggled to "Stay on Top", the window actually on top will be the last window you clicked in. All windows toggled to "Stay on Top", however, will stay on top of any other Voice@Work windows that are open.
Edit External Function Definition	Allows you to edit the selected external function definition; see "The External Function Editor" below.

The External Function Editor

The External Function Editor allows you to define and edit the attributes of any external function defined for the application.

From the External Function Editor, you can:

- Type in a brief description of the external function for display in the External Functions Manager.
- Name or rename the source file for the external function.
- Define the external function as a system function.
- Set the Function Arguments attributes.
- Set the Function Return Values attributes and define custom return branches.
- Define the external function’s appearance on the External Functions palette.
- Use a right-click popup menu to perform various operations in the External Function Editor.

Opening the External Function Editor

The External Function Editor is accessed through the External Functions Manager.

To open the External Function Editor, do the following:

1. Open the External Functions Manager. See “Opening the External Functions Manager” above for the procedure.
2. Do one of the following:
   - Create and name a new external function. See “Creating New External Functions” above for the procedure.
   - Right-click in the External Functions Manager window on the name of the external function you want to edit, and then select **Edit External Function Definition...** from the popup menu.
   - Double-click the name of the external function you want to edit.

   Voice@Work displays the External Function Editor window (Figure 52).

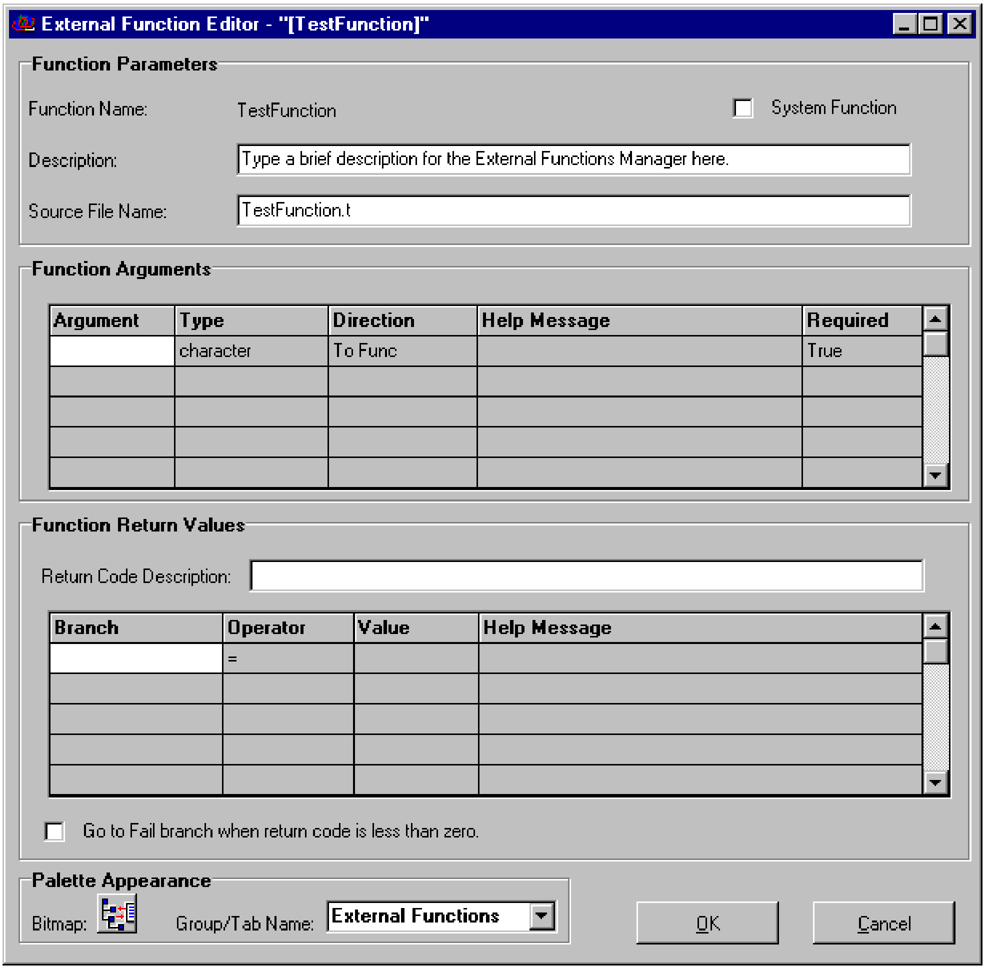


Figure 52. External Function Editor

Table 32 describes the elements of the External Function Editor.

Table 32. External Function Editor Elements

Element	Description/Comments
Function Name	Displays the name of the external function being defined or edited.
System Function checkbox	When checked, it defines the external function as a system function; see "Setting the External Function Parameters" below.
Description field	(Optional) Allows you to type in a brief description of the external function for display in the External Functions Manager.
Source File Name	Displays the filename of the source file for the external function residing on the target voice response system Typically, this is the same as the external function name, with a **.t** extension.
Function Arguments box	(Optional) Allows you to define the function arguments for the external function; see "Setting the Function Arguments Attributes" below.
Function Return Values box	(Optional) Allows you to define function return values for the external function; see "Setting the Function Return Values Attributes (Optional)" below.
Palette Appearance box	Allows you to select a bitmap image for use on the External Functions Palette and assign the external function to a group or tab on that palette.

Using the External Function Editor

The External Function Editor is used to define and set various parameters for an external function. These parameters are discussed in the following sections.

Setting the External Function Parameters

The External Function Parameters section includes fields for identifying the external function and controlling its attributes.

System Function Status (Checkbox)

When checked, this parameter causes the external function to be defined as a system function.

NOTE:
When defined as a system function, you cannot edit most attributes of the external function. This includes the function name, description, arguments, and most return value attributes.

The predefined external functions provided with Voice@Work are all defined as system functions.

To customize system functions, you must uncheck this box. You should then rename the external function.

Defining an external function as a system function after you have defined it in the External Function Editor is a good way to protect the external function definition from inadvertently being overwritten on the PC.

Description Field

Type in a brief description of the external function. Include what it is used for, what arguments it takes, and so forth. This description is displayed in the External Functions Manager **Description** field.

NOTE:
This description can only be typed in and edited in the External Function Editor.

Source File Name Field

This parameter displays the filename of the source file for the external function residing on the target voice response system.

Typically, this is the same as the external function name, with a **.t** extension.

NOTE:
The external function *must* have the **.t** extension to be recognized by a target voice response system.

This source file name is automatically assigned whenever you create or import an external function and in most cases should not be changed.

Setting the Function Arguments Attributes

Function arguments determine how the target system uses the external function in the application.

Argument Field

Type in the name of the argument as you want it to display in the Node Inspector window.

When naming external function arguments, keep the following in mind:

- Names are for display purposes only (in the Node Inspector window) and can be any length. However, keep argument names as short as possible.
- Do not use spaces in external function argument names. Voice@Work strips out spaces when creating the external function argument.
- Argument names are case sensitive. Voice@Work preserves cases as you assign them when naming the argument.

Type field

Select from the pulldown menu which type you want to assign to the argument. Table 33 describes the Type options.

Table 33. External Function Argument Types

Type	Description
Character	Accepts as input all standard ASCII characters, including special characters and punctuation; the extended ASCII character set is not supported.
Currency	Accepts as input numerals (0–9), dollar signs, minus signs, and decimal points.
Date	Accepts as input numerals (0–9), minus signs, slashes, and spaces.
Number	Accepts as input numerals (0–9) and minus signs.
Real	Accepts as input numerals (0–9), minus signs, and decimal points.
Time	Accepts as input numerals (0–9) and the letters A, P, and M.

Direction Field

The **Direction** field determines which way the external function will direct the argument.

Select from the pulldown menu one of the following options:

- **To Func** — The argument is passed to the external function.
- **From Func** — The argument is received from the external function.
- **To/From Func** — The argument is either passed to or received from the external function or both.

Help Message Field (Optional)

Type in a message that will appear as a popup “balloon” message when the cursor rests over the argument in the Node Inspector window.

When writing popup help messages, keep the following in mind:

- Do not use commas in your message. Voice@Work strips out all commas in popup help messages.
- Keep your messages as short as possible. When a message pops up, it only displays for approximately 2.5 seconds. If your message is too long, the reader will not have time to read it before it disappears.

Required Field

When set to **True**, this field defines this argument as a required argument. This means that a variable or literal value *must* be assigned to it in the Node Inspector window whenever you use this external function in an application. You cannot leave it blank.

Setting the Function Return Values Attributes (Optional)

Voice@Work allows you to create new branches to take when specified values are returned by the external function. Use this section to define those branches and return values.

NOTE:
If you do not define custom branches in this area, Voice@Work sends returns to the default branch.

Return Code Description Field

Type in a brief description of the return code values.

This description appears as a popup “balloon” help message whenever the cursor rests over the **Put Return Code In** cell of the Node Inspector window for the external function. See hints for writing popup help messages in the “Help Message Field (Optional)” of “Setting the Function Arguments Attributes” above.

NOTE:
The following three fields (**Branch**, **Operator**, and **Value**) are used in conjunction with each other.

Branch Field

Type in the name you want to assign to the branch to be taken if the specified value is returned by the external function. When assigned, this creates a new branch on the external function node with the name entered here.

Operator Field

Select from the pulldown menu the operator to be used in evaluating values returned from the external function.

Value Field

Type in a specific value that the external function may return.

Help Message Field

This field is used only to document the return values’ use and is viewed only in the External Function Editor. It does *not* appear anywhere else, such as in a popup help message.

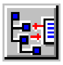
Go to Fail branch... checkbox

When checked, the application goes automatically to the Fail branch whenever a return value is less than zero.

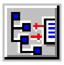
Defining the External Function’s Appearance

In the Palette Appearance box, you can determine how the external function will display on the External Functions Palette (see Chapter 12, “Palettes”).

Bitmap Field

Click the  icon to specify a file containing a 20 x 20 pixel bitmap to be used as the icon for this external function.

Each standard external function included in the Voice@Work package has its own icon which appears on the External Functions palette when you import the external function into your application.

The  icon is the default bitmap icon for External Functions and will be used if no other bitmap image is selected.

NOTE:
Bitmap icons should be exactly 20 x 20 pixels. If you attempt to use larger or smaller bitmaps, Voice@Work resizes the bitmap file to fit in a 20 x 20 pixel space.

Group/Tab Name Field

Use the pulldown menu to assign the external function to a group/tab on the External Functions palette (see Chapter 12, “Palettes”).

This is the only place in Voice@Work you can add new groups/tabs for external functions or assign an external function to a different group/tab.

To create a new group/tab (because the external function does not logically fit within any defined group/tab, for example), clear the field and type in the name for a new group/tab. The new group/tab appears as a separate group/tab on the External Functions palette, and you can further configure the group/tab using the **Configure External Functions Palette...** option (see Chapter 12, “Palettes”).

To assign an external function to a *different* group/tab, select the name of a group/tab from the pulldown menu in the Palette Appearance area of the External Function Editor, or create a new group/tab (see paragraph above).

NOTE:
You cannot change the name of an existing group/tab once it has been created.

You cannot delete a group/tab unless you first delete or move all external functions currently residing in that group/tab. Once you delete or move all external functions from a group/tab, the group/tab is automatically deleted.

You cannot delete the default External Functions group/tab at all.

Right-Click Menu Options for the External Function Editor

The External Function Editor has two different right-click menus. The menu that pops up depends on which area of the External Function Editor you are in.

Right-Click Menu for Fields

This is the menu that appears when you right-click in the **Function Name**, **Description**, **Source File Name**, or **Group/Tab Name** fields. Table 34 summarizes the options on this right-click popup menu.

Table 34. Right-Click Menu Options for Fields

Menu Option	Action
Undo	Undoes the last action done in the selected field.
Cut	Removes highlighted text and places it on the clipboard.
Copy	Makes a copy of highlighted text and places it on the clipboard.
Paste	Pastes whatever is on the clipboard at the selected location.
Delete	Removes highlighted text without placing it on the clipboard.
Select All	Highlights the entire contents of the selected field. **NOTE:** This option is active only if you have first placed the cursor in the field of the text you want to select.

Right-Click Menu for Grids

This is the menu that appears when you right-click in either the Function Arguments or the Function Return Values grids. Table 35 summarizes the options for this right-click popup menu.

Table 35. Right-Click Menu Options for Grids

Menu Option	Action
Insert	Inserts a new argument line just above the selected argument (moving everything else down).
Delete	Deletes the selected argument from the grid. **NOTE:** This action cannot be undone.
Rename...	Renames the selected argument without altering its attributes.
Add Line	Appends a new argument line at the end of all defined argument lines.
Move Up	Moves the selected argument up one line.
Move Down	Moves the selected argument down one line.

The External Function Source Editor

The External Function Source Editor allows you to create an external function for the target system, by writing or editing TAS code from within Voice@Work, rather than having to write it directly on the target system using the **vi** editor or other text editing tool.

Opening and Using the External Function Source Editor

To open and use the External Function Source Editor, do the following:

1. Open the Code Generation/Application Transfer tool window (see Chapter 23, "The Code Generation Tool").
2. Click the  icon on the Code Generation tab.

   Voice@Work displays the External Function Source Editor (Figure 53). Table 36 describes the columns in the External Function Source Editor.

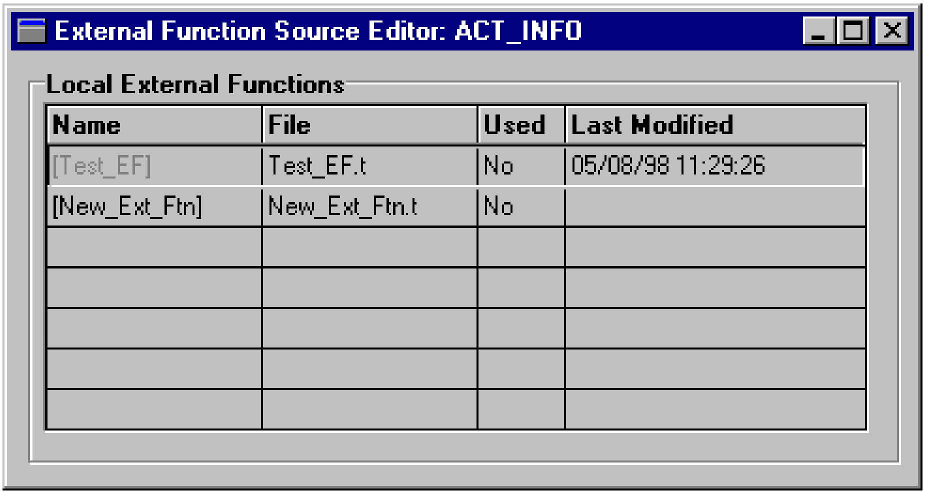


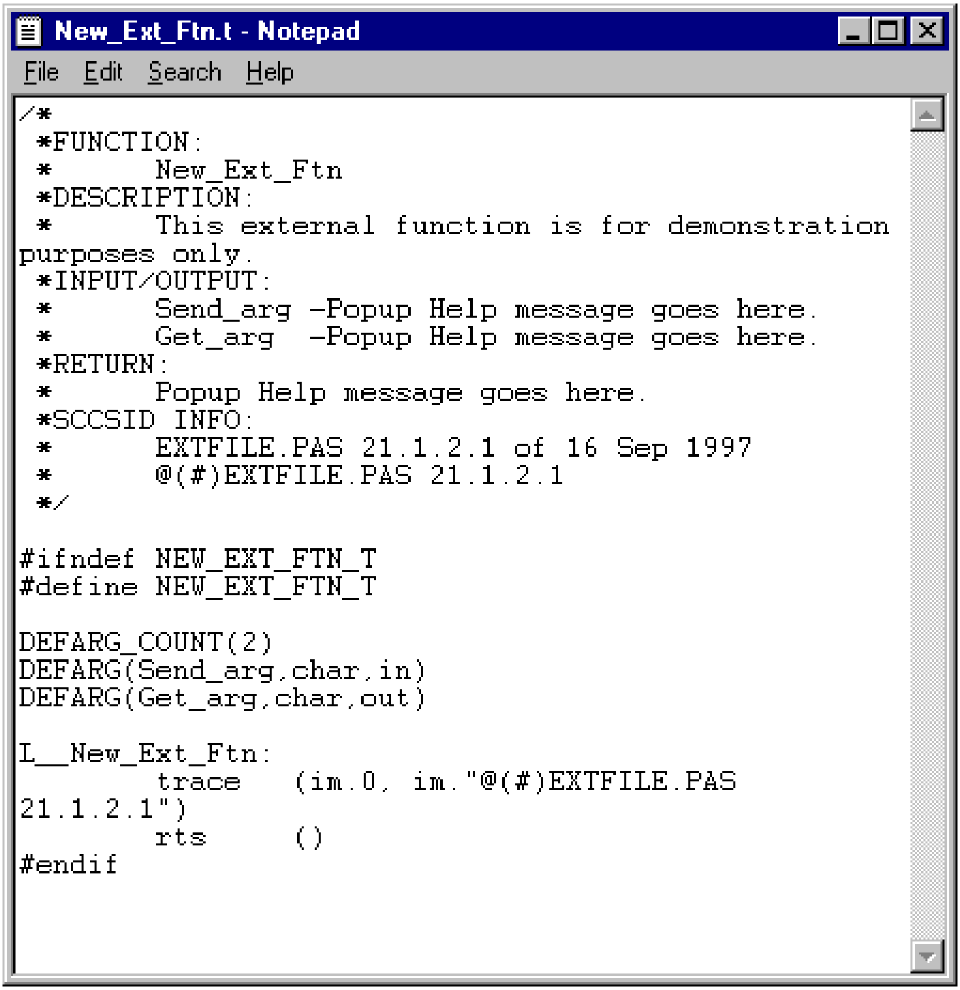
Figure 53. External Function Source Editor

NOTE:
Only external functions *not* defined as system functions are displayed in the External Function Source Editor. Standard external functions and others defined as system functions do not appear.

3. Right-click in the line of the external function you want to create or edit.

NOTE:
Before you can create or edit an external function for the target system using the External Function Source Editor, you must first create the external function definition (see "Creating New External Functions" above).

Notepad opens, allowing you to edit or write your own TAS code for the external function (Figure 54). Voice@Work incorporates this code during the Code Generation process and transfers it to the target system using the Application Transfer tool.




```
/*
 *FUNCTION:
 *       New_Ext_Ftn
 *DESCRIPTION:
 *       This external function is for demonstration
purposes only.
 *INPUT/OUTPUT:
 *       Send_arg -Popup Help message goes here.
 *       Get_arg  -Popup Help message goes here.
 *RETURN:
 *       Popup Help message goes here.
 *SCCSID INFO:
 *       EXTFILE.PAS 21.1.2.1 of 16 Sep 1997
 *       @(#)EXTFILE.PAS 21.1.2.1
 */

#ifndef NEW_EXT_FTN_T
#define NEW_EXT_FTN_T

DEFARG_COUNT(2)
DEFARG(Send_arg,char,in)
DEFARG(Get_arg,char,out)

L__New_Ext_Ftn:
        trace   (im.0, im."@(#)EXTFILE.PAS
21.1.2.1")
        rts     ()
#endif
```

Figure 54. Notepad Display of Sample Code for External Function

NOTE:
Make sure you save your code before closing **Notepad**, or your code will be lost.

Table 36. External Function Source Editor Columns

Name of Column	Description/Comments
Name	This column displays the names of the external function defined for the application. **NOTE:** Only external functions *not* defined as system functions are displayed in the External Function Source Editor. Standard external functions and others defined as system functions do not appear.
File	This column displays the name of the external function's ***.t** file.
Used	This column displays whether or not the external function is used in the current application.
Last Modified	This column displays the date and time the external function was last modified using the External Function Source Editor. If this column field is blank, no external function exists in Voice@Work for the definition. **NOTE:** This column only reflects modifications made to the external function code using the External Function Source Editor. Modifications made to the external function definition in the External Function Source Editor are not reflected here. External functions already residing on the target system are also not reflected here.

Right-Click Menu Options for the External Function Source Editor

Table 37 lists and briefly describes the right-click menu options available from the External Function Source Editor.

Table 37. External Function Source Editor Right-Click Menu Options

Option	Action/Description
Create File	(Available only if you have *not* already used the External Function Source Editor to create external function source code) Allows you to create the external function TAS code for the selected external function definition.
Recreate File	(Available only if you have already used the External Function Source Editor to create external function source code) Allows you to recreate the external function TAS code for the selected external function definition. **NOTE:** Selecting this option deletes any code you have written, before creating the new code. Use this option to “clear the slate” and start new with the external function.
Delete File	Deletes the external function file.
Edit File	Allows you to edit existing external function code without deleting the original code.
Refresh	Redraws all columns in the External Function Source Editor in cases where one or more columns are not visible.

Working with Hosts

17

Overview

Some applications are designed to exchange data with a 3270 IBM-type mainframe host remote to the target voice response system. Before the two machines can exchange data, they must be physically connected, and the interface protocols must be established. To accomplish this, you must define the interface for your target system and the host.

Defining a host interface is a three-step process:

1. Identify the host you want to establish a connection and exchange data with. This also requires that you know the screens you want to access on that host, because the connection is established on the target system, not in Voice@Work.
2. Import or create a host definition in Voice@Work.
3. Set up a host flow to define the login, logout, and recovery processes.

Host definitions are imported, created, and edited using the Hosts Manager.

Purpose

The purpose of this chapter is to provide information about using hosts with Voice@Work applications. Topics include:

The Hosts Manager

The Hosts Manager displays the names of all hosts defined for the current application. From the Hosts Manager, you can:

- Display host definitions imported using the Application Resources tool.
- Create new host definitions.
- Rename host definitions.
- Edit a host definition.
- Define a host flow.
- Use the right-click menu in the Hosts Manager to perform other operations with host definitions.

Opening the Hosts Manager

To open the Hosts Manager, do one of the following:

- From the **View** menu, select **Hosts**.
- Click the  button on the Toolbar.
- Open the Globals Manager and locate the Hosts Manager, if it is docked to the Globals Manager.

Voice@Work displays the Hosts Manager window (Figure 55).

Figure 55. Hosts Manager Window

The Hosts Manager displays all hosts defined for the current application listed in alphabetic order, along with the headings of the following tabs or areas from the Host Definition Editor (see "The Host Definition Editor" below):

- Host Parameters
- Screen Definitions
- Login/Password Pairs
- Screen Identifications
- Field Definitions

NOTE:
Only headings are displayed here. You cannot view or edit the contents of any of these fields directly from the Hosts Manager. Instead, you must use the Host Definition Editor to view and edit host definitions (see "The Host Definition Editor" below).

Importing Host Definitions

It is often desirable to reuse host definitions created for other applications when they exist. Voice@Work allows you to import host definitions that have been created for other applications and that have been shared out by other developers. This can save you the trouble of having to create new host definitions each time you want to use one in your application.

To import a host definition, do the following:

1. Open the Application Resources tool. See Chapter 21, "The Application Resources Tool" for the procedure.
2. Click the **Import** tab at the bottom of the window.
3. (Optional) From the pulldown **Drive** menu at the bottom of the window, select the mapped network drive containing available resources.
4. Select **Hosts** from the Application Resources Options menu. See Chapter 21, "The Application Resources Tool" for the procedure.

   Voice@Work displays a list of available host definitions and their locations.
5. Select whether you want to import the resource as **Shared** or as a **Private Copy** by clicking the appropriate radio button.
6. Do one of the following:
   - Highlight the line entry for the host definition you want, and then click the **Import** button.
   - Double-click the line entry for the host definition you want.

   Voice@Work imports the selected host definition and displays it as an entry in the Hosts Manager.

Creating New Host Definitions

To create a new host definition, do the following:

1. Open the Hosts Manager. See "Opening the Hosts Manager" below for the procedure.
2. Right-click in the Hosts Manager somewhere *other* than on the name bar.
3. Select **New...** from the popup menu.

   Voice@Work displays a dialog box and prompts you for a name for the new host (Figure 56).

Figure 56. Name New Host Definition Window

4. Enter the name for the host in the space provided.

   When naming hosts, keep the following in mind:

   - Names must be 11 characters or less in length. Voice@Work truncates host names with more characters to the first 11 characters.
   - Do not use spaces in host names. Voice@Work ignores and deletes spaces when creating the actual host name.
   - Host names are case sensitive. Voice@Work preserves cases as you assign them when naming the branch. The one exception is that the first letter of a name is always converted to upper case, regardless of how you type it in. For example, if you typed in **new_ Host** as the host name, Voice@Work converts it to **New_Host** when creating the host.

     NOTE:
     Although Voice@Work preserves cases when saving names and transfers them to the target system with cases preserved, you cannot have two names with the same letters. If, for example, you have a host called **New_Host**, you cannot also create a host called **NEW_HOST**.

   - You may use any standard ASCII character as part of a host name, but the first letter must be alphanumeric. Voice@Work does not support the use of extended ASCII characters in host definition names.

5. Click **OK**.

   Voice@Work displays the Host Definition Editor window (Figure 57).

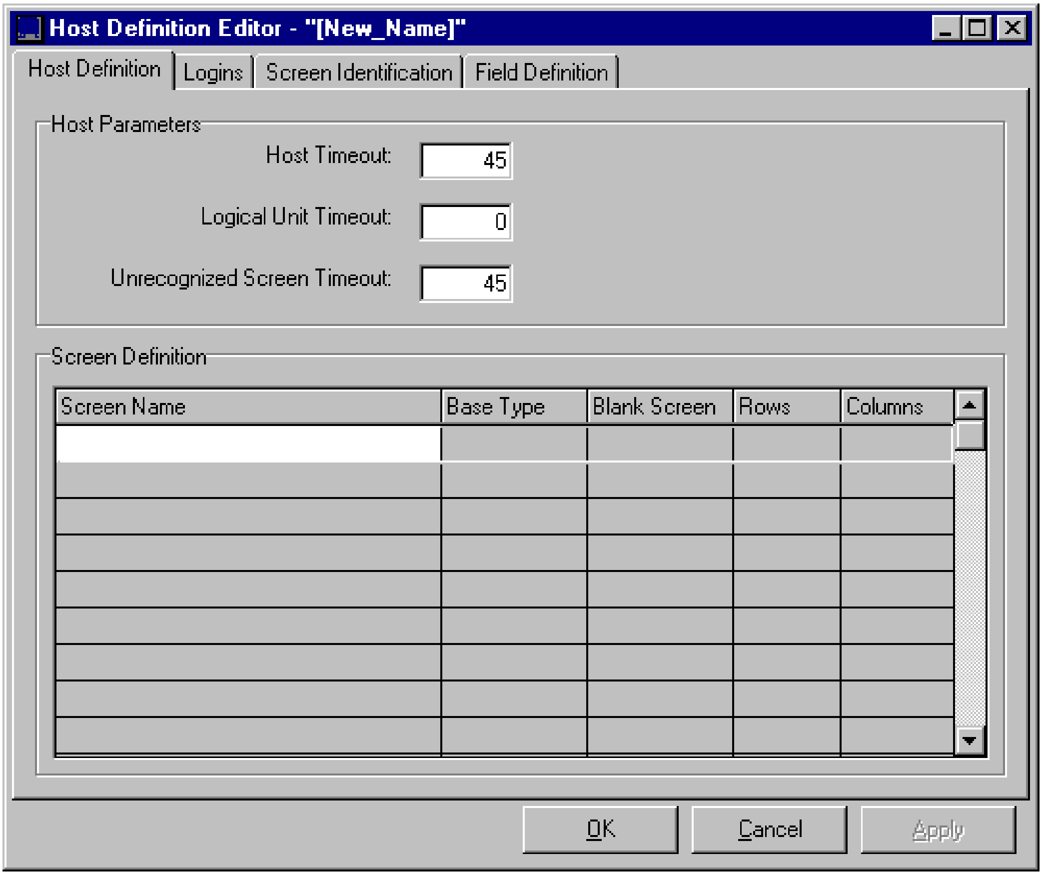


Figure 57. Host Definition Editor Window

6. Define the timeouts and the screens to use in exchanging data with the host in the Host Definition Editor. See "The Host Definition Editor" in this chapter.
7. Click **OK**.

   Voice@Work closes the Host Definition Editor and displays the new host definition name as an entry in the Hosts Manager.

Renaming a Host Definition

To rename a host definition, do the following:

1. Open the Hosts Manager. See “Opening the Hosts Manager” above for the procedure.
2. Right-click the name of the host definition you want to rename.
3. Select **Rename...** from the popup menu.

   Voice@Work displays a dialog box and prompts you for a new name for the host definition (Figure 58).

Figure 58. Rename Host Definition Window

4. Enter the new name for the host definition in the space provided, keeping in mind the naming conventions for host definitions (see Step 4 in “Creating New Host Definitions” above).
5. Click **OK**.

   Voice@Work changes the name of the host definition in the Hosts Manager window (Figure 59).

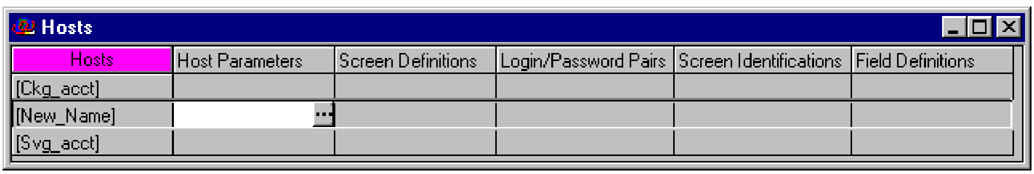


Figure 59. Host Manager Window Showing New Host Definition Name

Editing a Host Definition

To edit a host definition, do the following:

1. Open the Hosts Manager. See "Opening the Hosts Manager" above for the procedure.
2. Right-click the name of the host definition you want to edit.
3. Select **Edit Host** from the popup menu.

   Voice@Work displays the Host Definition Editor window (Figure 60).
4. Edit the Host Definition as needed. See "The Host Definition Editor" section in this chapter.
5. Click **OK**.

Right-Click Menu Options for the Hosts Manager

Table 38 describes the right-click menu options for the Hosts Manager.

Table 38. Hosts Manager Right-Click Menu Options

Menu Option	Action/Description
New...	Allows you to create a new host definition; see "Creating New Host Definitions" above.
Delete	Deletes the host definition from the Hosts Manager, if it is not in use. **NOTE:** If you have used the host definition in your application, you must first remove its use from the application before you can delete it.
Rename...	Allows you to rename the selected host definition without altering its contents; see "Renaming a Host Definition" above.
Find Uses of	Opens the Find window, searches for Uses Of the selected host definition, and presents the results of the search in the **Find window results** field; see "The Find... Feature" in Chapter 6, "The Edit Menu".

Continued on next page

Table 38. Hosts Manager Right-Click Menu Options — *Continued*

Menu Option	Action/Description
Sharing...	Allows you to designate a custom host definition as Shared, so that you (or other developers) can use it in other applications. When a host definition is marked as shared, it can be imported into other applications using the Application Resources tool. See Chapter 21, "The Application Resources Tool" for additional information.
Stay on Top	(Available only when the Hosts Manager is undocked) Toggles on and off. When checked (toggled on), it causes the Hosts Manager window to remain on top of other open Voice@Work windows. **NOTE:** If you have more than one resource manager window toggled to "Stay on Top", the window actually on top will be the last window you clicked in. All windows toggled to "Stay on Top", however, will stay on top of any other Voice@Work windows that are open.
Copy Host...	Makes a copy of the selected host definition and prompts you for a new name for the copy.
Edit Host	Allows you to edit the selected host definition; see "Editing a Host Definition" above.
Edit Host Flow	Allows you to edit the host (call) flow for the selected host definition; see "Host Flows" below.

The Host Definition Editor

The Host Definition Editor allows you to define all the interface elements needed to enable the target voice-response system to exchange data with the host.

From the Host Definition Editor you can:

- Set host timeout parameters.
- Select and define screens to use (on the host).
- Set login and password information.
- Specify unique elements to identify each selected screen.
- Identify fields for each selected screen to be used for the exchange of data.
- Use the right-click menu to perform various operations in the Host Definition Editor.

Opening the Host Definition Editor

To open the Host Definition Editor, do the following:

1. Open the Hosts Manager. See “Opening the Hosts Manager” above for the procedure.
2. In the Hosts Manager window, right-click on the name of the host definition you want to work with.
3. Select **Edit Host**.

   Voice@Work displays the Host Definition Editor. The Host Definition Editor has four tabs, each used for defining different elements of the host interface:

   - The Host Definition tab (Figure 60)
   - The Logins tab (Figure 61)
   - The Screen Identification tab (Figure 62)
   - The Field Definition tab (Figure 63)

The Host Definition Tab

The Host Definition tab (Figure 60) is used to set timeout parameters and define the screens to be accessed on the host.

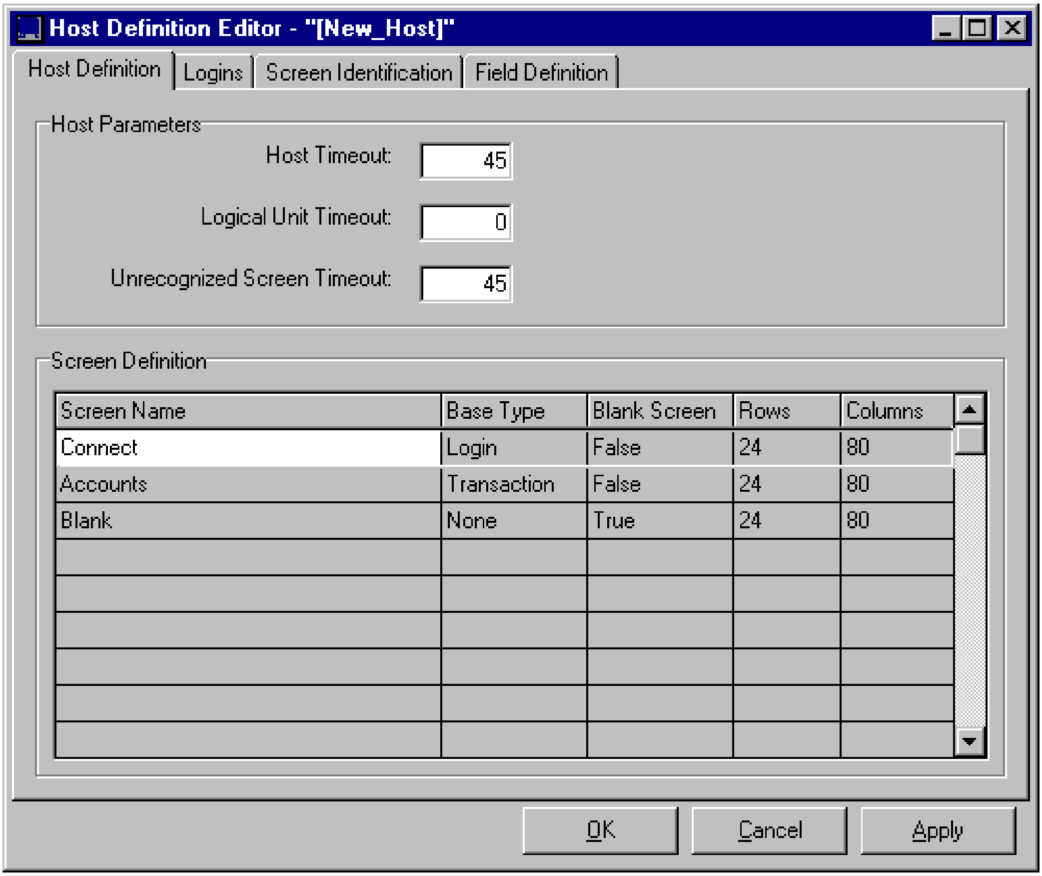


Figure 60. Host Definition Editor — Host Definition Tab

Setting Host Timeout Parameters

The Host Parameters section of the Host Definitions tab contains the fields for setting timeout parameters for connecting to the host. Table 39 summarizes the options in the Host Parameters section.

Table 39. Host Parameters on the Host Definition Tab

Parameter	Notes/Comments
Host Timeout	Enter the number of seconds to wait for a response from the host after a Host Send. If the host does not respond within this time frame, a Host Timeout condition occurs. The range is 1–300 seconds. The default is 45 seconds. **NOTE:** If this field is set to 0 (zero), the application waits indefinitely for a response from the host.
Logical Unit Timeout	Enter the number of seconds to wait for a Logical Unit (LU) to be assigned to the application. If an LU is not available within the time frame specified, a Timeout condition occurs. The range is 1–45 seconds. The default is 0 seconds. **NOTE:** If this field is set to 0 (zero), the application does not wait for an LU to become available.
Unrecognized Screen Timeout	Enter the number of seconds to wait for a specific screen of data from the host. If a screen is received that is not the specified screen, and the time has elapsed, an Unrecognized Screen Timeout condition occurs. The range is 1–300 seconds. The default is 45 seconds. **NOTE:** If this field is set to 0 (zero), the application waits indefinitely for a response from the host.

Selecting and Defining Screens

The Screen Definition section of the Host Definitions tab contains the fields for naming and classifying the screens to be used in exchanging data with the host. Table 40 summarizes the options in the Screen Definition section.

Table 40. Screen Definitions on the Host Definition Tab

Field	Notes/Comments
Screen Name	Type in the name to be assigned to the screen.
Base Type	From the pulldown menu, select the base type for the screen. There are three options: ■ **Login** — Typically, the first screen expected from the host ■ **Transaction** — The screen you want the system to be on when a call comes in ■ **None** — (Default) All other screens
Blank Screen	From the pulldown menu, select **True** if this screen is a blank screen. Select **False** (the default) if it is any other kind of screen. **NOTE:** You can only have one Blank Screen per host definition.
Rows	Enter the number of rows defined for the screen, normally the default – 24.
Columns	Enter the number of columns defined for the screen, normally the default – 80.

NOTE:
Before going to another tab, click the **Apply** button.

The Logins Tab

The Logins tab (Figure 61) contains the fields for logins and passwords required for logging into the host.

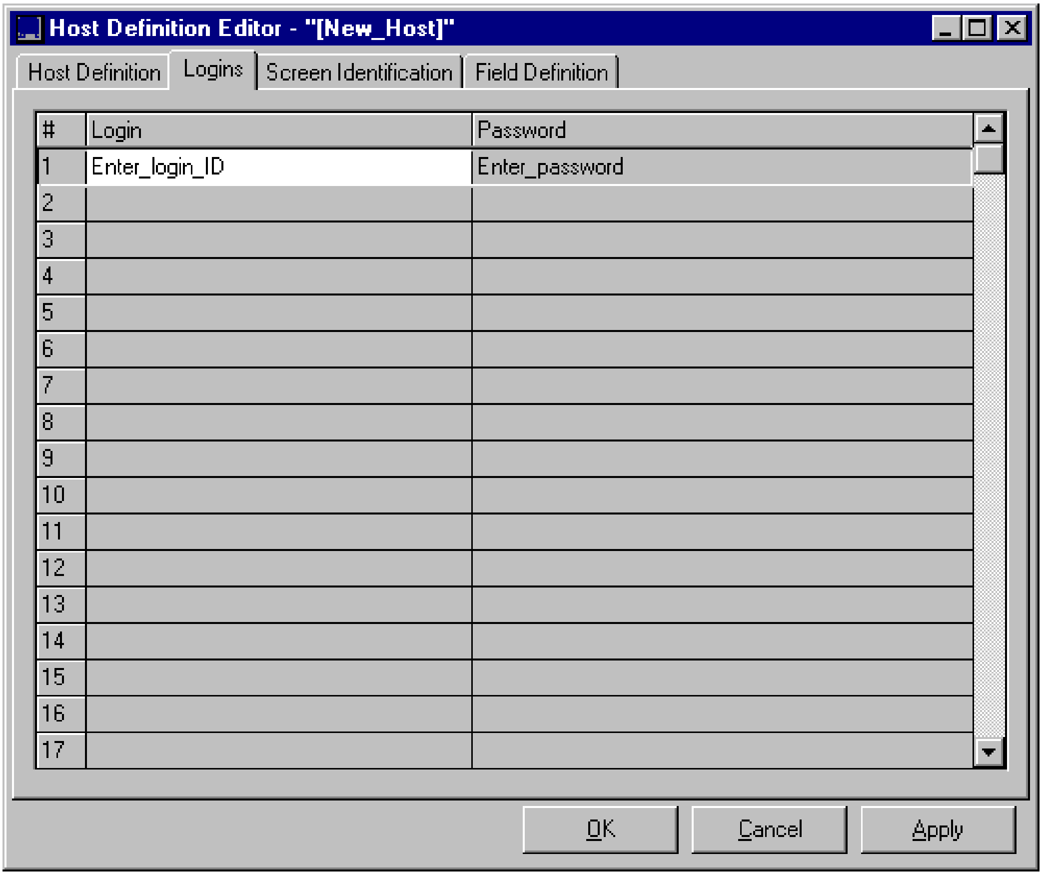


Figure 61. Host Definition Editor — Logins Tab

The Logins tab has three columns. Table 41 summarizes the fields on the Logins tab.

Table 41. Host Definition Editor — Logins Tab Fields

Column/Field	Notes/Comments
#	Displays the number assigned to each login and password set A login and password set must be defined for the maximum number of LUs the host definition will be assigned to. For example, if this host definition will be assigned to 50 LUs, then there must be 50 login and password sets defined on this tab. **NOTE:** Login and password sets do not have to be unique. If the login and password is the same for each login, then Lucent Technologies recommends that you leave this tab blank and define the login and password as part of the Host Flow (see "Host Flows" below).
Login	Type in a login ID valid for logging in to the host.
Password	Type in a password valid for logging in to the host. This must be the password for the Login on the same line. If a password is not required for the Login, leave this field blank.

NOTE:
Before going to another tab, click the **Apply** button.

The Screen Identification Tab

Use the Screen Identification tab fields (Figure 62) to define areas of the screen from the host that will uniquely identify the screen.

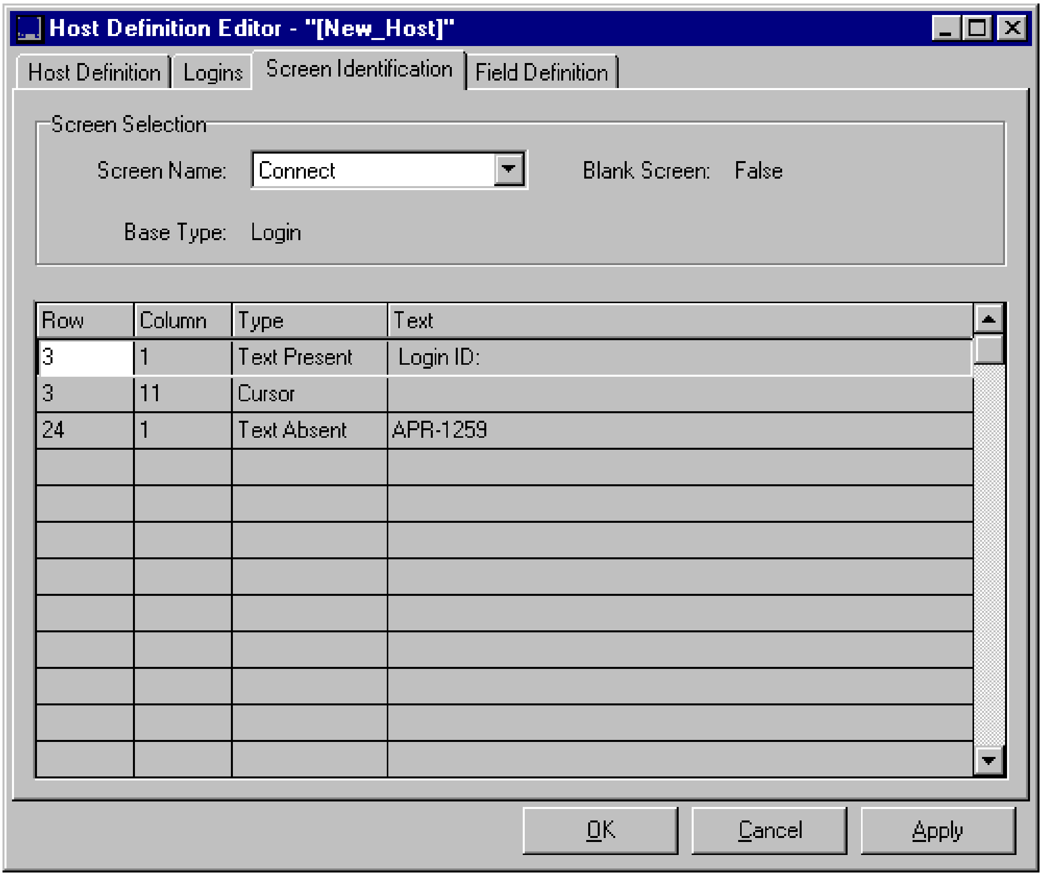


Figure 62. Host Definition Editor — Screen Identification Tab

The **Blank Screen:** and **Base Type:** fields reflect the options set in the Screen Definition area of the Host Definition tab.

There must be at least one **Text Present**, **Text Absent**, or **Cursor** field defined for the screen unless **Blank Screen** is selected on the Host Definition tab. A Blank Screen does not need or permit any identifiers.

NOTE:
Screens can be very similar in appearance. Some information on each screen remains constant (text and prompts, for example), while other information varies (data or error messages, for example). You should select constant information to use as identifiers.

The target system must be shown the part or parts of the screen that enable it to uniquely identify that screen from all the other screens when they arrive from the host.

This implies that you must be very familiar with the screens on the host, and that you must be extremely careful to find unique identifiers for the screens you want to use.

Table 42 describes the fields on the Screen Identification tab.

Table 42. Host Definition Editor — Screen Identification Tab Fields

Column/Field	Notes/Comments
Screen Name	From the pulldown menu, select the name of screen you are targeting on the host. **NOTE:** This list is blank until you have defined at least one screen in the Screen Definition area of the Host Definition tab.
Row	Specify the number of the row (1–24) within the host screen from which the identification of the screen is to begin.
Column	Specify the number of the column (1–80) within the host screen from which the identification of the screen is to begin.

Continued on next page

Table 42. Host Definition Editor — Screen Identification Tab Fields

Column/Field	Notes/Comments
Type	Select from the pulldown menu the type that defines whether the screen identification is based upon: ■ Text being present (Text Present option)... ■ Text being absent (Text Absent option)... ■ The cursor being present (Cursor option)... ...at the specified row and column. **NOTE:** – Data in **Text Present** fields cannot overlap. – Data in **Text Absent** fields can overlap. – Only one cursor can be defined for each screen.
Text	If **Text Present** has been selected in the screen Type column, type in the text that must be present for successful identification of the screen on the host. This text must start at the specified row and column. **NOTE:** This text must appear exactly as it does on the host screen. The text identifier is case sensitive. If **Text Absent** has been selected in the screen Type column, type in text that must *not* be present at the specified row and column. If **Cursor** has been selected in the screen Type column, this field is unavailable.

NOTE:
Before going to another tab, click the **Apply** button.

The Field Definition Tab (Optional)

Use the Field Definition tab fields (Figure 63) to define fields on screens sent to or received from the host.

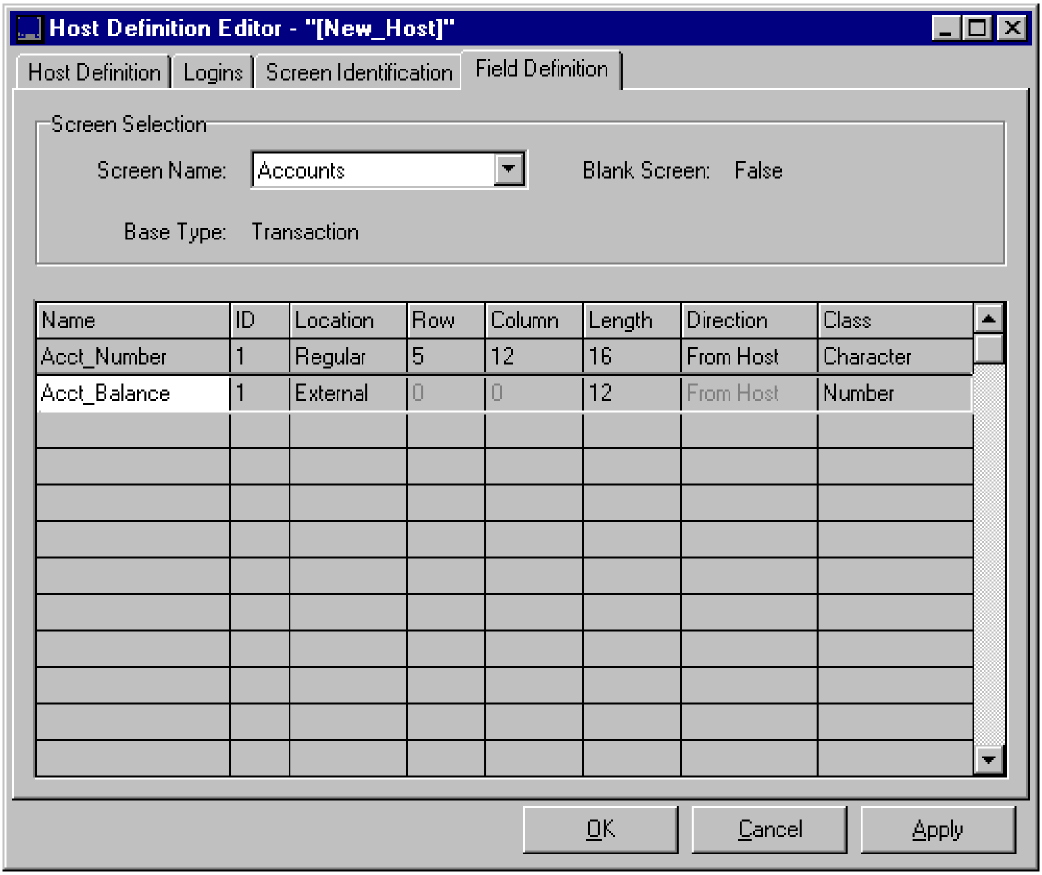


Figure 63. Host Definition Editor — Field Definition Tab

The **Blank Screen:** and **Base Type:** fields reflect the options set in the Screen Definition area of the Host Definition tab.

Table 43 describes the fields on the Field Definition tab.

Table 43. Host Definition Editor — Field Definition Fields

Column/Field	Notes/Comments
Screen Name	From the pulldown menu, select the name of screen you are targeting on the host. **NOTE:** This list is blank until you have defined at least one screen in the Screen Definition area of the Host Definition tab.
Name	Type in the name of the field to be defined for this screen.
ID	Automatically assigned; this number indicates the number of occurrences of this field name. Use this number or type another ID number over it. For example, if the field Name “Found_Baggage” appeared twice in the Field Definitions tab, the first occurrence is assigned an ID of 1 and the second is assigned an ID of 2. TIP: This field can be used for repeating lines on a host screen.
Location	From the pulldown menu, select: ■ **Regular** if the desired field's location is known ■ **External** if the desired field's location is not known For more information about external fields, see More about the Location Field and External Fields below.
Row	Type in the number of the row (1–24) within the host screen from which the field originates.
Column	Type in the number of the column (1–80) within the host screen from which the field originates.

Continued on next page

Table 43. Host Definition Editor — Field Definition Fields — *Continued*

Column/Field	Notes/Comments
Length	Type in the maximum length the field may have. **NOTE:** When assigning a length to a character field that will be used as part of a host definition, keep the following in mind: For a single host screen, the total size of all fields defined must be equal to or less than 2000 bytes minus 1 byte of overhead for each host field.
Direction	Select from the pulldown menu the direction that data may travel between the target voice response system and the host. Options include: ■ **To Host** — Data may travel only from the target system *to* the host for the field defined ■ **From Host** — Data may travel only to the target system *from* the host for the field defined ■ **Both** — Data may travel *both* directions between the two systems
Class	Select from the pulldown menu the class of data that may be used for the field defined. Options include: ■ **Character** — Includes all standard ASCII alphanumeric characters ■ **Number** — Includes only numbers and the minus symbol **NOTE:** The **Number** class strips out all dollar signs, decimal points, and commas.

More about the Location Field and External Fields

For most host definitions, the Location field is set to the default, **Regular**, setting. This means that you know the location (starting row and column) of the desired field on the host and that it is in a fixed location. In these cases, the Row and Column fields must be defined, so the application knows where to find the requested field.

In some cases, however, the requested field may not be in a fixed location. In these cases, you cannot know the location of the field, because it may be in a different location each time you need to access it. For these cases, you should set this Location field to **External**.

When this field is set to **External**, the Row and Column fields are set to zero (0) and made inactive (grayed out). Also, the Direction field is set to **From Host** and made inactive. You must still fill in the Name, ID, Length, and Class fields.

NOTE:
In earlier versions of Voice@Work (3.0 and below), this field did not exist. If you are upgrading applications created in an earlier version of Voice@Work or in Graphical Designer, this field is, by default, set to **Regular** for all host definitions.

When the running application encounters an external field, it invokes a “helper DIP”, where it expects to find the instructions necessary to locate the field. You, as the application developer, must provide the instructions for this helper DIP. For more information about this helper DIP, see The Helper DIP below.

When working with hosts that use external fields, you may find it helpful to print out the screen containing the external field. This can, for instance, be helpful in debugging an application. To print out the screen in question, you can use the **fancy_print** subroutine. For more information about using the fancy_print subroutine, see Using the “fancy_print” Subroutine below.

The Helper DIP

The "helper DIP" provides instructions necessary to locate a field defined as an **External** field in an application. This helper DIP must be modified from the existing DIP. This DIP must be written using the C programming language.

This file, named **extract.c**, is located in the /att/ag/hostdip/helper directory on the Intuity CONVERSANT. It already contains the basic structure required to pass field location instructions to an application. What you must provide is the instructions themselves.

NOTE:
You can provide instructions to locate as many external fields as necessary.

Editing the helper DIP

To edit the helper DIP, enter the following at the system prompts:

cd /att/ag/hostdip/helper
vi extract.c

There are two subroutines that need to be modified in the extract.c file. When a screen that uses external fields is received from the host, the **new_screen** subroutine is called and then the **extract** subroutine is called for every external field defined in that screen.

NOTE:
The extract.c file provided with Voice@Work is installed whenever the Intuity CONVERSANT system software is reloaded. Reloading the system software overwrites any existing customized extract.c files. If you want to retain a customized extract.c file for later use, first save this file under another name, preferably in a different directory. You can then copy the saved customized file over the generic extract.c file after you have completed the software upgrade procedure.

Using the "new_screen" Subroutine

Using the **new_screen** subroutine, you may want to check what screen it is and set up some global data structures. You should be aware that the Voice@Work host screen name used at the TAS level consists of the host name and the screen name with an underscore between them as follows:

<host name>_<screen name>

For example, if the host name in Voice@Work is “Hostappl” and the host screen name is “ACCOUNT”, then the host screen name that should be referenced in the helper DIP is:

hostappl_ACCOUNT

Using the "extract" Subroutine

Use the **extract** subroutine to check what screen this is, what field is to be extracted, and then to extract the field from the screen and put it into the variable field. You should be aware that the Voice@Work host field name used at the TAS level consists of the host name, screen name, field name, and ID with an underscore between them as follows:

<host name>_<screen name>_<field name><ID>

For example, if the host name in Voice@Work is “Hostappl”, the host screen name is “ACCOUNT”, the host field name is “BALANCE”, and the ID is “1”, then the host field name that should be referenced in the helper DIP is:

hostappl_ACCOUNT_BALANCE1

The **extract** subroutine is called for every external field defined in every application. If more than one external field is defined within the applications on the system, you should check the parameters of the **extract** subroutine to see which field should be extracted. The *appl_name*, *screen_name*, and *fld_name* parameters correspond to the names defined within Voice@Work. The field provided by the extract subroutine should not be larger than the field **Length** size specified within Voice@Work.

Using the RCTOINDEX Macro

A screen image (24 rows by 80 columns) is actually stored in a one-dimensional array of 1920 characters. The RCTOINDEX macro may be used to map row and column numbers to an index for the screen array. An example use of the RCTOINDEX macro within the extract.c routine follows:

strncpy(field, &screen[RCTOINDEX(10,16)], 3);

The following is the syntax for the RCTOINDEX macro:

#define RCTOINDEX(*r*,*c*) ((*r*-1) * 80 + (*c*-1))

where *r* is the row number (where the first row is 1) and *c* is the column number (where the first column is 1).

Using the “fancy_print” Subroutine

When working with hosts that use external fields, you may find it helpful to print out the screen containing the external field. This can, for instance, be helpful in debugging an application. To print out the screen in question, you can use the **fancy_print** subroutine, calling it from the **new_screen** subroutine or the **extract** subroutine.

When the screen data is printed out, nulls are replace with blank spaces and attributes with exclamation marks (!) for readability. After 78 characters, **db_pr** inserts newline characters, so if you try to print more than 78 characters per line, the line is split.

Installing the New Helper DIP

Once you have modified and saved the extract.c file, you must re-install it on the Intuity CONVERSANT system.

To install the new helper DIP, enter the following at the system prompts:

cd /att/ag/hostdip/helper
stop_vs
make
start_vs

NOTE:
The installation procedure temporarily halts any other applications running on the system.

Using DIP Instance Numbers

If you write your own DIP and access it through an external function, you need to know the DIP and the DIP instances reserved by the Intuity CONVERSANT system. The DIP "instance number" must be unique among all instances of this DIP in the system. The instance numbers range from 0 to 34. CONVERSANT uses 0, 1, 4, and 11 as shown below:

- DIP0, instance 0 – 3270 host DIP
- DIP1, instance 1 – local database DIP
- DIP4, instance 4 – recording speech
- DIP11, instance 11 – helper DIP

Duplicating an instance number within the same DIP number keeps the DIP from operating properly. For example, if you write a DIP and use instance number 0 twice, one instance of your DIP will never get past the call to start up. Whichever instance of the DIP gets started first uses the instance 0 and blocks the other from starting.

Right-Click Menu Options for the Host Definition Editor

The Host Definition Editor has its own right-click menu. Table 44 describes the menu options.

NOTE:
The right-click menu options may vary from the list below, according to which tab you are on.

Table 44. Host Definition Editor Right-Click Menu Options

Menu Option	Action
Validate Grid	This option checks the data on the selected tab to make sure that all required fields are populated.
Delete Row	This option deletes the selected row. **NOTE:** This action cannot be undone.
Insert Row	This option adds a new row just *above* the selected row.
Add Row	This option appends a new row *at the end* of all screens or fields (rows) defined.

Continued on next page

Table 44. Host Definition Editor Right-Click Menu Options — *Continued*

Menu Option	Action
Move Row Up	This option moves the selected screen up one row.
Move Row Down	This option moves the selected screen down one row.
Copy Screen	This option is available on Host Definition tab only. It allows you to make a copy of the selected screen. When this option is selected, Voice@Work prompts you for a new name for the copy.
Fill Login/Password with default...	This option is available on Logins tab only. It allows you to assign a default login ID and password to any number of sets you want. When this option is selected, Voice@Work displays the Create Default Login window (Figure 64), and prompts you for the default login ID and password and for the number of sets to which you want to assign these defaults. These defaults are assigned to the sets starting with the selected row.
Create Variables	This option is available on Field Definition tab only. It automatically creates variables for all the fields defined on the Field Definition tab. When selected, this option creates and names a new variable with the ID number appended to it for each field defined, opens the Variables Manager, and places all the newly created variables in the Variables Manager.
Create Variables Interactively	This option is available on Field Definition tab only. It allows you to selectively create variables for fields defined on the Field Definition tab. When selected, Voice@Work prompts you to decide whether you want to create a new variable with the ID number appended to it for each field defined. For each variable you decide to create, Voice@Work opens the Variables Manager and places the newly created variable in the Variables Manager.

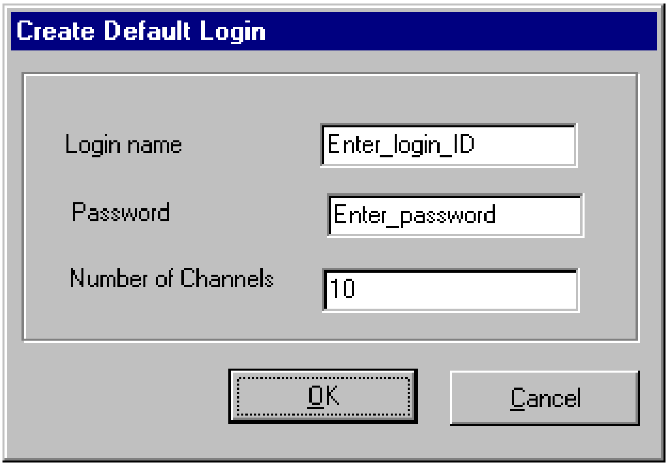


Figure 64. Create Default Login Window

NOTE:
Voice@Work does not automatically update variables based on Field Definition entries.

- If you create a variable for an entry in the Field Definition tab of the Host Definition Editor, and you later decide to change the entry in the Host Definition Editor, you must go back and manually alter the variable as well.
- Alternatively, you may choose to recreate the variable and keep both versions of the entry variable or delete the old variable definition.

If you attempt to create a variable from a Field Definition tab entry, and another variable already exists with the same name, no new variable is created, and the original variable definition remains as it was.

Additional Right-Click Menu Options

Table 45 lists and describes the right-click menu options available only from the Host Parameters section of the Host Definition tab.

Table 45. Additional Right-Click Menu Options

Menu Option	Keyboard Shortcut	Action
Cut	CONTROL X	Allows you to cut highlighted text from the selected field and saves it to the clipboard.
Copy	CONTROL C	Allows you to copy highlighted text without cutting it from the selected field and saves it to the clipboard.
Paste	CONTROL V	Allows you to paste whatever is on the clipboard to the selected field. **NOTE:** You must first place the cursor in the field where you want to paste. If you do not, the Paste function does not work.
Delete	DEL	Allows you to delete highlighted text from the selected field. **NOTE:** This action cannot be undone. The highlighted text is *not* saved to the clipboard.
Select All	CONTROL A	Allows you to select all text in the selected field. **NOTE:** You must first place the cursor in the field where you want to highlight text.
Insert Variable...	CONTROL B	Not functional in the current version.

Host Flows

The primary purpose of the Host Flow is to set up the interactions between the target system and the host. There is one Host Flow for each Host Definition in the application.

NOTE:
The only nodes that can be used in the Host Flow are the Host Send node and the Host Send/Receive node.

The Host Flow contains three call flows, each with its own function:

- **Login** — (Figure 65) Defines the sequence of screens needed to log in to the host. The goal of this flow is to get from the login screen to the transaction screen, that is, the screen you want the system to be on when a call comes in.

  The amount of time to wait for a timeout condition to occur is set in the Host Definition Editor. See “The Host Definition Tab” in “The Host Definition Editor” above for additional information.

  If the system cannot logged in or get to the transaction screen, the **Recover** sequence is started.

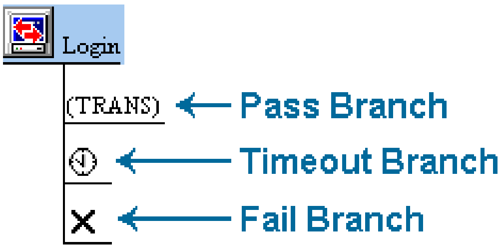


Figure 65. Host Login Flow

- **Logout** — (Figure 66) Defines the sequence of screens needed to log out from the host. This flow goes from the transaction screen to the login screen. The Logout flow is typically used when the system is coming down for some reason, such as when the voice system is being stopped.

  The amount of time to wait for a timeout condition to occur is set in the Host Definition Editor. See “The Host Definition Tab” in “The Host Definition Editor” above for additional information.

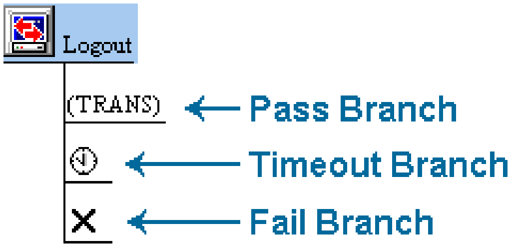


Figure 66. Logout Flow

- **Recover** — (Figure 67) Defines the recovery sequence for all screens. The goal of this flow is to recover when an unexpected screen is received, for example, or when a caller hangs up before completing a transaction. In cases like these, the Recover flow attempts to get back either to the transaction screen or the login screen.

  The amount of time to wait for a timeout condition to occur is set in the Host Definition Editor. See "The Host Definition Tab" in "The Host Definition Editor" above for additional information.

  If the system cannot recover and get back to either the login or the transaction screen, the **Fail** branch is taken.

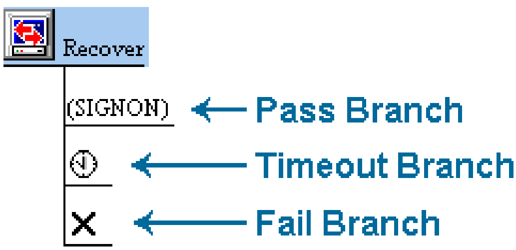


Figure 67. Host Recover Flow

When working with Host Flows, keep the following in mind:

- The login screen is the screen that your application first uses to establish a working connection with the host.
- The first action step of the Login flow sequence should be to get the host login screen. The last screen of the Login sequence must be a transaction screen.
- The Logout flow sequence should end with the login screen.
- The transaction screen is the screen that the application returns to after a caller transaction. You must specify to the system what screen to start with to get to any other systems. That screen is the transaction screen.
- In many ways, the Recover flow sequence is the most critical of the three host flow sequences. When developing the Recover flow, you must try to include all possible cases, including unrecognized screens and host timeouts. The purpose of the Recover flow is to tell an application how to recover if it detects a screen that it does not expect. Whenever this happens, the system automatically invokes the Recover flow.
- The Recover flow is initiated whenever one of the following occurs:
  - — A transaction ends and the current host screen is not the transaction screen.
  - — The Login procedure ends and the current host screen is not the transaction screen.
  - — The Recover procedure ends and the current host screen is not the transaction screen, unless the current host screen is the login screen, which results in the application going to the Login flow.
- The result of the Recover flow sequence is the transaction screen, unless the current host screen is the login screen, which results in the application going to the Login flow.
- Make sure to include the login screen in the Recover flow. If the login screen is not included, it could be treated as an "unrecognized screen" and subjected to whatever action is specified for the "unrecognized screen" case. If no action is specified for the login screen case, the Login flow is automatically invoked after recovery fails and the login screen is recognized.
- It is possible to define the Login and Recover flows so that the system goes into an infinite loop. If the Recover flow does result in arrival at the transaction screen, it causes the Recover flow to be invoked. Failing to arrive at the transaction screen again, it causes the Recover flow to again be invoked, and so on.
- Make sure you test your flow to ensure that it does indeed terminate at the transaction screen. This can save considerable debugging effort later, when the application is installed on the target system.

Working with Phrases

18

Overview

Phrases in Voice@Work are bits of spoken or written speech, often used to instruct customers or prompt them for pieces of information. Individual phrases are grouped together in *phrase tables* and used in prompts. See Chapter 19, "Working with Prompts" for detailed information about prompts.

By default, Voice@Work includes two phrase tables with each application:

- The StandardSpeech table, based on the Enhanced Basic Speech language selected for the application
- A custom phrase table, labeled simply "PhraseTable"

You can create a maximum of 8 other phrase tables per application.

The phrases of only one phrase table at a time are displayed in the Phrases Manager.

Purpose

The purpose of this chapter is to provide information about creating, editing, and using phrases in Voice@Work applications. Topics include:

- The Phrases Manager
- Importing Phrase Tables
- Creating New Phrases
- Editing Existing Phrases
- Right-Click Menu Options for the Phrases Manager
- The Phrase Editor
- The Phrase Table Editor
- Importing Text
- The Phrase Text Import Dialog Box
- Create Default Phrases Dialog Box

The Phrases Manager

The Phrases Manager displays the contents of one phrase table at a time.

From the Phrase Manager you can:

- Select which phrase table to display.
- Create new phrases.
- Edit existing phrases.
- Use the right-click menu to perform many other operations on phrases and phrase tables.

Opening the Phrases Manager

To open the Phrases Manager, do one of the following:

- From the **View** menu, select **Phrases**.
- Click the  button on the Toolbar.
- Open the Globals Manager and locate the Phrases Manager, if it is docked to the Globals Manager.

Voice@Work displays the Phrases Manager window (Figure 68).

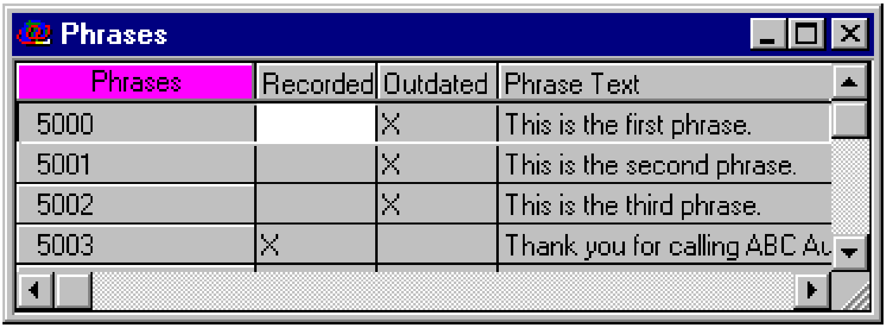


Figure 68. Phrases Manager Window

Table 46 describes the columns in the Phrases Manager window.

Table 46. Phrases Manager Columns

Column Heading	Notes
Phrase number	Displays the phrase numbers of all phrases defined for the current application.
Recorded	An X in this column indicates that an audio file exists for the corresponding phrase (on the PC).
Outdated	An X in this column indicates that the phrase text has been altered since a recording was made for it or that no recording has been made for the phrase (default).
Phrase Text	Displays the text assigned to each phrase. This text (the first 50 characters) becomes the phrase tag, which is used by the target system to identify the phrase. This text is displayed during Simulation.

Selecting a Phrase Table to Display

The Phrases Manager only displays the phrases of one phrase table at a time. If you want to view the contents of another phrase table, you must first select it from the right-click popup menu.

To select which phrase table to display:

1. Open the Phrases Manager. See "Opening the Phrases Manager" above for the procedure.
2. Right-click anywhere in the Phrases Manager *other than* the name bar.

   Voice@Work displays a right-click popup menu. Each phrase table for the current application is listed at the bottom enclosed in square brackets [ ]. The currently displayed phrase table is designated by a checkmark.
3. To select another phrase table for display, click the phrase table's name.

Importing Phrase Tables

Voice@Work allows you to import phrase tables using the Application Resources tool (for example, from other applications or phrase tables that have been shared by other developers). This feature is helpful when you want to reuse phrases that have been created elsewhere.

To import a phrase table into your application, do the following:

1. Open the Application Resources tool. See Chapter 21, "The Application Resources Tool" for the procedure.
2. Click the **Import** tab at the bottom of the window.
3. (Optional) From the pulldown **Drive** menu, select the mapped network drive containing available resources.
4. Select **Phrase Table** from the Application Resources pulldown menu at the top of the window.

   Voice@Work displays a list of available phrase tables and their locations.

   NOTE:
   Enhanced Basic (Standard) Speech is imported automatically, based upon the language selected for the application and cannot be deleted.
5. Select whether you want to import the resource as **Shared** or as a **Private Copy** by clicking the appropriate radio button.

6. Do one of the following:

   - Highlight the line entry for the phrase table you want, then click the **Import** button.
   - Double-click the line entry for the phrase table you want.

   Voice@Work imports the selected phrase table and makes it the active phrase table in the Phrases Manager window.

Creating New Phrases

Most of the time, you will want to create your own custom phrases for each application you develop in Voice@Work. This is usually done from the Phrases Manager.

NOTE:
When you create phrase text for a phrase, you do not necessarily create the audio file to play it with. This phrase text is required, however, as the first 50 characters become the phrase tag used to identify the phrase on the target system. You can, however, also create the audio files for your phrases in Voice@Work.

To create a new phrase, do the following:

1. Open the Phrases Manager. See "Opening the Phrases Manager" above for the procedure.
2. Right-click in the Phrases Manager somewhere *other* than on the name bar.
3. Select **New...** from the popup menu.

   Voice@Work displays a new line in the Phrases Manager and opens the Phrase Editor (Figure 69). A phrase number is automatically assigned to the new phrase.

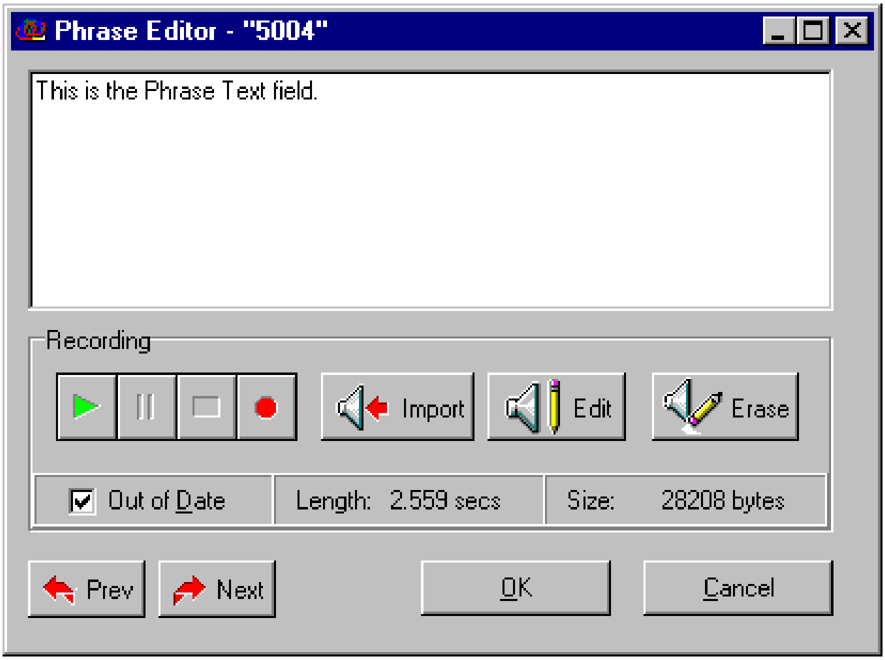


Figure 69. The Phrase Editor

4. Enter the phrase text you want for the phrase in the **Phrase Text** field. See "The Phrase Editor" below).
5. (Optional) Record the phrase using the **Recording** box controls, or import a prerecorded *.wav file using the **Import** button.

   NOTE:
   To record and play back recorded speech, you must have a sound card installed on your computer. You must also have speakers or headphones and a microphone.

   At present, Voice@Work supports only the *.wav format for audio files.

   To edit the *.wav file, click the **Edit** button. See "Editing Existing Phrases" below.
6. Click **OK**.

Editing Existing Phrases

To edit an existing phrase, do the following:

1. Open the Phrases Manager. See "Opening the Phrases Manager" above for the procedure.
2. Do one of the following:
   - Double-click the row containing the phrase you wish to edit.
   - Right-click the row containing the phrase you wish to edit, then select **Edit Phrase...** from the popup menu.

   Voice@Work displays the Phrase Editor window (Figure 69).
3. Edit the phrase text in the **Phrase Text** field. See "The Phrase Editor" below for the procedure.
4. (Optional) Record the phrase using the **Recording** box controls, or import a prerecorded *.wav file using the **Import** button. See "Audio Recording Tips" below.

   NOTE:
   To record and play back recorded speech, you must have a sound card installed on your computer. You must also have speakers and/or headphones and a microphone.

   At present, Voice@Work supports only the *.wav format for audio files.

   To edit the *.wav file, click the **Edit** button.
5. Click **OK**.

Right-Click Menu Options for the Phrases Manager

When you right-click in the Phrases Manager window anywhere except the name bar, Voice@Work displays a popup menu. Table 47 describes the menu options for the Phrases Manager right-click popup menu.

Table 47. Right-Click Menu Options for the Phrases Manager

Menu Option	Action
New...	Creates a new phrase, automatically assigns a phrase number to it, and opens the Phrase Editor window. See "The Phrase Editor" below.
Delete	Deletes the contents of the selected phrase from the table (both the phrase text and the *.wav file, if one exists, but *not* the phrase number).
Rename...	Inactive option in the Phrases Manager (grayed out).
Find Uses of	Opens the Find window, searches for Uses Of the selected phrase, and presents the results of the search in the **Find window results** field. See "The Find... Feature" in Chapter 6, "The Edit Menu".
Sharing...	Allows you to share an entire phrase table — individual phrases cannot be shared (see Chapter 10, "The Sharing Menu"). **NOTE:** The phrase table you want to share must have a unique name. If you are trying to share the custom phrase table, named "[PhraseTable]", which is created with each new application, you must first change the name of the phrase table using the Phrase Table Editor. See "The Phrase Table Editor" below.

Continued on next page

Table 47. Right-Click Menu Options for the Phrases Manager — *Continued*

Menu Option	Action
Stay on Top	(Available only when the Phrases Manager is undocked) Toggles on and off. When checked (toggled on), it causes the Phrases Manager window to remain on top of other open Voice@Work windows. **NOTE:** If you have more than one resource manager window toggled to "Stay on Top", the window actually on top will be the last window you clicked in. All windows toggled to "Stay on Top", however, will stay on top of any other Voice@Work windows that are open.
Edit Phrase...	Opens the Phrase Editor so you can edit the selected phrase. See "The Phrase Editor" below.
Display Alphabetically by Tag	Toggles on and off. When checked (toggled on), it lists all phrases in the table alphabetically by phrase tags.
Fill Phrases with Default...	Opens the Create Default Phrases dialog box, which allows you to create a user-specified number of new phrases. It also fills each new phrase with user-defined default phrase text. See "Create Default Phrases Dialog Box" below.
Import Phrases from Text...	Opens the Phrase Text Import dialog box, which allows you to select and import predefined text from sources outside Voice@Work. See "The Phrase Text Import Dialog Box" below.
Export Phrases for Spell Check...	Inactive in the current release of Voice@Work (grayed out).
New Phrase Table...	Opens the Phrase Table Editor, which allows you to create a new phrase table and define its basic parameters; see "The Phrase Table Editor" below.
Edit Phrase Table...	Opens the Phrase Table Editor and populates it with information about the currently open phrase table, which you can then edit. See "The Phrase Table Editor" below.

Continued on next page

Table 47. Right-Click Menu Options for the Phrases Manager — *Continued*

Menu Option	Action
Delete Phrase Table...	Deletes the currently selected phrase table. If any phrases from the table are being referenced by the current application, Voice@Work warns you and instructs you to find and remove the references before you can delete the table. **NOTE:** The Delete Phrase Table action cannot be undone. Enhanced Basic (Standard) Speech is imported automatically into each application and cannot be deleted.
[Phrase Table Names]	Lists the names of all phrase tables currently defined for the application. The currently active phrase table is designated by a checkmark. To select a different table as the active table, click that table's name.

The Phrase Editor

The Phrase Editor is used to:

- Write or edit phrase text.
- Make an audio recording of the phrase.
- Import an audio file.
- Edit an audio file.

Opening the Phrase Editor Window

To open the Phrase Editor window, do the following:

1. Open the Phrases Manager. See "Opening the Phrases Manager" above for the procedure.
2. Do one of the following:
   - Double-click any phrase table line entry.
   - Right-click anywhere in the Phrases Manager window *other than* the name bar; then from the popup menu, do one of the following:
     — Select **New...**
     — Select **Edit Phrase...**

   Voice@Work displays the Phrase Editor window (Figure 70).

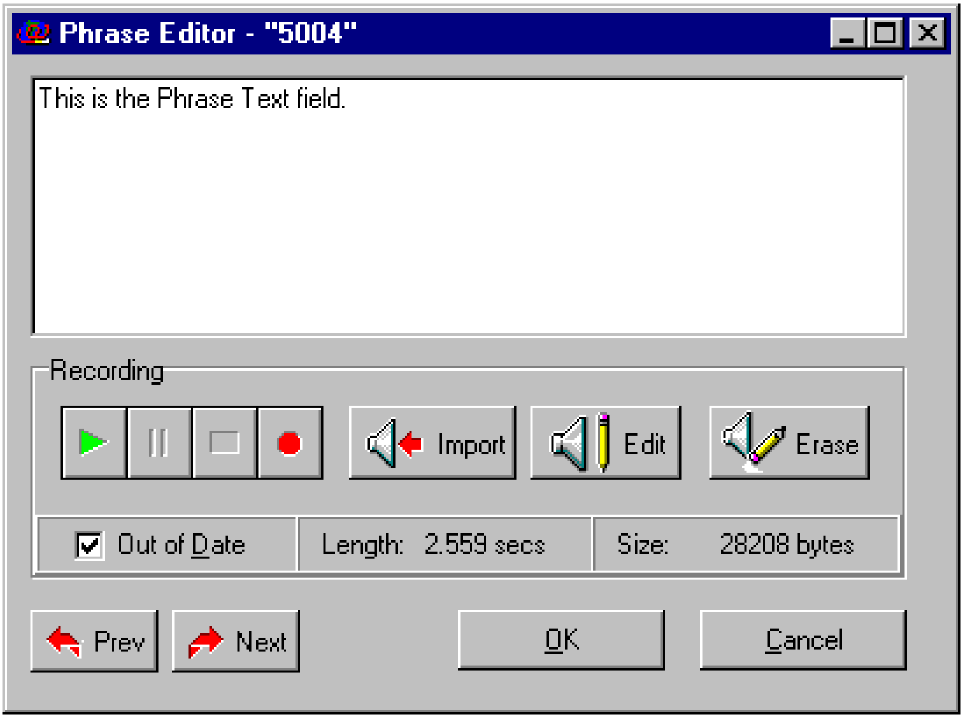
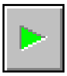
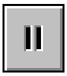
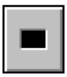
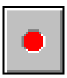


Figure 70. The Phrase Editor Window

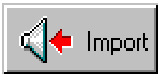
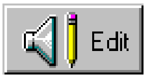
Table 48 describes the elements of the Phrase Editor window.

Table 48. Phrase Editor Window Elements

Element	Icon	Action
Phrase Text field		Type or import the text for the phrase you are creating here. Text in this field can be edited much like text in any basic word processor. Text created here appears in the Phrase Text column of the Phrases Manager. **NOTE:** This text (the first 50 characters) becomes the phrase tag, which is used by the target system to identify the phrase. See "Creating New Phrases", "Importing Text", and "The Phrase Text Import Dialog Box" in this chapter for more information.)
Play button		Allows you to play digitally recorded audio files. These audio files must be in *.wav format.
Pause button		Allows you to pause the playback of a recorded phrase. **NOTE:** The Pause button is currently operable only in Playback mode.
Stop button		Stops a recording or playback in progress and resets the recording to its beginning.
Record button		Allows you to digitally record a phrase. See "Audio Recording Tips" below for additional information.

Continued on next page

Table 48. Phrase Editor Window Elements — *Continued*

Element	Icon	Action
Import button		Allows you to import a prerecorded audio file from another source. **NOTE:** Audio files must be in *.wav format to playback in Voice@Work. Other audio formats are not currently supported.
Edit button		Allows you to edit a recorded phrase. Clicking on this button opens the default audio editing application on your machine This is typically the Microsoft Sound Recorder, unless you have specified another application. This button is only active when there is a recorded phrase present, either as part of the application or in the clipboard buffer.
Erase button		Allows you to delete the recording on the system clipboard and reset the **Length** and **Size** indicators to 0. **NOTE:** This action cannot be undone and does not warn you that you are permanently erasing the recording.
Out of Date checkbox		When checked, this signifies that changes have been made to the written text since a recording was made for the phrase. If no recording has been made, this box is checked by default. **NOTE:** You can override the “Outdated” status.
Length box		Displays the running time of the recording to the nearest tenth of a second.
Size box		Displays the size of the recording in numbers of bytes.

Continued on next page

Table 48. Phrase Editor Window Elements — *Continued*

Element	Icon	Action
Previous phrase (Prev) button		Allows you to select the previous phrase in the phrase table for editing without having to close and reopen the Phrase Editor.
Next phrase (Next) button		Allows you to select the next phrase in the phrase table for editing without having to close and reopen the Phrase Editor.
OK button		Accepts the edits made to the selected phrases and makes the changes in the application.
Cancel button		Cancels any changes made to the phrase and closes the Phrase Editor.

Writing and Editing Phrases Text

To write or edit phrase text, do the following:

1. Click anywhere inside the **Phrase Text** field (Figure 70).
2. Add, change, or delete the phrase text as desired.
3. Click **OK**.

Recording Phrases

To make an audio recording of the phrase, do the following:

NOTE:
To make or edit audio recordings, you must have a sound card, speakers or headphones, and a microphone properly connected to and configured on your computer.

1. Click the **Record** button.
2. Use the microphone to record the phrase.
3. Click the **Stop** button when finished recording.

Importing a Phrase Recording

To import a prerecorded audio file, do the following:

1. Make sure the audio file you want to import is in *.wav format and has been recorded at the proper sampling rate. The *.wav format is currently the only audio format supported by Voice@Work. See "Audio Recording Tips" below.
2. Click the **Import** button.

   If an audio file already exists for the phrase, Voice@Work displays a warning message asking if you want to overwrite the existing file.
3. Select the *.wav file you want from the **Open** dialog box.
4. Click **Open**.

Editing Phrase Recordings

To edit an audio (*.wav) file, do the following:

1. Click the **Edit** button.

   The default audio editing application for your machine opens. This is typically Microsoft Sound Recorder, unless you have specified another application.
2. Use the audio editing application to edit the file.
3. Make sure you save the edited file using the *.wav file format. See "Audio Recording Tips" below.
4. Close the audio editing application.

Audio Recording Tips

Keep the following in mind when recording phrases:

- When you are finished recording your phrase, check your recording by playing it back. To play it back, click the **Playback** button.
- In most cases, you will also want to edit unwanted "dead space" at the beginning and ending of your recording. This is known as "trimming silence".

- To get the best results from your recording efforts, Lucent Technologies recommends that you use a microphone of professional quality. This is especially important if you want to convert your recordings to a format that the target voice response system can use.
- The INTUITY CONVERSANT system plays only specially formatted ADPCM files sampled at 8 kHz. When you use the recording application included with Voice@Work, it automatically records at an 11.025-kHz sampling rate and converts it during the code generation process to a form the target system can use. If you import *.wav files into your application, record them at the same rate to avoid conversion problems.

The Phrase Table Editor

The Phrase Table Editor is used to:

- Create new phrase tables.
- Edit existing phrase tables.

Opening the Phrase Table Editor Window

To open the Phrase Table Editor window, do the following:

1. Open the Phrases Manager. See “Opening the Phrases Manager” above for the procedure.
2. Right-click anywhere in the Phrases Manager window *other than* the name bar. Then, from the popup menu, do one of the following:
   - Select **New Phrase Table...**
   - Select **Edit Phrase Table...**

   Voice@Work displays the Phrase Table Editor window (Figure 71).

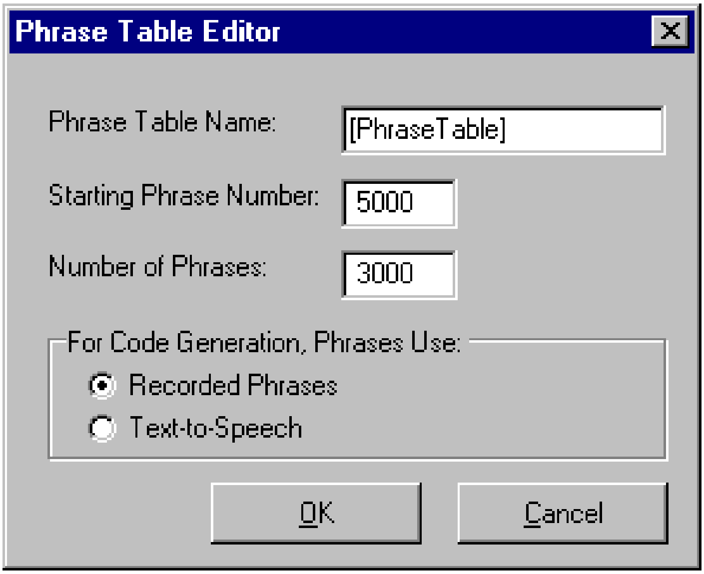


Figure 71. The Phrase Table Editor Window

Table 49 describes the elements of the Phrase Table Editor window.

Table 49. Phrase Table Editor Window Elements

Element	Action
Phrase Table Name field	Allows you to type a new phrase table name here or check the existing name. Phrase table names must be 8 characters in length or less. Voice@Work truncates names with more than 8 characters to the first 8 characters. **NOTE:** Do not put square brackets [ ] around a new name. Voice@Work automatically inserts the brackets when it creates a new table.
Starting Phrase Number field	Allows you to enter a starting phrase number for this phrase table. By default, Voice@Work assigns the first unallocated free phrase number.

Continued on next page

Table 49. Phrase Table Editor Window Elements — *Continued*

Element	Action
Number of Phrases field	Allows you to enter the number of phrases you want to allocate for this phrase table.
For Code Generation, Phrases Use field	Allows you to select whether phrases should be generated for playback on the target system using recorded speech or Text-to-Speech.

Creating Phrase Tables

To create a new phrase table, do the following:

1. Open the Phrase Table Editor. See "Opening the Phrase Table Editor Window" above for the procedure.
2. Type a name for the new phrase table in the **Phrase Table Name** field.

   NOTE:
   Phrase table names must be 8 characters in length or less. Voice@Work truncates names with more than 8 characters to the first 8 characters.

3. Enter a **Starting Phrase Number** for the phrase table.

   By default, Voice@Work places the next unallocated phrase number in this field.

4. Enter the **Number of Phrases** you want to allocate to this phrase table.

   The default for new phrase tables is 100 phrases.

5. Select whether you want to use **Recorded Phrases** or **Text-to-Speech** for the phrase table during the code generation process.

   Voice@Work defaults to **Recorded Speech** for code generation.

   NOTE:
   Text-to-Speech is available only if you have purchased it as an option on your INTUITY CONVERSANT system. In many cases, you will find it preferable to use recorded speech.

   To assign Text-to-Speech, select the **Text-to-Speech** option in the Phrase Table Editor window.

6. Click **OK**.

Editing Phrase Tables

To edit existing phrase tables, do the following:

1. Open the Phrase Table Editor. See "Opening the Phrase Table Editor Window" above for the procedure.
2. Edit the phrase table fields as desired.
3. Click **OK**.

Importing Text

A phrase table record consists of two fields: the phrase number and the phrase text. When creating text files to import as phrases into Voice@Work, you may create them with or without phrase numbers.

- If you include phrase numbers, there should be a one-to-one correspondence between the phrase numbers and the phrase text when you construct your text file.
- If you do not include phrase numbers, you must construct your files so as to separate your phrases clearly.

When importing files, keep the following considerations in mind:

- If you do not include phrase numbers in your source text, Voice@Work assigns phrase numbers automatically, beginning with the first unallocated phrase number.
- If you include phrase numbers in your source text:
  - — Do not use numbers less than 5000. Voice@Work defaults to phrase numbers equal to or greater than 5000.

    CAUTION:
    You can use a number less than 5000, but if you do so, be careful. Some Enhanced Basic Speech tables use a high number of phrases, and by choosing a number less than 5000, you run the risk of overwriting phrases in that table.

  - — Use a starting phrase number as close as possible to the last defined phrase number in the table. This is because Voice@Work creates blank phrases to fill in the missing numbers.

    For example, suppose that the last phrase number in the table is 5005, and you select a starting phrase number of 6000. Voice@Work then creates blank phrases numbered 5506–5999 before creating the phrases you want and populating them with the imported phrase text.

— Decide ahead of time what kind of separator you will use and be consistent with it. If you use more than one kind of separator, Voice@Work is not able to correctly create and populate the new phrases.

One advantage of using phrase numbers in your source text file is that you can replace noncontiguous phrases in a phrase table.

For example, suppose you have a list of eleven phrases, numbered 5000–5010, and you want to replace only three of them, 5003, 5006, and 5010, with new phrases. By creating a text file containing only those three phrase numbers and their corresponding phrase text, you can import that text file and replace only those three phrases.

The Phrase Text Import Dialog Box

You can use Voice@Work to import predefined text as a phrase from outside Voice@Work into the Phrases Manager. Choose from the following options to import text:

- From the clipboard to an existing phrase
- From the clipboard to a new phrase
- From a file to an existing phrase
- From a file to a new phrase

Importing Text from the Clipboard into an Existing Phrase

To import text from the clipboard to an existing phrase, do the following:

1. Copy the phrase you want to impot to the clipboard using the **Copy** command.
2. Double-click the phrase you want to replace in the Phrases Manager.

   Voice@Work displays the Phrase Editor window (Figure 70).
3. Highlight the text in the **Phrase Text** field.
4. Press CONTROL V.

Importing Text from the Clipboard into a New Phrase

To import text from the clipboard to a new phrase, do the following:

1. Copy the phrase you want to import to the clipboard using the **Copy** command.
2. Right-click in the Phrases Manager window, anywhere *other than* the name bar, and select **Import Phrases from Text...** from the popup menu.

   Voice@Work displays the Phrase Text Import dialog box (Figure 72).

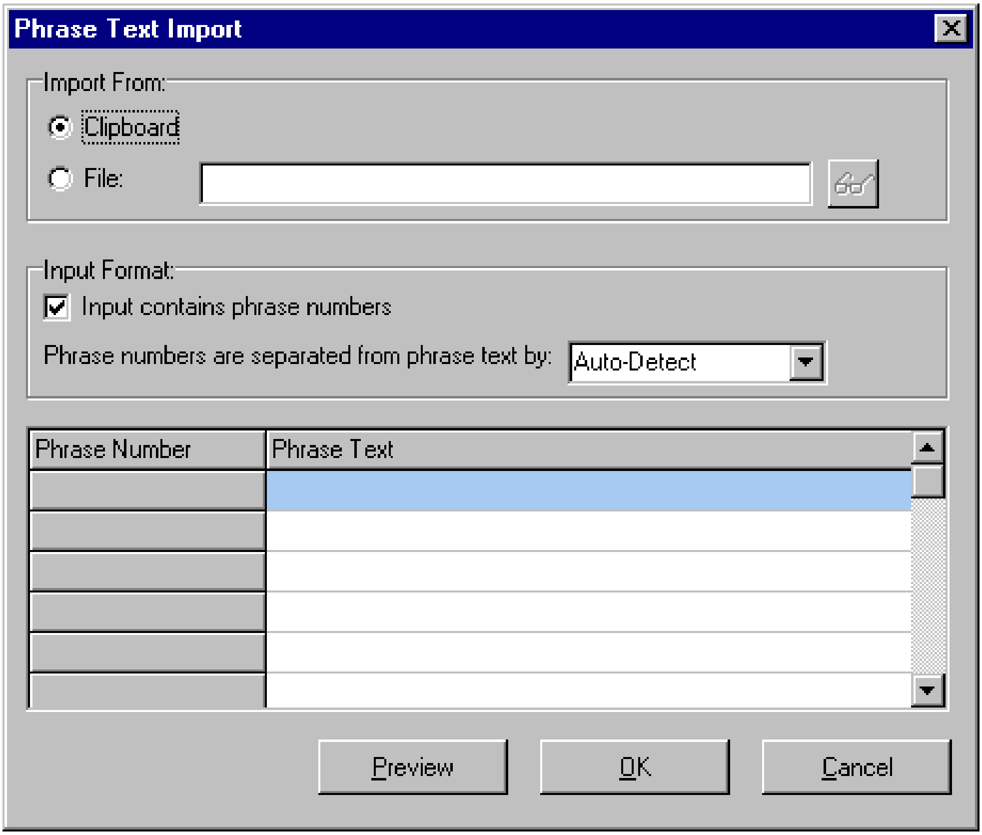


Figure 72. Phrase Text Import Dialog Box

3. Select **Clipboard** in the **Import From:** box.
4. If the text you are importing contains phrase numbers, check the **Input contains phrase numbers** checkbox. If it does not, make sure this box is unchecked. See "Importing Text" above for additional information.

5. If you checked the **Input contains phrase numbers** checkbox, select the type of phrase number/phrase text separator you used from the **Phrase numbers are separated from phrase text by:** pulldown menu. Choices include:

   - Auto-Detect – Checks for all types of separators, such as white space, commas, or tabs, and places the separation at the appropriate places
   - White Space – Looks for spaces in text files between numbers and letters and places the separation where it finds the spaces
   - Commas – Looks for commas in text files and places the separation where it finds them
   - Tabs – Looks for tab markers between numbers and letters and places the separation where it finds them

6. (Optional) To preview the phrases as they will appear in the Phrases Manager, click the **Preview** button.
7. Click **OK**.

Importing Text from a File to an Existing Phrase

To import text from a file to an existing phrase, do the following:

1. Use a text editor to create the phrase text you want to import, and save the file.

   Be sure to include the phrase numbers for the phrases you want to replace. See “Importing Text” above for additional information.

2. Right-click in the Phrases Manager window anywhere *other than* the name bar, and select **Import Phrases from Text...** from the popup menu.

   Voice@Work displays the Phrase Text Import dialog box (Figure 73).

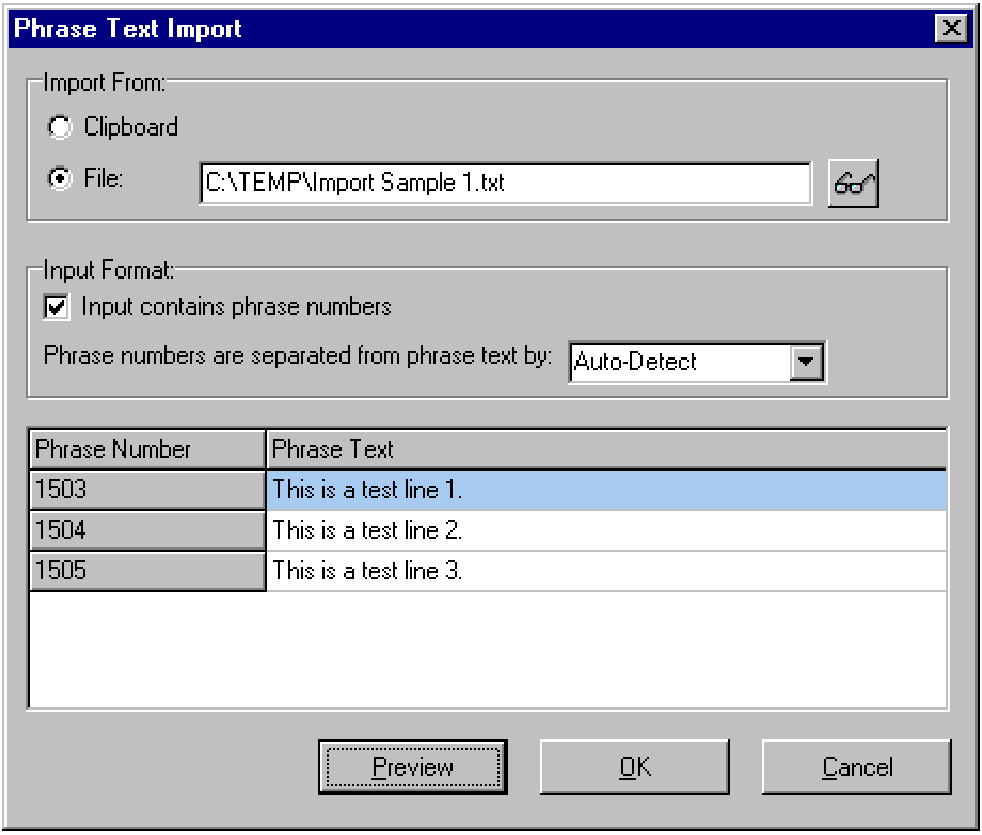


Figure 73. Phrase Text Import Dialog Box with Existing Phrases

3. Select **File:** in the **Import From:** box.
4. Do one of the following:
   - Use the **Browse Files**  button to locate and select the file you want to import.
   - Type in the full pathname of the file you want to import.
5. Check the **Input contains phrase numbers** checkbox.
6. Select the type of phrase number/phrase text separator you used from the **Phrase numbers are separated from phrase text by:** pulldown menu. Choices include:
   - Auto-Detect – Checks for all types of separators, such as white space, commas, or tabs, and places the separation at the appropriate places
   - White Space – Looks for spaces in text files between numbers and letters and places the separation where it finds the spaces

- Commas – Looks for commas in text files and places the separation where it finds them
- Tabs – Looks for tab markers between numbers and letters and places the separation where it finds them

7. (Optional) To preview the phrases as they will appear in the Phrases Manager, click the **Preview** button.
8. Click **OK**.

   Voice@Work displays a warning box and asks whether you want to start numbering at 5000, the default that Voice@Work expects, or continue with your choice.
9. Select **Yes**.

   Voice@Work displays a warning box for each phrase you are trying to overwrite, and asks if you want to overwrite the existing phrases.
   - To overwrite all targeted phrases, select **All**.
   - To overwrite only selected phrases, answer **Yes** only to those phrases you want to overwrite.
   - If there are any additional phrases at the end, Voice@Work automatically assigns them new phrase numbers and populates the table with them.

Importing Text from a File to a New Phrase

To import text from a file to a new phrase, do the following:

1. Use a text editor to create the phrase text you want to import, and save the file. See "Importing Text" above for additional information.
2. Right-click in the Phrases Manager window anywhere *other than* the name bar, and select **Import Phrases from Text...** from the popup menu.

   Voice@Work displays the Phrase Text Import dialog box (Figure 73).
3. Select **File:** in the **Import From:** box.
4. Do one of the following:
   - Use the **Browse Files**  button to locate and select the file you want to import.
   - Type in the full pathname of the file you want to import.
5. If the text you are importing contains phrase numbers, check the **Input contains phrase numbers** checkbox. If it does not, make sure this box is unchecked.

6. If you checked the **Input contains phrase numbers** checkbox, select the type of phrase number/phrase text separator you used from the **Phrase numbers are separated from phrase text by:** pulldown menu. Choices include:
   - Auto-Detect – Checks for all types of separators, such as white space, commas, or tabs, and places the separation at the appropriate places
   - White Space – Looks for spaces in text files between numbers and letters and places the separation where it finds the spaces
   - Commas – Looks for commas in text files and places the separation where it finds them
   - Tabs – Looks for tab markers between numbers and letters and places the separation where it finds them
7. (Optional) To preview the phrases as they will appear in the Phrases Manager, click the **Preview** button.
8. Click **OK**.

   Voice@Work automatically assigns the imported phrases new phrase numbers and populates the table with them.

Create Default Phrases Dialog Box

To create default phrases and use them to populate phrase entries in the Phrases Manager, do the following:

1. Right-click in the Phrases Manager anywhere *other than* on the name bar.
2. Select **Fill Phrases with Default...** from the popup menu.

   Voice@Work displays the Create Default Phrases window (Figure 74).

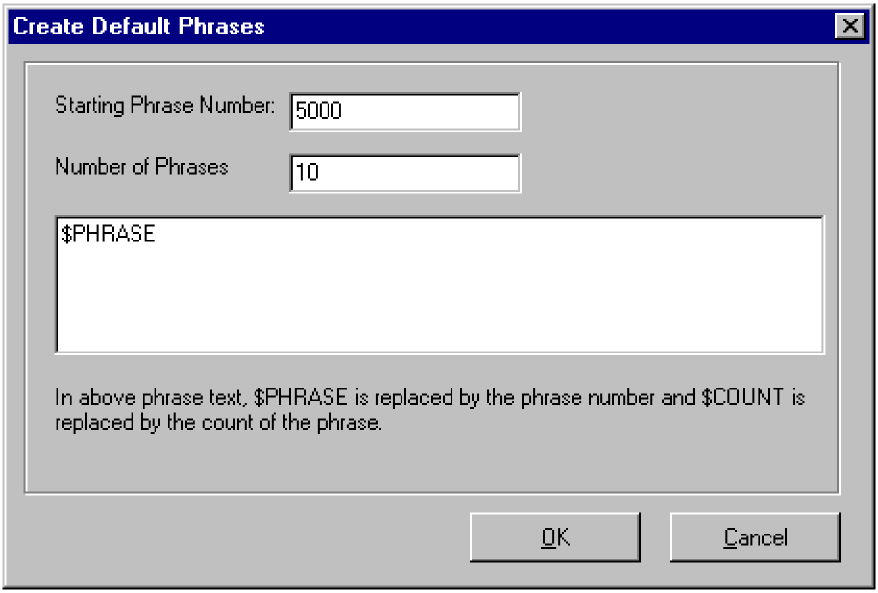


Figure 74. Create Default Phrases Window

3. Enter the phrase number you want to start with in the **Starting Phrase Number:** field.

   Voice@Work defaults to the first unallocated phrase number. If you want to start with a different number, delete the current number and replace it with the number you want.

If you have designated a phrase number that is already in use and then click **OK**, the following message appears (Figure 75):

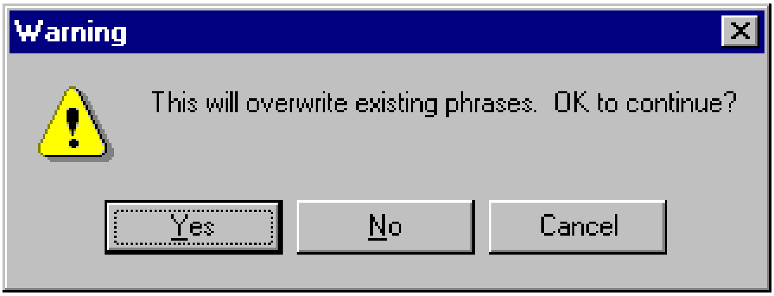


Figure 75. Phrase Number Warning

- To continue, overwriting existing phrases, click **Yes**.
- To return to the Create Default Phrases dialog box and change the starting number, click **No** or **Cancel**.

4. Enter the number of new phrases you want to create in the **Number of Phrases:** field.
5. Type the default text you want for the phrases in the **Phrase Text** field.
6. Click **OK**.

   Voice@Work creates the number of new phrases you requested and populates each phrase with the default text.

Working with Prompts

19

Overview

Prompts are used in Voice@Work as components of the Interactive Voice Response (IVR) nodes. Prompts are used to instruct callers, to get callers to select choices, and to obtain information from callers.

Prompts are most commonly composed of phrases. They may also include variables, phrase breaks, or controls.

Prompts are grouped together and displayed in the Prompts Manager.

Purpose

The purpose of this chapter is to provide information about defining and using prompts in Voice@Work applications. Topics include:

- The Prompts Manager
- Opening the Prompts Manager
- Creating New Prompts
- Renaming Prompts
- Right-Click Menu Options for the Prompts Manager
- The Prompt Editor

The Prompts Manager

The Prompts Manager is used to perform many of the actions related to creating and using prompts in Voice@Work applications. From the Prompts Manager, you can:

- Create new prompts.
- Rename existing prompts.
- Edit existing prompts.
- Use the right-click menu to perform a variety of other operations on prompts.

The Prompts Manager displays the names and the first line of data for all prompts defined for the current application.

Opening the Prompts Manager

To open the Prompts Manager, do one of the following:

- From the **View** menu, select **Prompts**.
- Click the  button on the Toolbar.
- Open the Globals Manager and locate the Prompts Manager, if it is docked to the Globals Manager.

Voice@Work displays the Prompts Manager window (Figure 76).

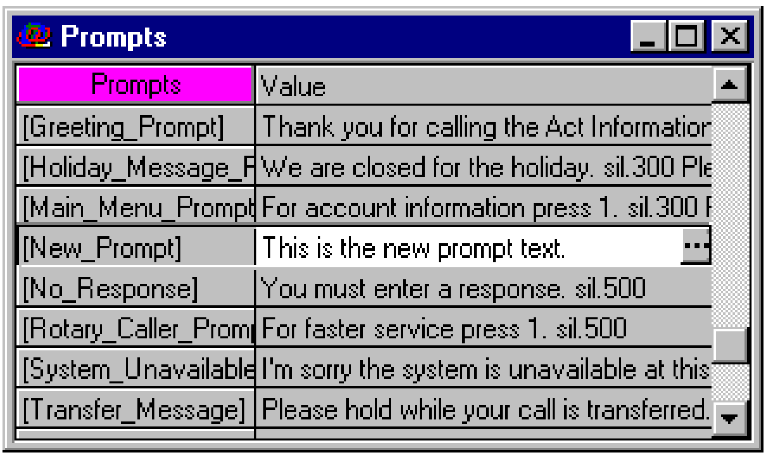


Figure 76. Prompts Manager Window

The left column lists all prompts defined for the current application by name in alphabetic order. The right (Value) column displays the phrases or other elements, such as variables, assigned to the prompts.

Creating New Prompts

There are two ways you can create a new prompt in Voice@Work:

- Using the Prompts Manager
- Using the Node Inspector

Using the Prompts Manager

To create a prompt using the Prompts Manager, do the following:

1. Open the Prompts Manager. See "Opening the Prompts Manager" above for the procedure.
2. Right click in the Prompts Manager grid anywhere *other than* on the name bar.
3. Select **New...** from the popup menu.

   Voice@Work displays a dialog box and asks you for a name for the new prompt (Figure 77).

Figure 77. Name New Prompt Window

4. Enter the name for the new prompt in the space provided.

   When naming prompts, keep the following in mind:

   - Names can be any length. However, keep prompt names as short as possible.
   - Do not use spaces in prompt names. Voice@Work ignores and deletes spaces when creating the prompt name.
   - Prompt names are case sensitive. Voice@Work preserves cases as you assign them when naming the prompt. The one exception is that the first letter of a name is always converted to upper case, regardless of how you type it in. For example, if you typed in **new_ Prompt** as the prompt name, Voice@Work converts it to **New_ Prompt** when creating the prompt.

     NOTE:
     Although Voice@Work preserves cases when saving names and transfers them to the target system with cases preserved, you cannot have two names with the same letters. For example, if you have a prompt called **New_Prompt**, you cannot also create a prompt called **NEW_PROMPT**.

   - Begin prompt names with alphanumeric characters only. Voice@Work ignores and deletes all other initial characters when creating the prompt name.

     NOTE:
     You can use the underscore (_) symbol after the initial character of a prompt name.

5. Click **OK**.

   Voice@Work displays the Prompt Editor window (Figure 79).

6. Define the prompt attributes as desired in the Prompt Editor. See "The Prompt Editor" below.

7. Click **OK**.

   Voice@Work creates the new prompt and places it in the Phrases Manager in alphabetical order by prompt name.

   If you typed in text as part of the prompt definition, Voice@Work displays the New Phrase Assignments window (Figure 81). For details about using the New Phrase Assignments window, see "The New Phrase Assignments Window" below.

Using the Node Inspector

To create a prompt using the Node Inspector, do the following:

1. Open the Node Inspector. See Chapter 26, "The Node Inspector", for the procedure.
2. Double-click the line for the prompt you want to create.

   Voice@Work displays a dialog box and asks you for a name for the new prompt (Figure 77).
3. Enter the name for the new prompt in the space provided. Follow the naming conventions described in Step 4 of "Using the Prompts Manager" above.
4. Click **OK**.

   Voice@Work displays the Prompt Editor window (Figure 79).
5. Define the prompt attributes in the Prompt Editor. See "The Prompt Editor" below.
6. Click **OK**.

   Voice@Work creates the new prompt and places it in the Phrases Manager in alphabetical order by prompt name.

   If you typed in text as part of the prompt definition and if those phrases do not already exist in the Phrases Manager window, Voice@Work displays the New Phrase Assignments window (Figure 81). For details about using the New Phrase Assignments window, see "The New Phrase Assignments Window" below.

Renaming Prompts

To rename an existing prompt without altering its contents, do the following:

1. Open the Prompts Manager window. See "Opening the Prompts Manager" above for the procedure.
2. Right-click the name of the prompt whose name you want to change.
3. Select **Rename...** from the popup menu.

   Voice@Work displays a dialog box and asks you for a new name for the variable (Figure 78).

Figure 78. Rename Prompt Window

4. Enter the new name for the prompt in the space provided. Follow the naming conventions for prompts described in Step 4 of "Using the Prompts Manager" above.
5. Click **OK**.

   The name of the prompt changes in the Prompts Manager window and anywhere the prompt is used or referenced within the application.

Right-Click Menu Options for the Prompts Manager

When you right-click in the Prompts Manager window anywhere *except* the name bar, Voice@Work displays a popup menu. Table 50 describes the options available from the Prompts Manager right-click popup menu.

Table 50. Prompts Manager Right-Click Menu Options

Menu Option	Action
New...	Allows you to create a new prompt. See "Creating New Prompts" above.
Delete	Deletes the selected prompt from the Prompts Manager, if it is not in use. **NOTE:** If you used the prompt in your application, you must first remove it from the application before you can delete it.
Rename...	Allows you to rename an existing prompt. See "Renaming Prompts" above.
Find Uses of	Opens the Find window, searches for Uses Of the selected prompt, and presents the results of the search in the **Find window results** field. See "The Find... Feature" section in Chapter 6, "The Edit Menu".
Sharing...	Inactive in the current release of Voice@Work.
Stay on Top	Available only when the Prompts Manager is undocked. Toggles on and off. When checked (toggled on), it causes the Prompts Manager window to remain on top of other open Voice@Work windows. **NOTE:** If you have more than one resource manager window toggled to "Stay on Top", the window actually on top will be the last window you clicked in. All windows toggled to "Stay on Top", however, will stay on top of any other Voice@Work windows that are open.
Edit Prompt...	Opens the Prompt Editor so you can edit the selected prompt. See "The Prompt Editor" below.

The Prompt Editor

The Prompt Editor allows you to define, edit, and select the attributes and elements of a prompt.

From the Prompt Editor, you can:

- Choose an alternate (list format) view of the elements of a prompt.
- Create a new phrase within a prompt.
- Insert an existing phrase into a prompt.
- Edit a phrase in a prompt.
- Insert a variable into a prompt.
- Insert a phrase break into a prompt.
- Insert a control into a prompt.
- Delete a prompt element—such as a phrase, variable, or control—from a prompt.
- Insert a new line—alternate view only.
- Choose to use all selected records for a prompt—Automenu node only.
- Use the right-click popup menus to perform various other operations on prompt attributes and elements.

Opening the Prompt Editor

To open the Prompt Editor, do the following:

1. Open the Prompts Manager window. See “Opening the Prompts Manager” above for the procedure.
2. Right-click on the name of the prompt you want to edit.
3. Select **Edit Prompt...** from the popup menu.

Choosing an Alternate View for the Prompt Text Field

When you first open the Prompt Editor window, Voice@Work displays the window like the example in Figure 79, called here the “normal view”.

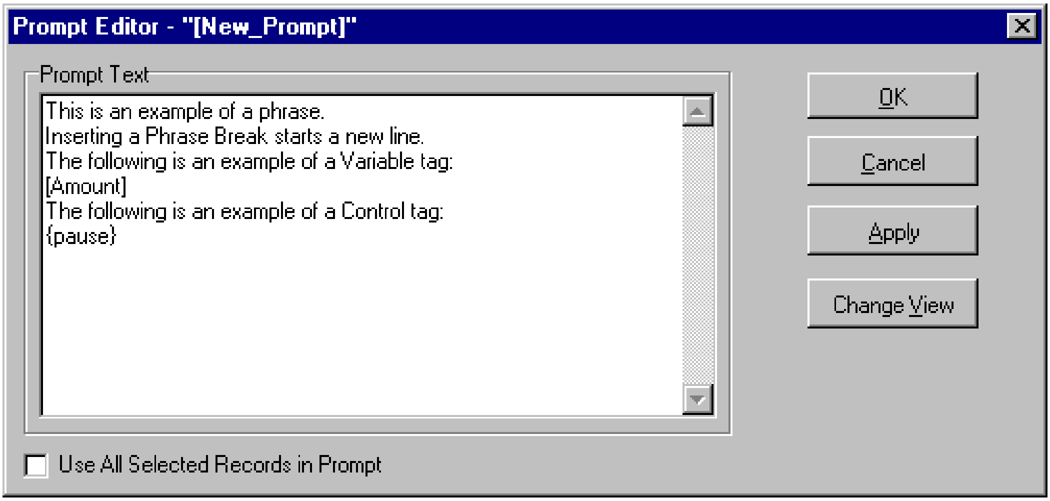


Figure 79. Prompt Editor Window – Normal View

To change to the alternate view of the contents field, click the **Change View** button (Figure 80).

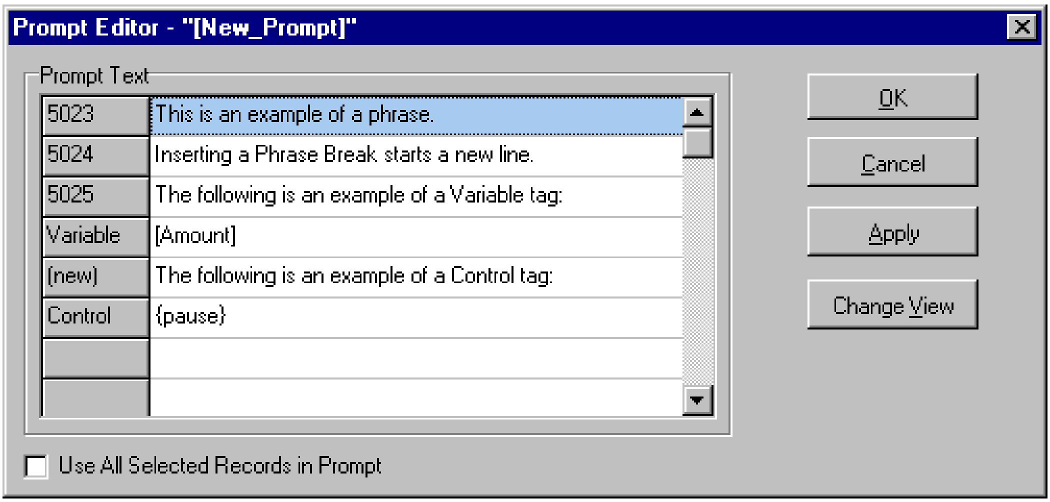


Figure 80. Prompt Editor Window – Alternate View

One advantage of the alternate view is that it tags each element of the prompt. These tags let you see exactly how many elements are used in the prompt and what type they are.

NOTE:
You can also use the alternate view with the **Use All Selected Records in Prompt** checkbox checked (see the "The Use All Selected Records in Prompt Option (Prompt Editor)" section below).

Creating a New Phrase within a Prompt

To create a new phrase within a prompt, type the new phrase directly in the **Prompt Text** field (Figure 79).

NOTE:
You cannot create phrases in the alternate view.

When creating new phrases in the Prompt Editor, keep the following in mind:

- Every time you press the ENTER key or select **Insert a Phrase Break**, you start a new phrase with its own phrase number.
- Voice@Work displays the New Phrase Assignment window and automatically assigns phrase numbers to new phrases when you click the:
  - **Apply** button
  - **OK** button
- Variable names appear enclosed in square brackets [ ].
- Controls appear enclosed in curly brackets { }.

The New Phrase Assignments Window

Voice@Work displays the New Phrase Assignments window when you create a new phrase or edit an existing phrase within a prompt.

When you create a new phrase within a prompt and click either the **Apply** button or the **OK** button, Voice@Work displays the New Phrase Assignments window (Figure 81).

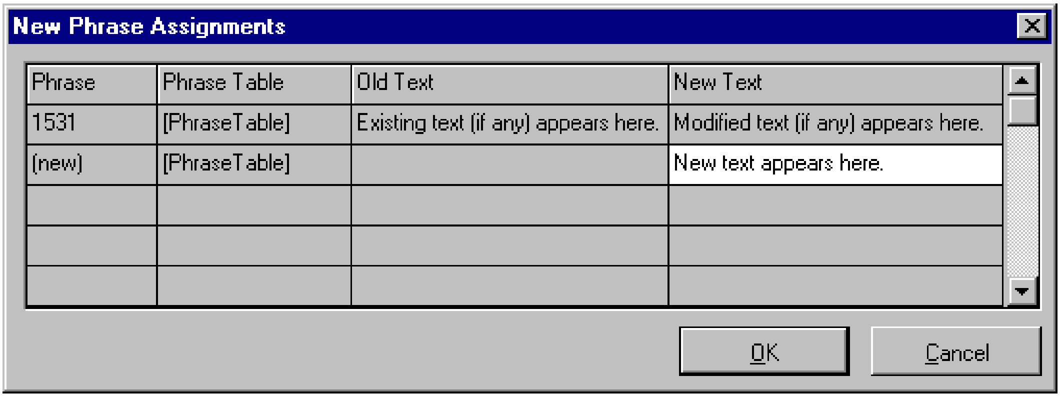


Figure 81. New Phrase Assignments Window

The New Phrase Assignments window has four columns. Table 51 describes each column.

NOTE:
You cannot edit the data in any of these fields. You can only select the phrase number and/or phrase table you want from the pulldown menus.

Table 51. New Phrase Assignments Window Columns

Column	Function/Possible Actions
Phrase	From the pulldown menu, select one of the following: ■ [new] — Creates a new phrase and automatically assigns a phrase number to it when you close the window ■ A phrase number (displayed) — Exchanges any old text with the new or modified text **NOTE:** Any old text is displayed in the Old Text column. The text that replaces it is displayed in the New Text column.

Continued on next page

Table 51. New Phrase Assignments Window Columns — *Continued*

Column	Function/Possible Actions
Phrase Table	If you are creating a new phrase, select from the pulldown menu the phrase table you want to assign the phrase to. To edit or modify an existing phrase, select from the pulldown menu the phrase table that contains the phrase you want to edit or modify. You must do this *before* selecting the phrase to edit or modify.
Old Text	If you are editing or modifying an existing phrase, this column displays the text that will be edited or modified. If you are editing or modifying a new phrase, this column is empty. **NOTE:** You cannot alter the contents of this column. This column is for display only.
New Text	This column displays the text to be assigned to the phrase. **NOTE:** You cannot alter the contents of this column. This column is for display only.

Using the New Phrase Assignments Window to Create a New Phrase

To create the new phrase and assign it a phrase number, do the following:

1. Do one of the following:
   - Select **[new]** from the Phrase pulldown menu.
   - From the Phrase pulldown menu, select the number of the phrase you want to replace.
2. From the Phrase Table pulldown menu, select the Phrase Table you want to assign the new phrase to.
3. To confirm your choices and close the dialog box, click the **OK** button.

   Voice@Work displays the new phrase in the Phrases Manager window.

Inserting an Existing Phrase into a Prompt

Voice@Work offers two methods for inserting existing phrases into prompts.

Method A:

1. In the *normal* view (Figure 79), place the cursor where you want the phrase to go.
2. Do one of the following:
   - Right-click in the **Prompt Text** field and select **Insert Phrase...** from the popup menu
   - Press CONTROL P

   Voice@Work displays the Insert Phrase dialog box (Figure 82). This window displays all the phrases that exist for the active phrase table.

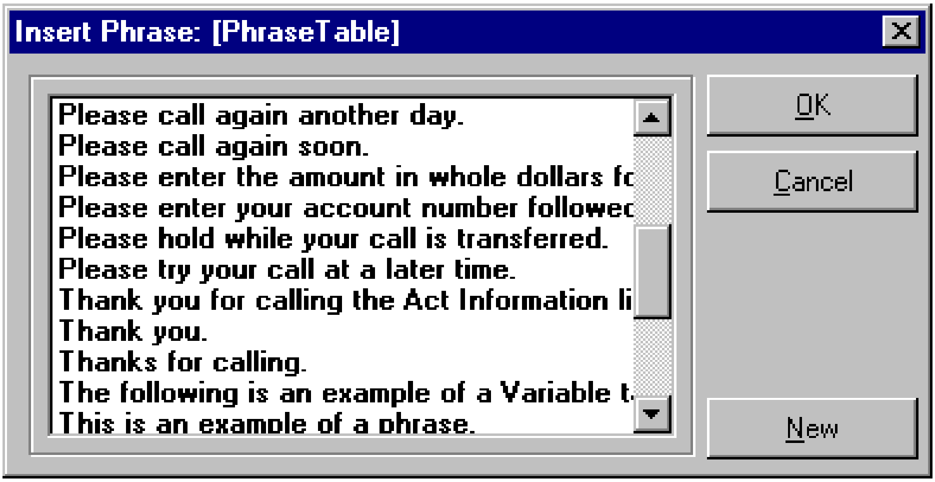


Figure 82. Insert Phrase Window

3. Highlight the phrase you want from the list.
4. Click **OK**.

   The phrase is inserted into the **Prompt Text** field at the point where you placed the cursor.

Method B:

1. Open the Phrases Manager window. See "Opening the Prompts Manager" above for the procedure.
2. Do one of the following:
   - Place the cursor where you want the phrase to go in the *normal* view of the Prompt Editor. Drag-and-drop the selected phrase, using the phrase number field to drag from the Phrases Manager window to anywhere in the **Prompt Text** field.
   - Select the *alternate* view of the Prompt Editor. Drag-and-drop the selected phrase, using the phrase number to drag from the Phrases Manager window to the line of the Prompt Editor **Prompt Text** field where you want the phrase to go.

     NOTE:
     If you drop the phrase on top of a line already containing an element in the alternate view, the new phrase replaces what is already on that line. To insert a phrase before the last line, first insert a new line in the **Prompt Text** field where you want the phrase to go. See "Inserting a New Line (Alternate View)" below for the procedure.

Editing a Phrase in a Prompt

You can edit a phrase in Voice@Work much the same way you would edit text in any basic word processor. You can change phrase text only in the normal view of the **Prompt Text** field.

Select the text you want to edit by highlighting it. Once the text is highlighted, the following functions on the right-click popup menu become active (see "Prompt Editor Right-Click Menus" above):

Cut CONTROL X

Copy CONTROL C

Paste CONTROL V

Delete DEL

Select All CONTROL A

In addition to typing in new text, you can use any of these standard Windows functions.

If you make changes to phrase text in the Phrases Manager, those changes are automatically reflected in the Prompt Editor.

If you make changes to phrase text in the Prompt Editor, Voice@Work treats the phrase as a new phrase and tries to assign a new phrase number to it when you click the **Apply** or **OK** buttons. Finish editing the phrase in the New Phrase Assignments window. See "The New Phrase Assignments Window" above for the procedure.

Inserting a Variable into a Prompt

Any variable can be used as part of a prompt. Variables can be used to speak stored information to a caller.

Voice@Work offers two methods for inserting a variable into a prompt:

Method A:

1. In the *normal* view, place the cursor where you want the variable to go.
2. Do one of the following:
   - Right-click in the **Prompt Text** field and select **Insert Variable...** from the popup menu
   - Press CONTROL B
   - Type a left square bracket character **[**.

   Voice@Work displays the Insert Variable dialog box (Figure 83).

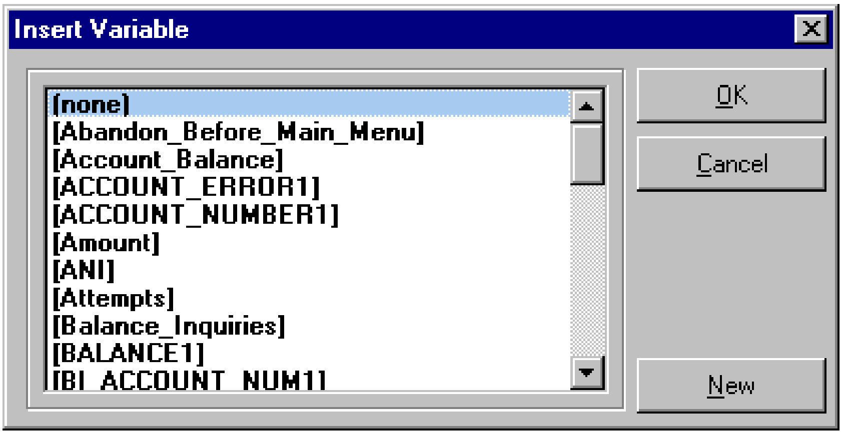


Figure 83. Insert Variable Dialog Box

3. Highlight the variable you want from the list.
4. Click **OK**.

   The variable is inserted into the **Prompt Text** field at the point where you placed the cursor.

Method B:

1. Open the Variables Manager window. See "Opening the Prompts Manager" above for the procedure.
2. Do one of the following:
   - In the *normal* view of the Prompt Editor, place the cursor where you want the variable to go. Drag-and-drop the selected variable from the Variables Manager to anywhere in the **Prompt Text** field.
   - Select the *alternate* view of the Prompt Editor. Drag-and-drop the selected variable from the Variables Manager to the line of the Prompt Editor **Prompt Text** field where you want the variable to go.

NOTE:
If you drop the variable on top of a line already containing an element in the alternate view, the variable replaces what is already on that line. To insert a variable before the last line, first insert a new line in the **Prompt Text** field where you want the variable to go, using the right-click popup menu. See "Inserting a New Line (Alternate View)" below for the procedure.

Inserting a Phrase Break

A Phrase Break signals the end of a phrase. You can use phrase breaks to divide long phrases into shorter phrases or start a new line in the Prompt Editor to add another component.

Voice@Work offers three ways to insert a phrase break in the Prompt Editor:

- Place the cursor where you want the break, and then press the ENTER key.
- Right-click in the *normal* view of the Prompt Editor, and then select **Insert Break...**
- Press CONTROL I in the *normal* view of the Prompt Editor.

Inserting a Control into a Prompt

Controls in Voice@Work are used to:

- Insert silences of specific lengths.
- Adjust the rate at which Text-to-Speech (TTS) speaks the phrase text.
- Select risky modes for TTS phrases.

Types of Controls

There are two types of prompt controls in Voice@Work:

- Pauses
- Text-to-Speech Options

Pauses

Pauses are programmed intervals of silence, used mostly for making TTS (Text-to-Speech) responses sound more natural, though they may also be used with Recorded Speech. Intervals offered are:

0.05 sec silence

0.1 sec silence

0.2 sec silence

0.3 sec silence

0.4 sec silence

0.5 sec silence

pause (indefinite length)

Text-to-Speech Options

Text-to-Speech offers a variety of options for how it may be used. Table 52 describes these options.

Table 52. Controls: Text-to-Speech (TTS) Options

TTS Option	Description/Comments
tts	Speaks the text at a normal speed.
tts address	Determines how certain abbreviations, such as "St." or "Dr.", are spoken. EXAMPLE: "St." is spoken as "street," not "saint." "Dr." is spoken as "drive," not "doctor."
tts fast	Speaks the text at a speed faster than normal.

Continued on next page

Table 52. Controls: Text-to-Speech (TTS) Options — ***Continued***

TTS Option	Description/Comments
tts fraction	Determines how numbers with slashes in them are spoken. EXAMPLE: The number 1/4 is spoken as "one fourth" and not as "one slash four", as it would be, for instance, as part of a date.
tts normal	Speaks the text at a normal speed.
tts phone number	Determines how numbers are spoken. Numbers are spoken as individual digits, except for "800" which, because of its common use in toll-free calls is spoken as "eight hundred" when used in the area code positions. EXAMPLE: The number **5551368** is spoken as "five five five (pause) one three six eight" and not as "five million (pause) five hundred fifty-one thousand (pause) three hundred sixty-eight"
tts proper name	Determines how certain abbreviations, such as "St." or "Dr.", are spoken. EXAMPLE: "St." is spoken as "saint," not "street." "Dr." is spoken as "doctor," not "drive."
tts slow	Speaks the text at a speed slower than normal.

NOTE:
Text-to-Speech (TTS) is only available if you have purchased it as an option on your INTUITY CONVERSANT system. In many cases, you will probably find it preferable to use Recorded Speech.

Inserting the Control

To insert a control into a prompt, do the following:

1. In the *normal* view, place the cursor where you want the control to go.
2. Do one of the following:
   - Right-click in the **Prompt Text** field and select **Insert Control...** from the popup menu.
   - Press CONTROL L.
   - Type a left curly bracket **{**.

   Voice@Work displays the Insert Control dialog box (Figure 84).

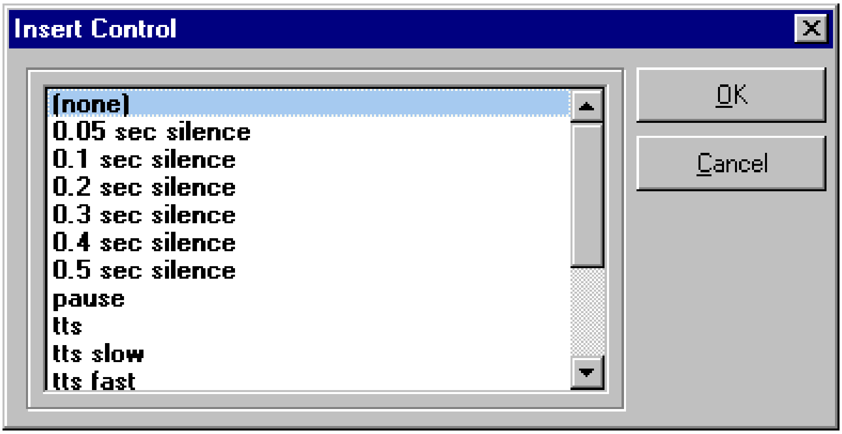


Figure 84. Insert Control Dialog Box

NOTE:
The options that appear in this window depend upon the language selected for the application. Not all silence or pause options exist for all languages.

3. Highlight the control you want from the list.
4. Click **OK**.

   Voice@Work inserts the control into the **Prompt Text** field at the point where you placed the cursor.

Deleting a Prompt Element

Voice@Work offers several options for deleting elements from a prompt:

NOTE:
Delete actions cannot be undone in the Prompt Editor, and no warning is given of this fact.

- In the *normal* view:
  - — Highlight the elements you want to delete, and then press the DEL or Backspace key.
  - — Backspace over the elements you want to delete.
- In the *alternate* view:
  - — Right-click the line containing the element, and then select **Delete Line** from the popup menu.
  - — Highlight the line containing the element, and then press the DEL key.

Inserting a New Line (Alternate View)

To insert a new line in the **Prompt Text** field of the Prompt Editor (*alternate* view):

1. Highlight the line below where you want the new line to go.

   NOTE:
   Inserting a line places it just above whatever line is highlighted.

2. Do one of the following:
   - Press the INS key.
   - Right-click the line and select **Insert Line** from the popup menu.

Prompt Editor Right-Click Menus

The right-click popup menu you see in the Prompt Editor depends on whether you are in the *normal* view or the *alternate* view.

Table 53 describes the menu options for the *normal* view in the Prompt Editor window.

Table 53. Right-Click Menu for the Normal View of the Prompt Editor Window

Menu Option	Keyboard Shortcut	Action
Cut	CONTROL X	Removes the element from the **Prompt Text** field and places it on the clipboard. This option is active only if something has already been selected in the **Prompt Text** field.
Copy	CONTROL C	Copies the element to the clipboard, leaving the original where it is. This option is active only if something has already been selected in the **Prompt Text** field.
Paste	CONTROL V	Pastes whatever is on the clipboard to the cursor’s location.
Delete	DEL	Deletes the element from the Prompt Editor. This option is active only if something has already been selected in the **Prompt Text** field. **NOTE:** The Delete action cannot be undone and does not copy the element to the system clipboard.
Select All	CONTROL A	Selects or highlights all elements in the **Prompt Text** field
Insert Variable...	CONTROL B	Allows you to insert a variable at the cursor’s location; see “Inserting a Variable into a Prompt” above
Insert Phrase...	CONTROL P	Allows you to insert a phrase at the cursor's location; see “Inserting an Existing Phrase into a Prompt” above
Insert Phrase Break	CONTROL I	Allows you to insert a phrase break at the cursor's location; see “Inserting a Phrase Break” above
Insert Control	CONTROL L	Allows you to insert a control at the cursor's location; see “Inserting a Control into a Prompt” above

Table 54 describes the menu options for the *alternate* view in the Prompt Editor.

Table 54. Right-Click Menu for the Alternate View of the Prompt Editor Window

Menu Option	Keyboard Shortcut	Action
Insert Line	INS	Allows you to insert a new line just *above* the currently selected (highlighted) line
Add Line		Adds a line at the *end* of the element list
Delete Line	DEL	Deletes the currently selected (highlighted) line **NOTE:** The Delete action cannot be undone and does not copy the element to the system clipboard.
Move Up	ALT ▲	Moves the currently selected line up one line, exchanging places with the line above it.
Move Down	ALT ▼	Moves the currently selected line down one line, exchanging places with the line below it.

The Use All Selected Records in Prompt Option (Prompt Editor)

The Use All Selected Records in Prompt option is typically used only with the Automenu node (see “The Automenu Node” under “IVR (Interactive Voice Response) Nodes” in Chapter 31, “Node Descriptions”). When selected, this option allows you to:

- Select the prompt components to be played as an **Introduction**.
- Select the prompt components to be played as the **Body**.
- Select the prompt for playing **More Records**.
- Set the **Loop Control Options** for playing **More Records**.
- Select the prompt components to be played as the **Conclusion**.

Appearance of the Prompt Editor with Use All Selected Records in Prompt Option Selected

Figure 85 shows the normal view of the Preference Editor with the Use All Selected Records in Prompt option checked.

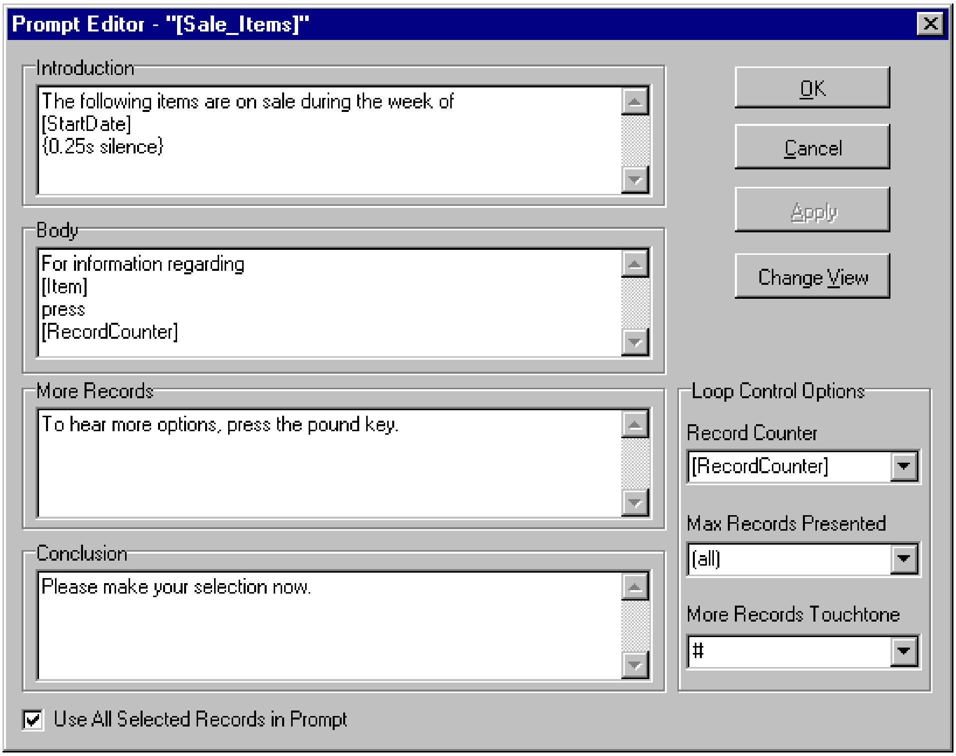


Figure 85. Preference Editor – Use All Selected Records in Prompt – Normal View

Figure 86 shows the alternate view of the Preference Editor with the Use All Selected Records in Prompt option checked.

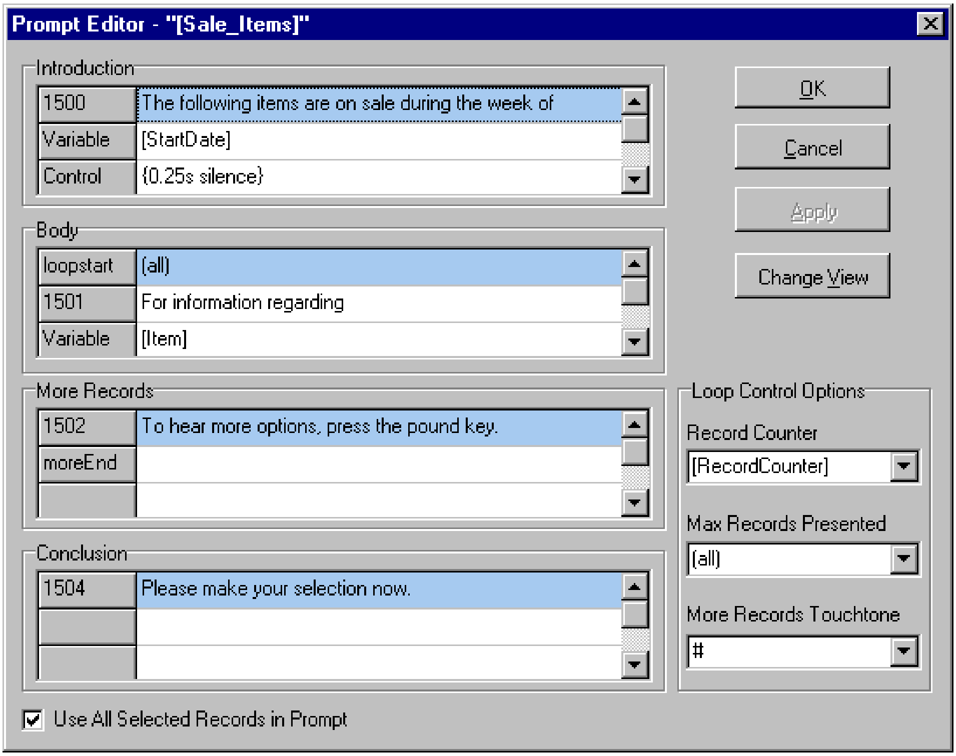


Figure 86. Preference Editor – Use All Selected Records in Prompt – Alternate View

Selecting Introduction Prompt Components

When the application first begins to execute the Automenu node, the **Introduction** plays once.

As with any other prompt, the **Introduction** can consist of any combination of prompt phrases, variables, breaks, and controls.

For more information about writing and selecting prompt components, see "The Prompt Editor" above.

Selecting Body Prompt Components

After the **Introduction** plays, the prompt plays the **Body** components. The **Body** typically consists of those phrases and variables that you want to repeat for each record found.

As with any other prompt, the **Body** can consist of any combination of prompt phrases, variables, breaks, and controls.

For more information about writing and selecting prompt components, see "The Prompt Editor" above.

Selecting More Records Prompt Components

If the Code Generation Options **Maximum number of records returned per Database query** field is set to retrieve more than the number of records set in the **Max Records Presented** field, and the application finds more than the number of records set in the **Max Records Presented** field, the **More Records** prompt plays next. **More Records** is typically a phrase used to tell the caller which key to press to hear more menu options or records.

NOTE:
The key for the caller to press is defined in **Loop Control Options** (see below).

If the application finds fewer than the number of records set in the **Max Records Presented** field, the **More Records** prompt does not play. Also, if the Code Generation Options **Maximum number of records returned per Database query** field is set to retrieve fewer than the number of records set in the **Max Records Presented** field, that determines the number of records that are retrieved, and the **More Records** prompt does not play.

Examples:

1. If you have set the **Max Records Presented** field to **(all)**, and the Code Generation Options **Maximum number of records returned per Database query** field is set to **30**, the caller is presented 9 records, and then the **More Records** prompt plays next. If the caller presses the key defined in the **More Records Touchtone** field, the next 9 records are presented and the **More Records** prompt plays again.
2. If you have set the **Max Records Presented** field to **7**, and the application finds only 4 records, then only those 4 records are presented to the caller, and the **More Records** prompt does not play.
3. If you have set the **Max Records Presented** field to **8**, but the Code Generation Options **Maximum number of records returned per Database query** field is set to **5**, then only 5 records are retrieved and presented to the caller, and the **More Records** prompt does not play.

For more information about writing and selecting prompt components, see "The Prompt Editor" above.

Setting the Loop Control Options

The **Loop Control Options** are used only in conjunction with the **More Records** prompt components (see above).

- **Record Counter** – Select from the pulldown menu the variable to use as a record counter for presenting the records to the caller. A maximum of nine records can be presented at any one time.
- **Max Records Presented** – Type in the number or select from the pulldown menu the variable that determines the number of records to make available to the caller. The default is **(all)**, which allows all records found to be presented, 9 at a time.

  NOTE:
  If you have the **Max Records Presented** field set to **5**, for example, and the Code Generation Options **Maximum number of records returned per Database query** field is set to retrieve more than 5 records, the caller has the option to be presented with all records retrieved, 5 at a time.

- **More Records Touchtone** – Type in the character key or select from the pulldown menu the variable that contains the character key to be used by the caller to hear more records. The default is the # key.

Selecting Conclusion Prompt Components

After the other prompt components have all finished playing, the **Conclusion** plays once. This is usually a simple phrase instructing the caller to make a choice and press the corresponding key.

As with any other prompt, the **Conclusion** can consist of any combination of prompt phrases, variables, breaks, and controls.

For more information about writing and selecting prompt components, see "The Prompt Editor" above.

Working with Tables 20

Overview

Database tables have two primary uses in Voice@Work applications:

- To store input, such as account numbers or order selections
- To provide information stored in a database, such as an account balance or order status

Tables can be imported (“shared-in”) using the Application Resources tool. They can also be created and populated from within Voice@Work applications.

Tables are grouped together, displayed, and accessed through the Tables Manager. They are created and edited using the Table Schema Editor and the Table Editor.

Purpose

The purpose of this chapter is to provide information on how to import, create, and edit tables in Voice@Work applications.

This chapter covers the following topics:

The Tables Manager

The Tables Manager displays brief descriptions of all tables defined for the current application. From the Tables Manager, you can:

- Import database tables from outside your application.
- Use the **Save Table As** command to make backup copies of tables.
- Create new tables.
- Rename tables.
- Edit a table's schema.
- Edit a table's data.
- Use the right-click menu to perform a variety of other operations on database tables.

Opening the Tables Manager

To open the Tables Manager, do one of the following:

- From the View Menu, select **Tables**.
- Click the  button on the Toolbar.
- Open the Globals Manager and locate the Tables Manager, if it is docked to the Globals Manager.

Voice@Work displays the Tables Manager window (Figure 87).

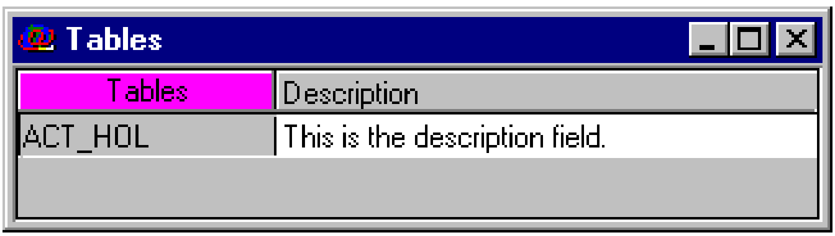


Figure 87. Tables Manager Window

Table 55 describes the elements of the Tables Manager.

Table 55. Tables Manager Elements

Element	Description
Name bar	Contains the global's name bar, distinguishable from other fields in a globals window by its magenta color
Name column	Displays the names of all tables defined for the current application
Description field column	Provides in a brief description of the table and its use

Importing Database Tables

To import a database table for use in your application, do the following:

1. Open the Application Resources tool. See Chapter 21, "The Application Resources Tool", for the procedure.
2. Click the **Import** tab.
3. Select **Table** from the Application Resources pulldown menu.

   A list of available database tables and their locations appears.
4. Do one of the following:
   - Highlight the line entry for the table you want, then click the **Import** button.
   - Double-click the line entry for the table you want.

   Voice@Work imports the selected table and displays it as an entry in the Tables Manager window. It is now available for use within your application.

Creating Database Tables

To create a new database table, do the following:

1. Open the Tables Manager. See "Opening the Tables Manager" above for the procedure.
2. Right-click in the Tables Manager anywhere *other* than on the name bar.
3. Select **New...** from the popup menu.

   A system displays a dialog box and prompts you for a name for the new table (Figure 88).

Figure 88. Name New Table Dialog Box

4. Enter the name for the table in the space provided.

   When naming tables, keep the following in mind:

   - Names must be 11 characters or less in length. Voice@Work truncates names with more characters to the first 11 characters.
   - Do not use spaces in names. Voice@Work ignores and deletes spaces when creating the actual table name.
   - Table names are not case sensitive. Whatever case you use, Voice@Work assigns upper case to all letters when creating the actual table name.
   - Table names must begin with letter characters only. After the first character, you can use other standard ASCII alphanumeric characters.

5. Click **OK**.

   The Table Schema Editor window appears. For more information about the Table Schema Editor, its use, and its appearance, see "The Table Schema Editor" below.

6. Define the fields for the table in the Table Schema Editor window. See "The Table Schema Editor" below.

7. Click **OK**.

   Voice@Work closes the Table Schema Editor window and displays the new table name as an entry in the Tables Manager (Figure 89). The table is now available for use within your application.

Figure 89. Tables Manager Window

   For a description of the elements of the Tables Manager, see Table 55.

8. (Optional, but recommended) Type a brief description of the table in the **Description** field (Figure 89).

Making Backup Copies of Tables

You might want to keep a copy of your original table, so you can restore it if needed. The best way to do this is to save the original table with a different name, using the **Save Table As** command.

To save a copy of your original table under a different name:

1. Open the Tables Manager. See "Opening the Tables Manager" above for the procedure.

2. Right-click the name of the table you want to save.

3. Select **Save Table As** from the popup menu.

   Voice@Work displays the Save Table As dialog box (Figure 90).

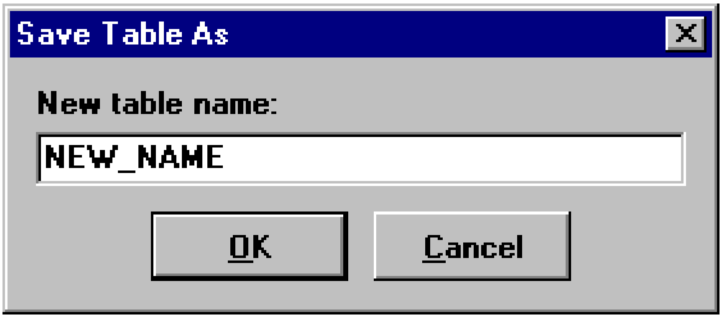


Figure 90. Save Table As Dialog Box

4. Type a new name for the table, keeping in mind the naming conventions given in Step 4 under "Creating Database Tables" above.
5. Click **OK**.

Renaming Database Tables

To rename a database table without altering its contents, do the following:

1. Open the Tables Manager. See "Opening the Tables Manager" above for the procedure.
2. Right-click in the Tables Manager anywhere *other* than on the name bar.
3. Select **Rename...** from the popup menu.

   Voice@Work displays a dialog box and prompts you for a new name for the table (Figure 91).

Figure 91. Rename Table Dialog Box

4. Enter the new name for the table in the space provided.
5. Click **OK**.

   Voice@Work changes the name of the table in the Tables Manager window (Figure 92).

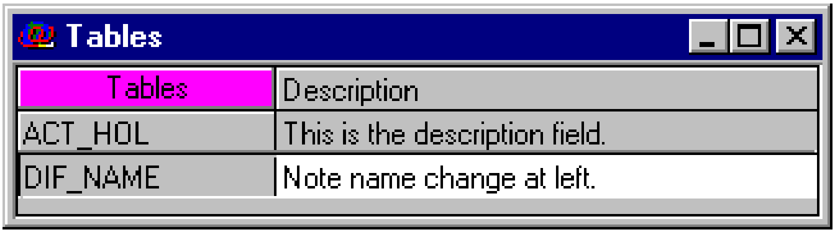


Figure 92. Tables Manager Window (Showing Table Name Change)

Editing a Table's Schema

To edit a table's schema, do the following:

1. Open the Tables Manager. See "Opening the Tables Manager" above for the procedure.
2. Right-click the name of the table whose schema you want to edit.
3. Select **Edit Table Schema...** from the popup menu.

   Voice@Work displays the Table Schema Editor window (Figure 93).
4. Edit the Table Schema fields as needed. See "The Table Schema Editor" below.
5. Click **OK**.

Editing a Table's Data

To edit a table's data, do the following:

1. Open the Tables Manager. See "Opening the Tables Manager" above for the procedure.
2. Right-click the name of the table whose data you want to edit.
3. Select **Edit Table Data...** from the popup menu.

   Voice@Work displays the Table Editor window (Figure 95).

4. Edit the table data as needed. See “The Table Editor” below.
5. Click **OK**.

Right-Click Menu Options for the Tables Manager

Table 56 describes the right-click popup menu options for the Tables Manager.

Table 56. Tables Manager Right-Click Menu Options

Menu Item	Description
New...	Allows you to create a new database table.
Delete	Deletes the database table from the Tables Manager, if not in use. **NOTE:** If you have used the table in your application, you must first remove its use from the application before you can delete it.
Rename...	Allows you to rename the selected database table without altering its contents and automatically changes each reference to it in the application.
Find Uses of	Opens the Find window, searches for Uses Of the selected table, and presents the results of the search in the **Find window results** field.
Sharing...	Makes a copy of a table (keeping a copy of the original) and places it in the Sharing directory. When a database table is marked as Shared, it can be imported into other applications using the Application Resources tool.

Continued on next page

Table 56. Tables Manager Right-Click Menu Options — ***Continued***

Menu Item	Description
Stay on Top	(Available only when the Tables Manager is undocked) Toggles on and off. When checked (toggled on), it causes the Tables Manager window to remain on top of other open Voice@Work windows **NOTE:** If you have more than one resource manager window toggled to “Stay on Top”, the window actually on top will be the last window you clicked in. All windows toggled to “Stay on Top”, however, will stay on top of any other Voice@Work windows that are open.
Save Table As...	Makes a copy of a table (keeping a copy of the original), assigns it a new name, and places it in the Tables Manager
Edit Table Schema...	Allows you to edit the schema of a table in the Tables Manager
Edit Table Data...	Allows you to edit data in a table in the Tables Manager

The Table Schema Editor

Use the Table Schema editor to define or edit a table’s schema. This editor appears whenever you create a new database table or select **Edit Table Schema...** from the Tables Manager right-click popup menu.

Opening the Table Schema Editor Window

To open the Table Schema editor window, do one of the following:

- Create a new database table. See “Creating Database Tables” in “The Tables Manager” above.
- Right-click in the Tables Manager window on the name of the table you want to edit, and select **Edit Table Schema...**

Voice@Work displays the Tables Schema editor window (Figure 93).

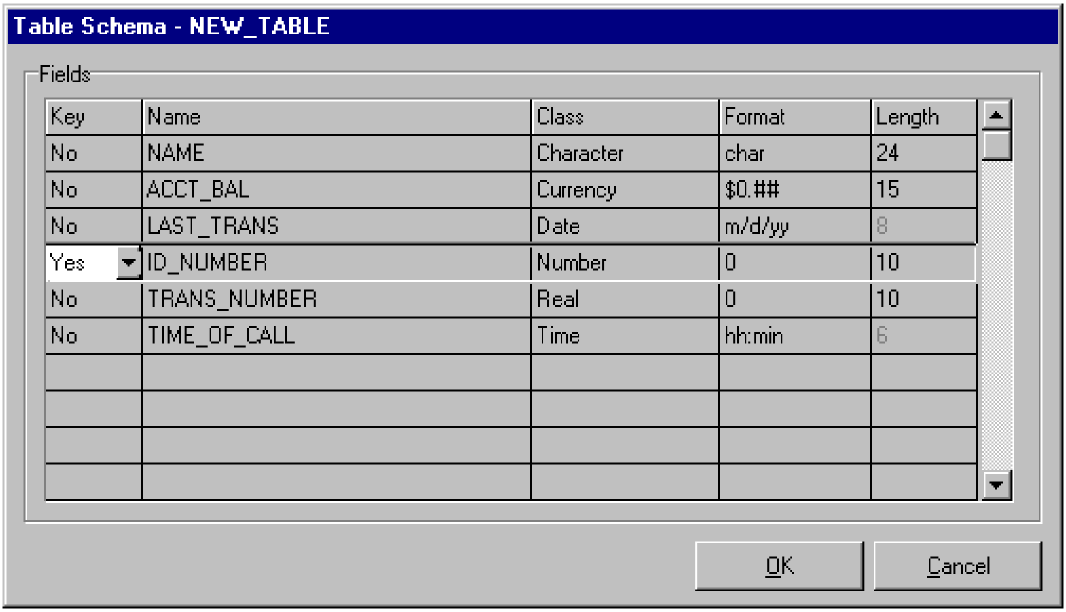


Figure 93. Table Schema Editor

Table 57 describes the fields in the Table Schema Editor and how to use them to define or edit a table's schema.

Table 57. Table Schema Editor Field Descriptions

Field	Description
Key	To designate a field as a "key" field, select **Yes** from the pulldown menu. Designating a table field as a key field makes it a unique index. Voice@Work creates only unique indexes. To create non-unique indexes, you must use ORACLE or another database program to create them. Voice@Work does allow you to designate more than one field per record as a key field. In such cases, Voice@Work utilizes all the key field entries to make the unique index.

Continued on next page

Table 57. Table Schema Editor Field Descriptions — *Continued*

Field	Description
Name	Enter the name of the field here. **NOTE:** When naming table fields keep the following in mind: ■ Table field names are limited to 25 characters. If you attempt to use more than 25 characters, the name reverts to the previous name or to UNTITLED, if there was no previous name. ■ Do not use spaces in table field names. Voice@Work eliminates them when you exit the field. ■ Table field names are not case sensitive. Whatever case you use, Voice@Work converts everything to upper case. ■ Voice@Work does not create variables for database field names. If you need the field names to be treated as variables, you must define them as such using the Variables Manager.
Class	From the pulldown menu, choose from among the following class types. Table 58 describes how Voice@Work treats each class type in a table. ■ Character ■ Currency ■ Date ■ Number ■ Real ■ Time
Format	The options in this column depend on what class is selected. Table 59 describes the options available and the defaults for each class.
Length	The values possible in this column depend on what class is selected. Table 60 describes the values and default values available for each class.

NOTE:
The date and currency formats for the following tables are all for US English only. Formats for other languages will vary. For more information, see *INTUITY™ CONVERSANT® System Speech Development, Processing, and Recognition*, or *INTUITY™ CONVERSANT® System Application Development with Script Builder*).

Table 58 describes how Voice@Work treats each **Class** type in a table.

Table 58. Table Field Class Types

Class Type	Comments
Character	Treats the input as ASCII characters or as numerical digits. Allows only standard ASCII characters as input and does not display extended ASCII characters.
Currency	Treats the input as monetary amounts.
Date	Treats the input as a date. Data must be entered using one of the following formats: ■ **yymmdd** ■ **yyyymmdd** ■ **mm-dd-yy** ■ **mm/dd/yy** ■ **mm dd yy** Voice@Work rejects other formats or values. **NOTE:** All dates are stored and used in INTUITY CONVERSANT system using the **yyyymmdd** format. ORACLE stores them using a different format (see your ORACLE documentation for details).

Continued on next page

Table 58. Table Field Class Types — ***Continued***

Class Type	Comments
Number	Treats the input as a number value or as a phrase number used to call a phrase.
Real	Treats the input as a real number with up to nine decimal places.
Time	Treats the input as a time. Data must be entered as two or three sets of digits separated by colons. Voice@Work rejects other formats or values. **NOTE:** All times are stored and used in the INTUITY CONVERSANT system using the **hhmmss** format.

The options in the **Format** column depend on what **Class** type is selected. Table 59 describes the options available and the defaults for each class type. The first option for each class is the default for that class.

Table 59. Table Class Formats

Class	Format	Description/Examples
Character	char	Treats the input as ASCII characters
	digits	Treats the input as numerical digits
Currency	$0.##	Displays the amount as dollars and cents EXAMPLE: $543.62
	$0	Rounds and displays the amount to the nearest dollar EXAMPLE: $544
Date	m/d/yy	Displays the month and day with the least possible number of digits and a two-digit year, separated by slashes For example: 5/23/98
	mm/dd/yy	Displays the month, day, and year all as two-digit numbers, separated by slashes For example: 05/23/98

Continued on next page

Table 59. Table Class Formats — *Continued*

Class	Format	Description/Examples
	m-d-y	Displays the month and day with the least possible number of digits and a two-digit year, separated by hyphens For example: 5-23-98
	mm-dd-yy	Displays the month, day, and year all as two-digit numbers, separated by hyphens For example: 05-23-98
	M d, y	Displays the month by name, the day with the least possible number of digits, and a four-digit year For example: May 23, 1998
	NOTE: All dates are stored and used in the INTUITY CONVERSANT system using the **yyyymmdd** format. The class format selects only how the date will be displayed and printed. ORACLE stores them using a different format (see your ORACLE documentation for details).	
Number	0	Treats the numeral input as a number which can be mathematically manipulated
	phrase	Treats the numeral input as a phrase number
Real	0	Rounds and displays the number to the nearest integer value For example: 5.77777777 becomes 6
	0.0	Rounds and displays the number to the nearest tenth For example: 5.77777777 becomes 5.8
	0.00	Rounds and displays the number to the nearest hundredth For example: 5.77777777 becomes 5.78
	0.000	Rounds and displays the number to the nearest thousandth For example: 5.77777777 becomes 5.778
	0.0000	Rounds and displays the number to the nearest ten-thousandth For example: 5.77777777 becomes 5.7778

Continued on next page

Table 59. Table Class Formats — *Continued*

Class	Format	Description/Examples
	0.00000	Rounds and displays the number to the nearest hundred-thousandth For example: 5.77777777 becomes 5.77778
	0.000000	Rounds and displays the number to the nearest millionth For example: 5.77777777 becomes 5.777778
Time	hh:min	Displays the time in two-digit hours and minutes For example: 8:25 in the evening (entered as 20:25) is displayed as 20:25
	hh:min:sec	Displays the time in hours, minutes, and seconds. If no second is entered, defaults to 0 seconds For example: 8:25 in the evening (entered as 20:25) is displayed as 20:25:00
	hh24:min	Same as hh:min
Time (cont.)	hh24:min:sec	Same as hh:min:sec
	hh:min AMPM	Displays the time in 12-hour format, appended with an AM or PM For example: 8:25 in the evening (entered as 20:25) is displayed as 8:25 PM
	NOTE: All times are stored and used in INTUITY CONVERSANT system using the hhmmss format. The class format selects only how the time is displayed and printed.	

The values possible in the **Length** column depend on what **Class** is selected. Table 60 describes the limits and default values available for each class.

Table 60. Table Field Lengths

Class Type	Field Length Limits/Comments
Character	Maximum of 255 characters Default: 24 **NOTE:** Voice@Work allows you to put a value greater than 255 in this field. If you do, however, the value reverts to the default (24) without warning when you exit the Table Schema Editor.
Currency	No limit, but values over 15 digits are displayed in scientific notation Default: 15
Date	Cannot be changed by the user Default: 8
Number	Maximum number of digits possible: 10 Maximum and minimum values: 2,147,483,647 and -2,147,483,648 Default: 10
Real	No limit, but values with more than 19 digits are displayed in scientific notation Default: 10
Time	Cannot be changed by the user Default: 6

Defining Fields Using the Table Schema Editor

The following sections describe procedures for defining and editing table schemas using the Table Schema Editor.

Adding a Field to a Table

To add a field to a table's schema, do the following:

1. Right-click in the **Fields** area of the Table Schema window.
2. Select **Add Field** from the popup menu.

   Voice@Work adds the new field *after* the last field already defined in the table.

Inserting a Field in a Table

To insert a field into a table's schema, do the following:

1. Right-click in the **Fields** area of the Table Schema window.
2. Select **Insert Field** from the popup menu.

   Voice@Work adds the new field *before* the currently selected field.

Deleting a Field from a Table

To delete a field from a table's schema, do the following:

1. Right-click in the **Fields** area of the Table Schema window.
2. Select **Delete Field** from the popup menu.

   Voice@Work displays the following dialog box (Figure 94):

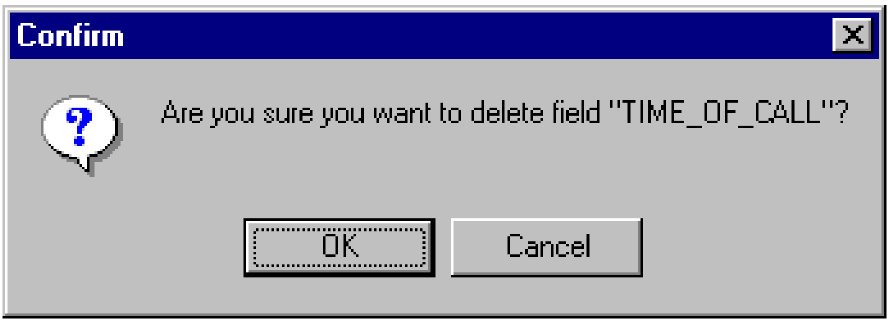


Figure 94. Delete Field Confirmation Box

3. Click **OK**.

Moving a Field within a Table

To move a field within a table's schema, do the following:

1. Right-click in the **Fields** area of the Table Schema window.
2. Do one of the following:
   - To move the field up (left) in the table, select **Move Field Up**.
   - To move the field down (right) in the table, select **Move Field Down**.

Naming or renaming a field

To name or rename a field in a table's schema, do the following:

1. Click in the **Name** column on the line of the field you want to name.
2. Type the name you want to assign to the field.

Defining a Field as a Key Field

To define a field as a "key" field in a table's schema, do the following:

1. Click in the Key column on the line of the field you want to define.
2. Click the pulldown menu pointer.
3. Select Yes from the pulldown menu.

Selecting a Class Type for a Field

To select a class type for a table field, do the following:

1. Click in the Class column on the line of the field whose class you want to define.
2. Click the pulldown menu pointer.
3. Select the class type you want from the pulldown menu.

Defining a Format for a Field Class

To define a format for a table field's class, do the following:

1. Click in the Format column on the line of the field whose class format you want to define.
2. Click the pulldown menu pointer.
3. Select the class format you want from the pulldown menu.

Setting the Field Length

To set the maximum length a table field can have, do the following:

1. Click in the Length column on the line of the field whose length you want to set.
2. Type in the length you want for the field.

When assigning a length to a character field, keep the following in mind:

- Make sure the field length as defined in Voice@Work is equal to (strongly recommended) or greater than the corresponding field length in the ORACLE database.

  CAUTION:
  In no case should you allow the field length as defined in Voice@Work to be less than the corresponding field length in the ORACLE database. Doing so can cause a table query and the application to fail.

- If it will be used as part of a host definition, the total size of all fields defined for a single host screen must be equal to or less than 2000 bytes minus 1 byte of overhead for each host field.

  If you determine that the total size of all fields defined could exceed this 2000-byte buffer during a database table query, you should explore the possibility of creating a view of the database in ORACLE. See your ORACLE documentation or your database administrator if you need help with this.

The Table Editor

The Table Editor allows you to manipulate the data in a database table defined for the current application on the PC.

NOTE:
Data manipulation in Voice@Work takes place only on your PC. If you want to preserve your data and transfer it to your INTUITY CONVERSANT system, you must perform the appropriate Code Generation, Application Transfer, and SCI (Service Creation Installation) actions.

From the Table Editor, you can:

- View data records contained in the table using a variety of navigational means.
- Edit existing data records.
- Add new data records to the table.
- Insert new data records into the table.

- Delete records from the table.
- Perform various operations in the Table Editor using the right-click menu.

Opening the Table Editor

To open the Table Editor, do the following:

1. Open the Tables Manager. See “Opening the Tables Manager” above for the procedure.
2. Right-click in the Tables Manager window anywhere *other than* on the name bar, and select **Edit Table Data...**

   Voice@Work displays Tables Editor window (Figure 95).

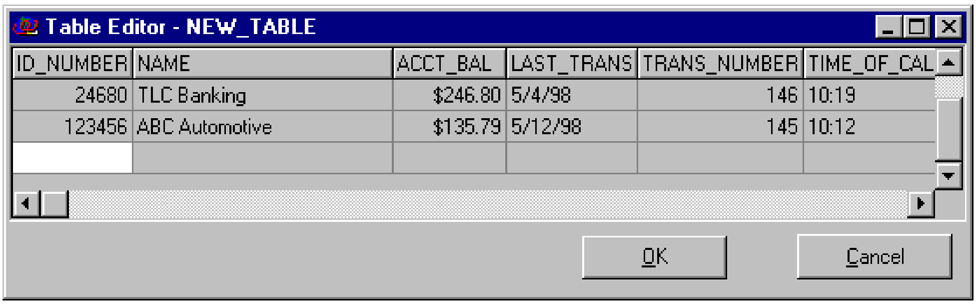


Figure 95. Table Editor

Viewing Records in the Table Editor

To view data records, do the following:

1. Open the Table Editor. See “Opening the Table Editor” above for the procedure.
2. Use the TAB, ENTER, and directional arrow keys to navigate through the table. See “Navigating in the Table Editor” below for details.

Editing Existing Records

To edit existing data records, do the following:

1. Open the Table Editor. See “Opening the Table Editor” above for the procedure.
2. Click in the field or fields you want to edit.

3. Make the desired changes.
4. Click **OK**.

Adding New Records

To add new data records to a database table, do the following:

1. Open the Table Editor. See “Opening the Table Editor” above for the procedure.
2. Choose from among the following actions:
   - Right-click in the Table Editor records area, and select **Add Record** from the popup menu.
   - Press CONTROL A.
   - Press ENTER from the last record field of a line or the down arrow from the last record line.
3. Fill in the record data fields as desired.

Inserting New Records

To insert new data records in a database table, do the following:

1. Open the Table Editor. See “Opening the Table Editor” above for the procedure.
2. Choose from among the following actions:
   - Right-click in the Table Editor records area, and select **Insert Record** from the popup menu.
   - Press CONTROL I.
   - Press TAB.
3. Fill in the record data fields as desired.

Deleting Records

To delete data records from a database table, do the following:

1. Open the Table Editor. See "Opening the Table Editor" above for the procedure.
2. Choose from among the following actions:
   - Right-click in the Table Editor records area, and then select **Delete Record** from the popup menu.
   - Press CONTROL D.

NOTE:
This action cannot be undone. Records are deleted with no warning box and no chance to cancel.

Navigating in the Table Editor

When working in the Table Editor, keep the following in mind:

- The INS key inserts a new record line *before* the currently selected record.
- The ENTER key moves from one field to the next. When you reach the end of a line, pressing ENTER again adds a new record line at the end of the table.
- The directional arrows can be used to move around already defined fields. If you are on the last line, the down arrow adds a new record line.
- Other navigational options are included in the right-click popup menu. This menu is available from anywhere in the Table Editor data records area.

Right-Click Popup Menu for the Table Editor

Table 61 describes the right-click popup menu options.

Table 61. Right-Click Menu Options for the Tables Editor

Menu Item	Keyboard Shortcut	Description/Comments
Delete Record	CONTROL D	Deletes the selected record from the table. **NOTE:** This action cannot be undone. Records are deleted with no warning box and no chance to cancel.
Insert Record	CONTROL I	Inserts a new record *before* the currently selected record.
Add Record	CONTROL A	Adds a new record *after the last record* in the table.
Go to First Record	CONTROL F	Moves the cursor to the first record in the table.
Go to Last Record	CONTROL L	Moves the cursor to the last record in the table.
Go to Next Record	CONTROL N	Moves the cursor to the next record in the table.
Go to Previous Record	CONTROL P	Moves the cursor to the previous record in the table.

The Application Resources Tool

21

Overview

The Application Resources tool makes it easy for you to:

- See what resources are in use in the current application
- Import additional resources into your application either as private copies or as shared resources
- See what resources in your application are being shared-out and shared-in

For more information about sharing application resources, including an explanation of sharing terminology, see Chapter 10, "The Sharing Menu".

Purpose

The purpose of this chapter is to give procedures for:

- Opening the Application Resources Tool
- Viewing Additional Resource Information
- Importing Additional Resources
- Right-Click Menu Options for the Application Resources Tool
- Viewing Shared Resources
- Right-Click Menu Options for the Application Resources Tool

Opening the Application Resources Tool

To open the Application Resources tool, do one of the following:

- From the Tools Menu, select **Application Resources**.
- Click the button on the Toolbar.
- Press ALT T, and then press A.

The Application Resources window opens and displays the types of resources in use by the application (Figure 96).

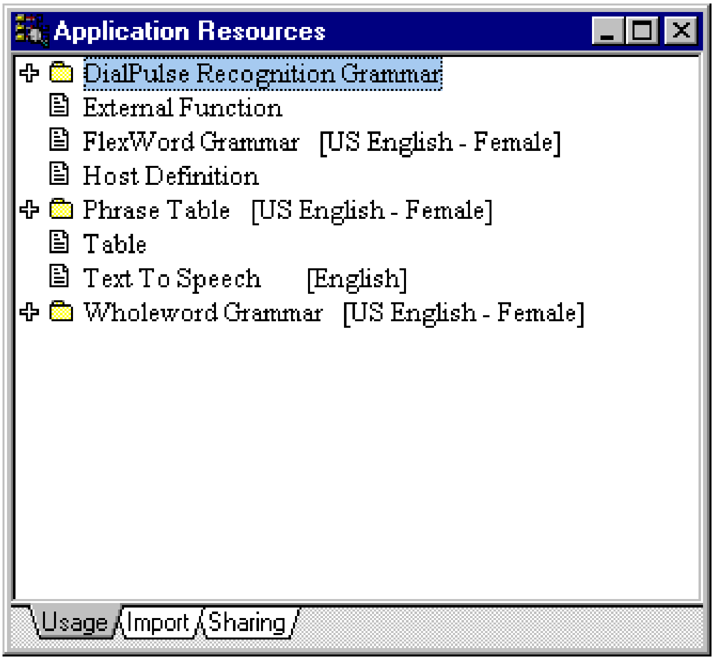
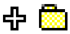
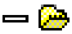


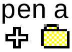
Figure 96. Application Resources Tool – Usage Tab

Table 62 describes the elements of the Application Resources window.

Table 62. Application Resources Tool Window Elements

Element	Description/Comments
	Signifies that there are no resources of that type currently in use.
	Signifies that there are resources of that type currently in use but that the directory for viewing them is closed.
	Signifies that there are resources of that type currently in use and that the directory for viewing them is open.
Usage tab	Allows you to view the resources used in the current application.
Import tab	Allows you to import additional resources into the application.
Sharing tab	Allows you to view what resources are shared-out and shared-in by the current application.

Viewing Additional Resource Information

To open a closed directory and view its contents, double-click a line containing the  icons.

The directory expands to display its contents (see Figure 97).

The  icon before a resource pathname indicates that the resource is in use. The pathname shows where the specific resource file is located.

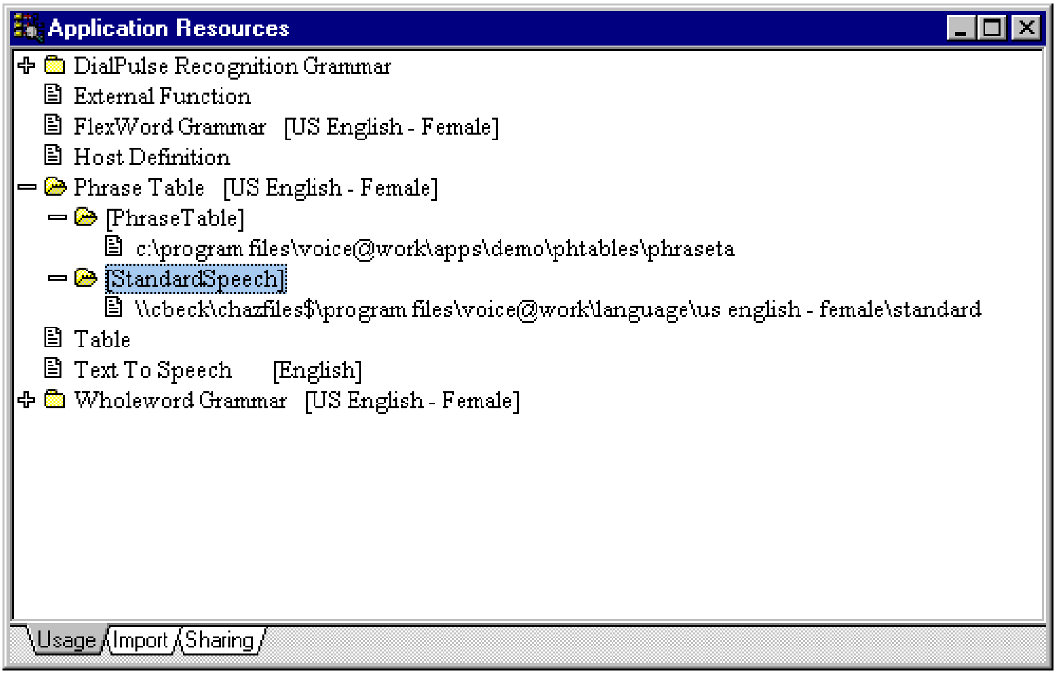


Figure 97. Application Resources Tool Window Showing Expanded Resource Directories

Importing Additional Resources

Voice@Work allows you to import six types of additional resources into your application. Table 63 lists and describes the types of resources you can import.

These resources can be:

- Resources on your PC that you have shared-out from other applications
- Resources that another developer has made available by sharing-out the resource and sharing the network drive it resides on

NOTE:
You must first map the other developer's drive as a network drive on your PC before you can import the resource.

Voice@Work also offers two additional types of resources which cannot be imported. Both of these are included as part of the standard Voice@Work package. They are available for use by all applications and do not need to be imported to be used. These additional types of resources are also described in Table 63.

Table 63. Application Resource Types

Type of Resource	Can Be Imported?	Description/Comments
Custom WholeWord Grammars	Yes	Custom WholeWord grammars are used when WholeWord functionality is desired that exceeds the capabilities of standard WholeWord grammars (see Appendix I, ‘‘Custom WholeWord Grammars’’). When imported, they are treated like any other WholeWord grammar.
External Functions	Yes	External Functions are used when you need Voice@Work to do something that no currently existing node does.
FlexWord lists	Yes	FlexWord speech recognition recognizes words by matching phonemes, or parts of words.
Host Definitions	Yes	Host Definitions are the specifications for the interface between the target system and a 3270 host remote to that machine.
Phrase Tables	Yes	Phrases are bits of spoken or written speech that instruct customers or prompt them for information. They are contained in Phrase Tables.
Tables (Database)	Yes	Tables can be imported into your application from outside or created and populated from within Voice@Work.
Dial Pulse Recognition Grammars	No	Dial Pulse Recognition (DPR) deals with recognizing dial pulse signals received from phones where touchtone service is not available. Obtain descriptions of these grammars by printing a report of the **DialPulse Grammars** from the File menu **Print...** option.

Table 63. Application Resource Types — ***Continued***

Type of Resource	Can Be Imported?	Description/Comments
WholeWord Grammars	No	WholeWord speech recognition recognizes entire words, not phonemes or part of words. Obtain descriptions of these grammars by printing a report of the **WholeWord Grammars** from the File menu **Print...** option.

To import additional resources into your application, do the following:

1. Open the Application Resources tool. See "Opening the Application Resources Tool" above for the procedure.

   Voice@Work displays the types of resources in use by the application (Figure 96 and Figure 97).

2. Click the **Import** tab.

   The Application Resources window displays the Import view (Figure 98).

3. Select which type of resource you want to import from the pulldown menu at the top of the window (Figure 98).

4. Select whether you want to import the resource as a private copy or as a shared resource (Table 64).

5. Use the **Drive** pulldown menu to select the drive you want to import the resource from. To use the default drive (that is, the drive where Voice@Work is installed), leave this field blank.

6. Select the resource you want to import.

7. Click the **Import** button.

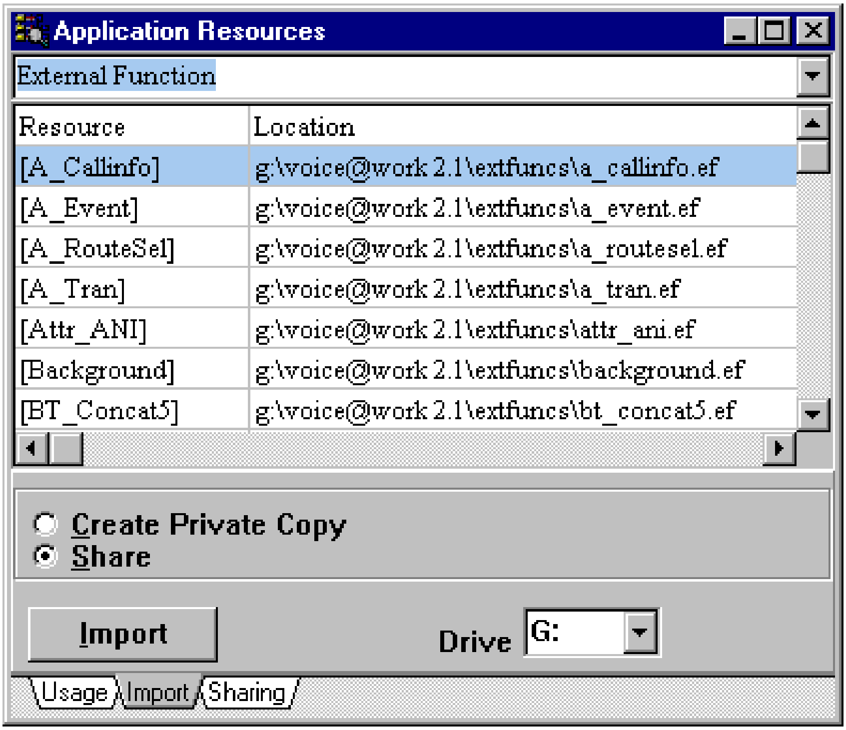


Figure 98. Application Resources Tool - Import Tab

Table 64 describes the elements found on the **Import** tab of the Application Resources tool.

Table 64. Application Resources - Import Tab Elements

Element	Description/Comments
Resource type pulldown menu	Use the pulldown menu at the top of the window to select which type of resource you want to import. Table 63 lists the types of resources you can import.
Resource column	Displays the resources available on a particular drive for the selected type. Select the resource you want from this list.
Location column	Shows the drive and full pathname of resources available for importing.

Continued on next page

Table 64. Application Resources - Import Tab Elements — *Continued*

Element	Description/Comments
Create Private Copy and **Share** radio buttons	Select from the two options in this box how you want to import the resource into your application. To import: ■ A private copy that you can modify and save with a different filename, select **Create Private Copy** ■ A resource that is being shared by another developer, select **Share** **NOTE:** You cannot select both options at the same time.
Import button	Click this button to import the resource using the parameters you have selected.
Drive select pulldown menu	Use this pulldown menu to select the drive you want to search for available resources. Assuming you have mapped the appropriate network drives, you can see resources shared by other developers (see Chapter 10, ''The Sharing Menu'').

Viewing Shared Resources

You can use the Application Resources tool to see at a glance what resources in your application are shared-out for use by others and what resources you have shared-in from outside your application.

To view shared resources, do the following:

1. Open the Application Resources tool (see "Opening the Application Resources Tool" in this chapter).

   Voice@Work displays the types of resources in use by the application.

2. Click the **Sharing** tab.

   The Application Resources window displays the **Sharing** view (Figure 99).

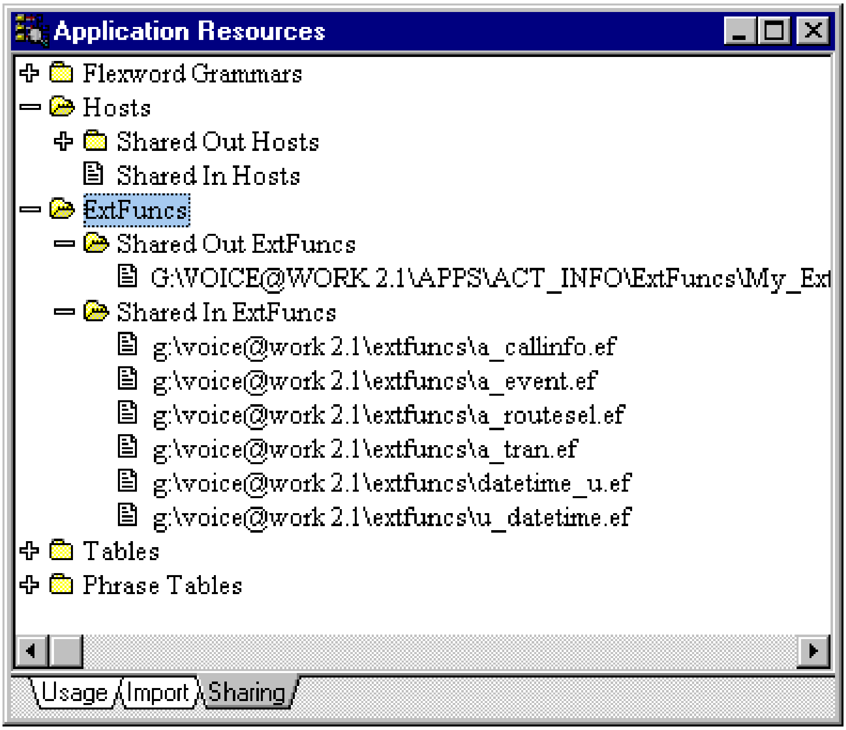
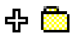
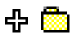


Figure 99. Application Resources Tool – Sharing Tab

Table 65 describes the elements found on the **Sharing** tab of the Application Resources tool.

Table 65. Application Resource Tool Elements – Sharing Tab

Icon	Location	Description
	In front of a resource type	Signifies nothing – Indicates only the types of resources that *might* be shared-in or -out.
	In front of a **Shared-out** or **Shared-in** directory	Signifies that there are no shared resources of that type (see **Shared-in Hosts** in Figure 99).
	In front of a **Shared-out** or **Shared-in** directory	Signifies that there *are* resources of that type shared-out or shared-in but that the directory for viewing them is closed (see **Shared-out Hosts** in Figure 99).
	In front of a **Shared-out** or **Shared-in** directory	Signifies that there are resources of that type shared-out or shared-in and that the directory for viewing them is open (see **Shared-out ExtFuncs** or **Shared-in ExtFuncs** in Figure 99).
	In front of a **Shared-in** resource (under a directory)	Shows the drive and pathname for the resource.

Right-Click Menu Options for the Application Resources Tool

Table 66 describes the Application Resources tool right-click menu options.

NOTE:
The right-click popup menu is accessible only from the **Usage** or **Sharing** tabs. To access this menu, select any resource subdirectory, and then right-click.

Table 66. Application Resources Right-Click Menu Options

Menu Item	Description/Comments
Delete	(**Usage** tab only) Allows you to delete a resource that has been imported. This option does not allow you to delete: ■ A system resource (such as Standard Speech) ■ A resource that is in use by the current application
Change to Private Copy	(**Usage** tab only) Allows you to create a private copy of a resource that you have previously designated as shared-in.
Expand	Provides an expanded view of the selected directory.
Collapse	Provides a collapsed view of the selected directory.
Expand All	Expands all directories.
Collapse All	Collapses all directories.

The Call Flow Inventory Tool

22

Overview

A call flow in Voice@Work is defined as a series of nodes or objects linked together.

The Call Flow Inventory tool allows you to see:

- What call flows exist for your application.
- Where each call flow is used within the application
- What other call flows, if any, are used within each call flow.

NOTE:
Host call flows are different, because they use only a certain set of nodes, and they are not linked to voice call flows. Host flows are dedicated to establishing and maintaining required communication with remote hosts.

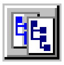
Purpose

The purpose of this chapter is to describe the Call Flow Inventory tool and how to use it. Topics include:

- Opening the Call Flow Inventory Tool
- Displaying the Uses Tab View
- Displaying the Used By View
- Dragging a Call Flow to Create a Copy
- Right-Click Menu Options for the Call Flow Inventory Tool

Opening the Call Flow Inventory Tool

To open the Call Flow Inventory tool, do one of the following:

- From the **Tools** menu, select **Call Flow Inventory**.
- Click the  button on the Toolbar.
- Press ALT T, then press I.

Voice@Work displays the Call Flows window (Figure 100).

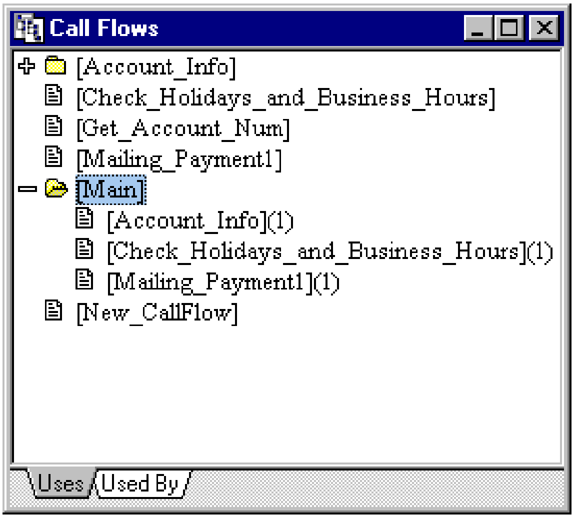


Figure 100. Call Flows Window

NOTE:
To locate any call flow displayed in the Call Flow Inventory window, double-click on the name of the call flow. Voice@Work locates the call flow, highlights it, and makes it the active window.

Displaying the Uses Tab View

To display the Uses tab view, click the **Uses** tab in the Call Flows window.

Table 67 describes the elements of the Call Flows window in the **Uses** tab view (Figure 100).

Table 67. Call Flows Window Elements—Uses Tab View

Element	Description
	Signifies that the designated call flow is defined but contains no other call flows.
	Signifies that the designated call flow contains other call flows but the directory for viewing them is closed.
	Signifies that the designated call flow contains other call flows and that the directory for viewing them is open so you can see them. In this example, the [Main] call flow, marked with the icon, uses the following subordinate call flows: ■ [Account_Info] ■ [Check_Holidays_and_Business_Hours] ■ [Mailing_Payment1]
Uses tab	Allows you to view the call flows defined for the current application and to see what other call flows they reference (use).
Used by tab	Allows you to view call flows and find out which other call flows reference (use) them.

Expanding a Call Flow

To expand a call flow containing other call flows, double-click a call flow containing the icons.

The call flow expands to display its contents, that is, the subordinate call flows it uses. At the same time, the design pad displays the parent call flow.

NOTE:
To locate any other subordinate call flows in the parent call flow, double-click that subordinate call flow.

Collapsing a Call Flow

To collapse a call flow containing other call flows, double-click a call flow containing the icons.

The call flow collapses to hide its contents, that is, the subordinate call flows it uses. At the same time, the design pad displays the selected call flow.

Displaying the Used By View

To display the Used By view, click the **Used By** tab on the Call Flows window.

The Call Flows window displays the call flows defined for the current application (Figure 101).

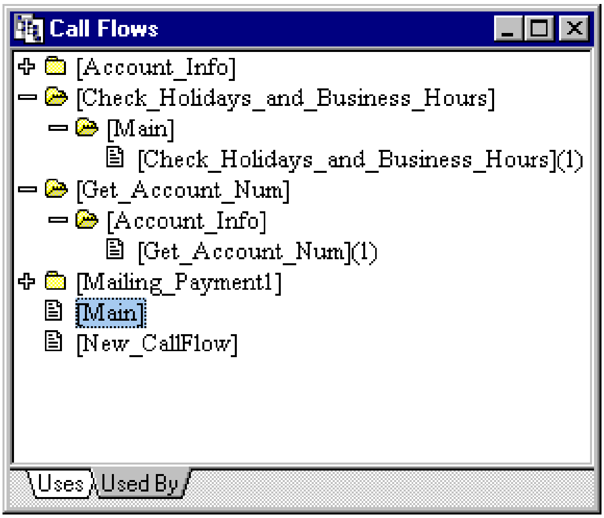


Figure 101. Call Flow Inventory — Used By Tab View

Table 68 describes the various elements of the Call Flows window in the **Used By** tab view (Figure 101).

Table 68. Call Flows Window Elements—Used By Tab View

Element	Description/Comments
Display pane	Displays the call flows currently defined for the application.
	At the primary level, signifies that the call flow is defined but is not used by any other call flow. At a subordinate level, signifies that the call flow uses the call flow in the level above.
	Signifies that the call flow is used by other call flows, but that the call flows using it are hidden (collapsed view).
	Signifies that the call flow is used by other call flows, which are displayed at a secondary level underneath (expanded view) The number in parentheses indicates how many times the parent call flow had used that subordinate call flow at the time the subordinate call flow was created. **NOTE:** If one or more references to the subordinate call flow are later deleted, Voice@Work does *not* renumber the remaining references to it.

Expanding a Call Flow Used by Other Call Flows

To expand a call flow used by other call flows, double-click a call flow containing the icons.

The call flow expands to display its contents, that is, the parent call flows that use it. At the same time, the design pad displays the used call flow, that is, the one you double-clicked on.

NOTE:
To locate a call flow in the parent call flow, double-click the call flow marked with the icon.

Collapsing a Call Flow Used by Other Call Flows

To collapse a call flow used by other call flows, double-click a call flow containing the icons.

The call flow collapses to hide its contents, that is, the parent call flows that use it. At the same time, the design pad displays the selected call flow.

Dragging a Call Flow to Create a Copy

You can use the Call Flow Inventory tool to create a copy of a call flow by dragging the call flow onto the design pad.

To create a copy of a call flow in this manner, do the following:

1. Click the name of the call flow you want to copy and drag it out of the Call Flow Inventory tool window.
2. Place and drop the call flow where you want it on the design pad, either on an open branch or above or below an occupied branch.

   Voice@Work places the call flow much as it would any other node object, with the following exceptions:

   - When you drag the call flow over the design pad, the cursor changes to the following:

     When the cursor has this appearance, you can drop it.
   - When you drag the call flow over an occupied branch or other invalid drop area, the cursor changes to the following:

     When the cursor has this appearance, you cannot drop it. If you try to drop the call flow here, Voice@Work aborts the process without copying the call flow.

Right-Click Menu Options for the Call Flow Inventory Tool

NOTE:
The right-click popup menu appears only when you right-click the name of a call flow.

Table 69 describes the right-click menu options.

Table 69. Right-Click Menu Options of Call Flow Inventory

Option	Description/Comments
Edit	Displays the selected call flow in the design pad, where you can edit it.
Delete	Allows you to delete the selected call flow from the application. **NOTE:** You can only delete a call flow that does not use and is not used by another call flow. The **[Main]** call flow cannot be deleted. You cannot delete a host flow except by deleting its host definition.
Expand	Expands the selected call flow and displays its contents.
Collapse	Collapses the selected call flow and hides its contents.
Expand All	Expands all call flows and displays their contents.
Collapse All	Collapses all call flows and hides their contents.

The Code Generation Tool

23

Overview

The Code Generation tool generates the code files necessary to install and compile the application on the target voice response system.

Purpose

The purpose of this chapter is to describe the Code Generation tool and how to use it. Topics include:

- Opening the Code Generation Tool
- Setting Code Generation Options
- Generating the Target Files
- Code Generation Application Transfer File Types
- The Code Generation Process
- Menus in the Code Generation Window
- Database Table Provisioning

Opening the Code Generation Tool

To open the Code Generation tool, do one of the following:

- From the **Tools** menu, select **Code Generation/Application Transfer**.
- Click the  button on the Toolbar.
- Press ALT T, then press C.

Voice@Work displays the Code Generation/Application Transfer window (Figure 102). The appearance of the window depends on the following conditions:

- If errors exist in files already generated for the application, the file types appear in red text in the **Target Files to Generate** box.
- The checkboxes in the **Target Files to Generate** box are checked by default whenever:
  — Target files of that type have not already been generated.
  — The application source files for that type have been changed since the last time code was generated.
- File types are inactive whenever no items for that type have been defined within the application. For example, if no tables are defined in the application, the **Database** files option is inactive. If no hosts are defined, the **Host definition** file option is inactive; and so on.

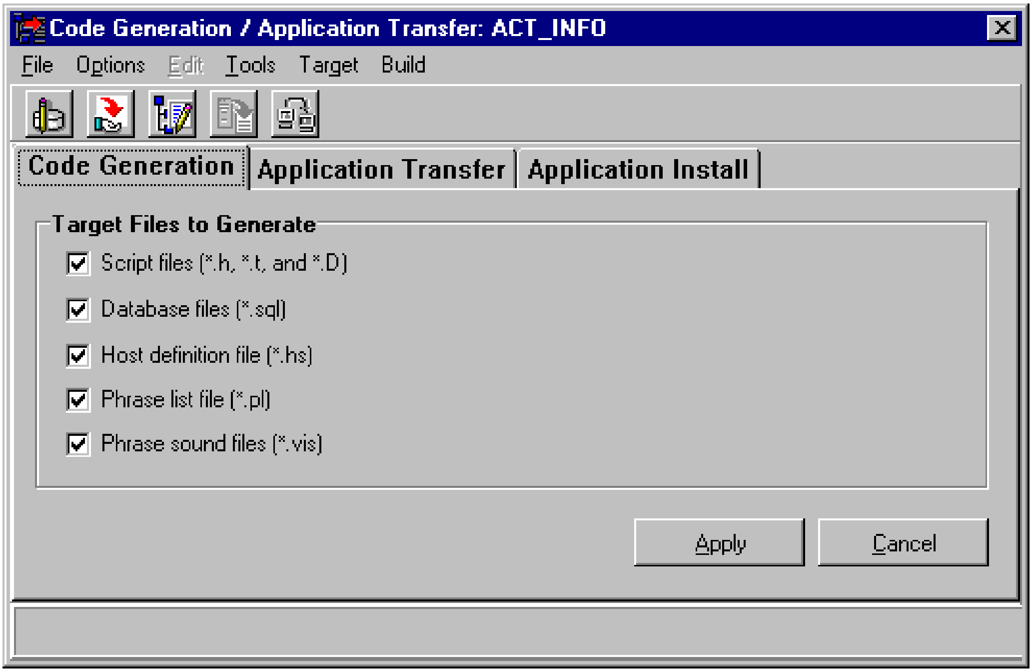


Figure 102. Code Generation/Application Transfer Window

Setting Code Generation Options

The Code Generation Options dialog box allows you to set options for the Code Generation process.

To set the Code Generation Options, do the following:

1. Open the Code Generation/Application Transfer tool window (Figure 102).
2. Click the **Code Generation** tab.
3. From the **Options** menu, select **Code Generation**.

   Voice@Work displays the Code Generation Options dialog box (Figure 103).

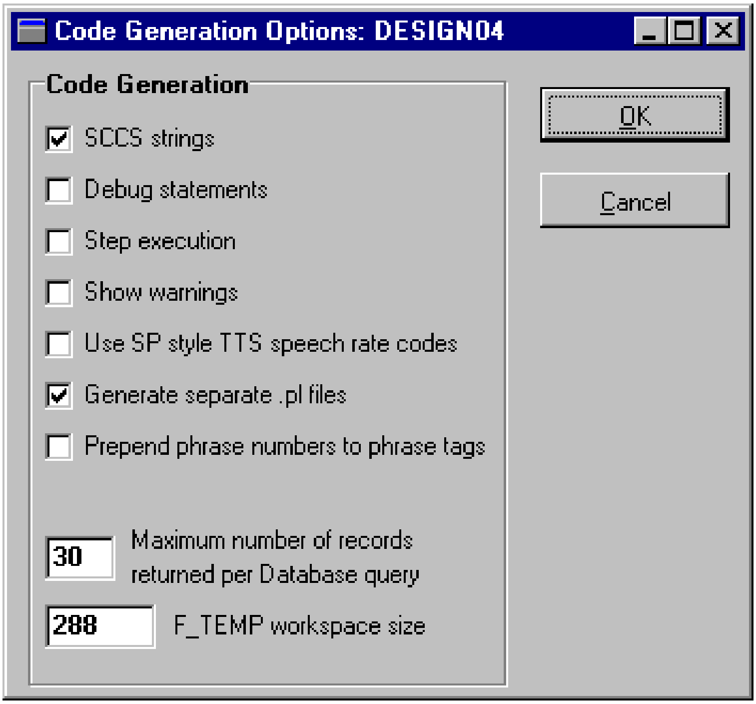


Figure 103. Code Generation Options Dialog Box

4. Check the appropriate boxes to select the options you want. Table 70 describes the options available.
5. Click **OK.**

Table 70. Code Generation Options

Option	Description
SCCS Strings	Causes Source Code Control System (SCCS) identification strings (for example, %W%) to be written to all target files that support a comment convention. Default = unchecked.
Debug Statements	Adds comment lines that trace function calls in the Code Generation source code to the *.t and *.h files. Useful for debugging the generated application code on the target system. Default = unchecked.

Continued on next page

Table 70. Code Generation Options — *Continued*

Option	Description
Step Execution	Causes the Code Generation process to pause after the code is generated to each call flow node. Useful only for debugging the Code Generation software. Default = unchecked.
Show Warnings	Generates warnings for possible application errors or missing files, such as unrecorded speech files. Default = unchecked.
Use SP style TTS speech rate codes	Causes the Code Generation process to use escape codes for the SP circuit card, rather than the default SSP circuit card escape codes. Default = unchecked.
Generate separate .pl files	Generates a separate *.pl file for each phrase table defined for the application. Default = unchecked. When checked, this option: ■ Assigns the filename ***EBS_language_name*.pl** to the phrase table containing the Enhanced Basic Speech (EBS) for the standard speech phrase table (called Standard Speech in Voice@Work). For example, if the EBS language is US English – Female, Voice@Work generates a phrase table file called **US_English.pl**. ■ Uses ***application_name*.pl** for the default custom phrase table (called Phrase Table in Voice@Work), unless you have renamed the default custom phrase table. For example, if you have an application named **new_app**, and you generate the phrase table file without renaming the file, Voice@Work generates a file called **new_app.pl**. ■ Gives any other phrase tables the name ***table*.pl**, where ***table*** represents the phrase table name you assigned to it when you created it. For example, if you create a phrase table and name it **my_table**, Voice@Work generates a phrase table file called **my_table.pl**. When this option is unchecked, Voice@Work generates a single *.pl file called ***application_name*.pl**. Voice@Work allows you to create a maximum of 8 phrase table (*.pl) files.

Continued on next page

Table 70. Code Generation Options — *Continued*

Option	Description
Prepend phrase numbers to phrase tags	Attaches the phrase numbers to the beginning of the phrase text when creating phrase tags. As Voice@Work does not allow you to create identical phrases, this allows you to create more than one phrase with the same text. Default = unchecked
Maximum number of records returned per database query	Allows you to define how many records can be returned for each database query operation. Default = 30. **NOTE:** This number represents the maximum limit of records for *all tables*, so you must select a value for this field carefully. Set the number high enough to allow for the greatest number of records you want to return for any one table. At the same time, consider that the higher this number, the greater the amount of memory you are committing.
F_TEMP workspace size	Allows you to set the temporary workspace required by an application. Default = 288, Range = 288 - 102400 **CAUTION:** *Because this workspace is allocated on a per-instance basis, you should exercise caution when increasing this workspace size. If you have a large enough workspace and a great enough number of instances, you can easily cause memory conflicts.*

Generating the Target Files

To generate the source code files for the target system, do the following:

1. Open the Code Generation/Application Transfer tool window (Figure 102).
2. If you have changed the name of your application since the last build, select **Remove** from the **Build** menu.

   This option removes all files previously generated for the application. Lucent Technologies recommends that you select this option whenever you change your application's name to avoid confusion later on.
3. (Optional) Select the Code Generation Options you want (see "Setting Code Generation Options" above).

4. (Optional) Select which tables you want to include in your build (see "Database Table Provisioning" below).

5. Select the files you want to generate.

   - To generate all files possible, select **Build All** from the **Build** menu.
   - To generate specific types of files, check the appropriate boxes on the list in the **Target Files to Generate** box. For a description of the types of files that can be generated using the Code Generation tool, see "Code Generation Application Transfer File Types" below.

6. Do one of the following:

   - Click the **Apply** button.
   - Select **Build All** from the **Build** menu.

   If there are any files existing from a previous build, Voice@Work asks whether you want to overwrite them.

   Voice@Work generates the files required for Application Transfer.

Code Generation Application Transfer File Types

Table 71 describes the types of files generated during the Code Generation process that are available for transfer during the Application Transfer process.

NOTE:
If no resource exists in the application for a particular file type (such as Host Definitions or Tables), that option is inactive in the **Code Generation** tab window, and no files are generated for that file type.

Table 71. Code Generation / Application Transfer File Types

File Types	Names/Extensions	Description
Script Files	*.t	Contains the actual TAS code (script) for the application.
	*.h	Contains the header information for the *.t file.
	*.D	Contains Call Data Events information.

Continued on next page

Table 71. Code Generation / Application Transfer File Types — *Continued*

File Types	Names/Extensions	Description
Database Files	**NOTE:** Use the **Database Table Provisioning** window to select which tables to include in the build. By default, all tables defined for the application are included.	
	*.sql	Contains the table definitions and data for tables being exported, formatted for use with ORACLE.
Host Definition Files	*.hs	Contains information about host definitions. One file is generated for each host defined for the application.
Phrase List Files	*.pl	Contains a list of all phrases defined for the application. Generated even if the application does not use recorded speech.
Phrase Sound Files	*.vis	Contains *.wav audio files converted into INTUITY CONVERSANT-specific ADPCM format.

Continued on next page

Table 71. Code Generation / Application Transfer File Types — *Continued*

File Types	Names/Extensions	Description
Automatic (Non-selectable) Files	*.ccv	Provides details on how much of the code was **Simulation** tested, the number of call flows in the application, and the number of different types of nodes used.
	*.fea	Contains a list of features used by the application.
	*.toc	Provides table of contents information for all the source files and target files in the application.
	*.E, *.Eaa	Generated as companion files to the *.D file (see above).
	*.err	Contains information about errors encountered during the Code Generation process.
	*.flx	Converts FlexWord lists into a format the target system can use.
	params	Contains information about application parameters.
	Resource	Lists all the database tables defined for the application. If the table is defined as a remote host table, there is an additional field to identify the remote host machine name.
	LangName	Contains the name of the Enhanced Basic Speech (EBS) language used by the application.
	symbols	Contains information about application symbols – not used by the target system.

The Code Generation Process

During the Code Generation process, the following actions occur:

- File type entries in the **Target Files to Generate** box appear in green text as those types of files are being generated.
- Each file type checkbox is unchecked as the Code Generation process is completed on the selected types of files.
- Any file type that generates an error is flagged with red text.

Figure 104 is a snapshot of a Code Generation process in progress. Notice that the Phrase list files are currently being generated. Notice also that there was an error generated somewhere in one or more of the Script files.

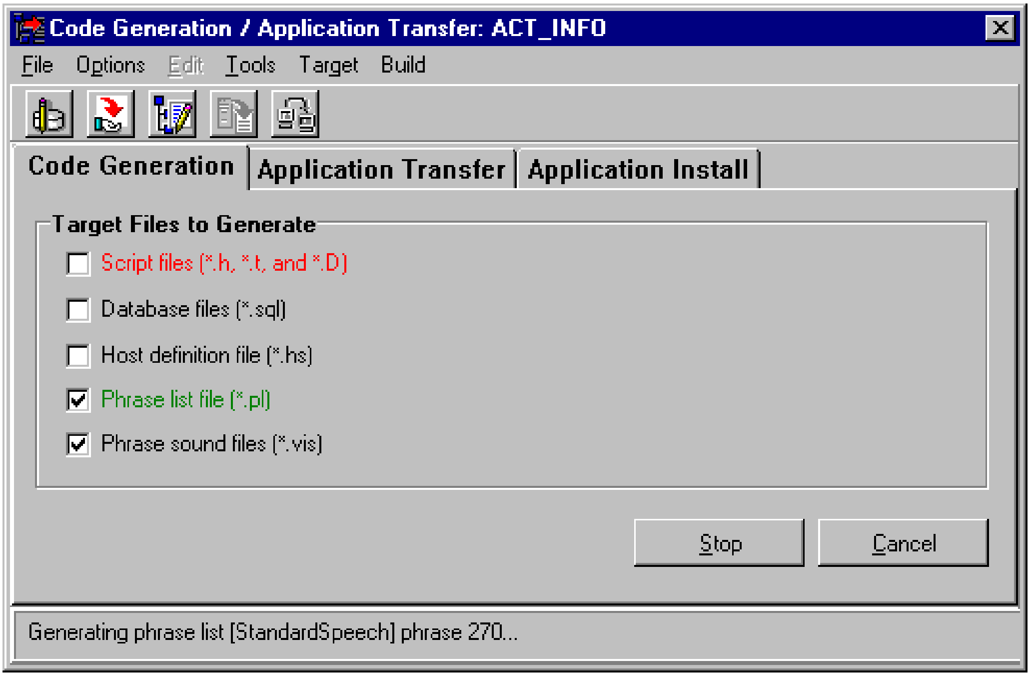


Figure 104. Code Generation/Application Transfer Window During the Code Generation Process

If Voice@Work encounters errors during the Code Generation process, Voice@Work displays the error message shown in Figure 105:

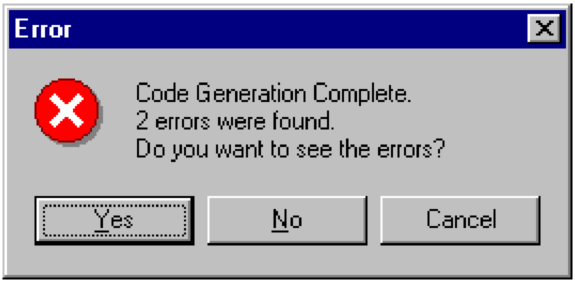


Figure 105. Code Generation Error Message

- To continue without viewing the error log, click the **No** button.
- To view the error log, click the **Yes** button (Figure 106).

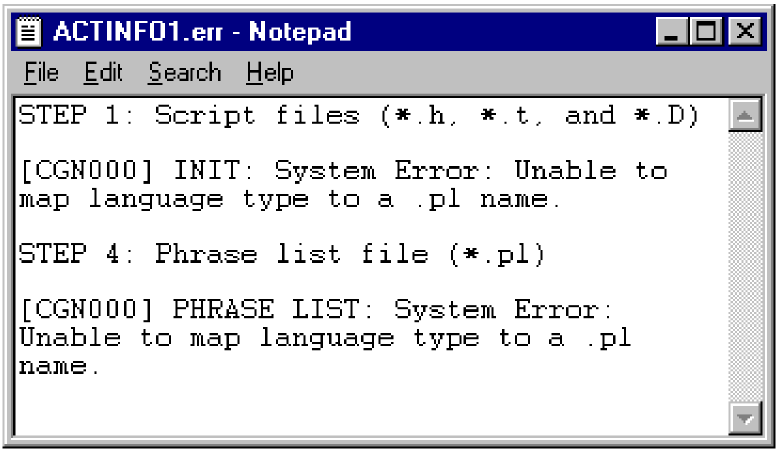


Figure 106. Sample Error Log File

Menus in the Code Generation Window

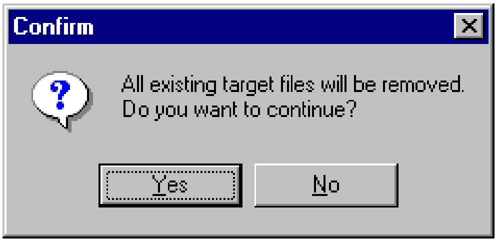
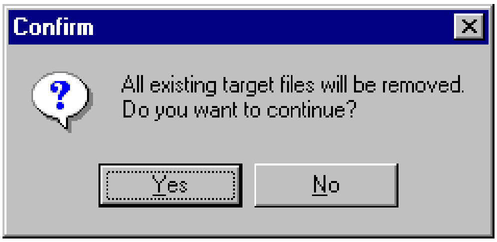
Figure 104 describes the menus in the Code Generation window and their options.

Table 72. Code Generation Window Menus

Menu	Option	Description
File	Open	Not available from the Code Generation tab view.
	Exit	Closes the Code Generation/Application Transfer dialog box immediately.
Options	Code Generation	Opens the Code Generation Options window (see "Setting Code Generation Options" above).
	Application Transfer	Not available from the Code Generation tab view.
Edit		Not available from the Code Generation tab view.
Tools	External Function Source Editor	Opens the External Function Source Editor window, where you can create or edit custom external function TAS files (see "The External Function Source Editor" in Chapter 16, "Working with External Functions"). Inactive if no custom external functions have been defined for the application.
	Database Table Provisioning	Opens the Database Table Provisioning window, where you can select which tables to include in the build (see "Database Table Provisioning").
	Refresh From Shared Resources	Refreshes any shared resources used in the application. Inactive if no application resources are shared (see "Refreshing Shared-In Resources" in Chapter 10, "The Sharing Menu").
Target	[Generated files list]	Lists all files that have been generated for the application. **NOTE:** The Target Menu is inactive and is grayed out on the menu bar if no files have been generated for the application. To view the contents of a particular file from the list, click that file's name.

Continued on next page

Table 72. Code Generation Window Menus — ***Continued***

Menu	Option	Description
Build	Build All	Includes all possible files in the build and overwrites any files that have been generated previously. If previously generated files exist, Voice@Work displays the following message:  ■ To continue with the build, click **Yes**. ■ To cancel the operation, click **No**.
	Remove	Removes all previously generated files from the application. If none exist, this menu item is inactive.  **NOTE:** Lucent Technologies recommends that you select this option whenever you change your application's name to avoid confusion later on. ■ To remove all previously generated files, click **Yes**. ■ To cancel the operation, click **No**.
	Errors	Displays the error log, if one exists. If no error log exists, this menu item is inactive.

Database Table Provisioning

Use the Database Table Provisioning window to select which tables you want to include for Code Generation and export during Application Transfer.

To open the Database Table Provisioning window, do one of the following:

- From the **Tools** menu of the Code Generation window, select **Database Table Provisioning**.
- Click the  button on the Code Generation toolbar.

Voice@Work displays the Database Table Provisioning window (Figure 107).

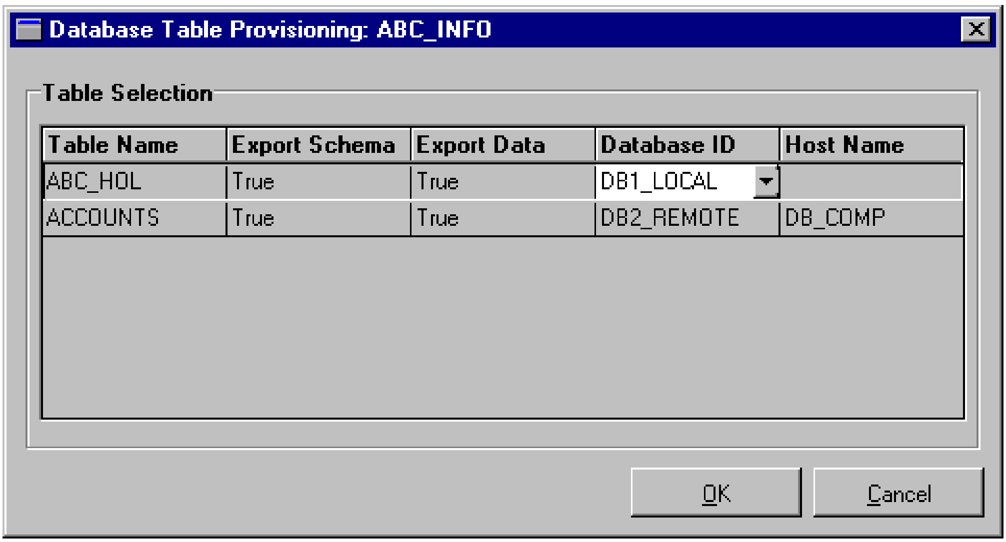


Figure 107. Database Table Provisioning Window

Table 73 describes the fields on the Database Table Provisioning window.

Table 73. Database Table Provisioning Window Fields

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>Table Name</td><td>Displays the names of all tables defined for the current application.</td></tr>
<tr><td>Export Schema</td><td>Used to select which table schemas will be exported during the Code Generation and Application Transfer processes.
■ For table schemas you want to include for export, select True in the Export Schema field.
■ For table schemas you do not want to include for export, select False in the Export Schema field.</td></tr>
<tr><td>Export Data</td><td>Used to select which table data will be exported during the Code Generation and Application Transfer processes.
■ For tables whose data you want to include for export, select True in the Export Data field.
■ For tables whose data you do not want to include for export, select False in the Export Data field.</td></tr>
<tr><td>Database ID</td><td>Displays the names to be assigned to the tables when they are transferred to the target voice response system.
You must assign a Database ID to each table defined for your application, according to the following guidelines. If the table is to reside on:
■ The target system, select a DB#_LOCAL format.
■ A remote host (relative to the target system), select a DB#_ REMOTE format.
NOTE:
If the table is to reside on a remote host, you must also enter a name in the Host Name column.</td></tr>
<tr><td>Host Name</td><td>Displays the name of the remote host where the remote database table resides.
NOTE:
Host names can have a maximum of 11 characters and must start with an alphabetic character.</td></tr>
</table>

NOTE:
By default, all table schemas and data are selected for export. If there are table schemas or data you do not want to export, you must select the **False** value for them.

For example, if you plan to populate the table with data from another source, you would export the table schema but not the data. But if the schema already exists on the target machine, and you want simply to populate it with new data, you would export the table data but not the schema.

To select tables and data for export, do the following:

1. Open the Database Table Provisioning window.

   Each table name appears in the **Table Name** column.

2. Select table schemas you want to include for export (see Table 73).
3. Select table data you want to include for export (see Table 73).
4. Assign a **Database ID** from the pulldown menu for each table in your build (see Table 73).
5. If the table is to reside on a remote host, enter the name of the remote host in the **Host Name** field.
6. Click **OK**.

The Application Transfer Tool

24

Overview

The Application Transfer tool transfers the application files to the target voice response system on which the application is to be installed and run. This tool allows you to transfer these generated code files using either a network connection or floppy disks. You can select all or only some of the generated code files for application transfer.

Purpose

The purpose of this chapter is to give information and procedures about:

- Opening the Application Transfer Window
- Setting Application Transfer Options
- Transferring an Application
- Transferring Files from Floppy Disks to the Target System
- Code Generation/Application Transfer File Types
- Transferring Application Source Files
- Application Transfer Window Menus

Opening the Application Transfer Window

To open the Application Transfer window, do the following:

- From within Voice@Work, do one of the following:
  - — Click the  button.
  - — Click the Tools Menu, and then select **Code Generation/Application Transfer**.
  - — Press ALT T, and then press C.
- From outside Voice@Work (the standalone version), do the following:
  1. Click on the **Start** button in Windows.
  2. Locate the shortcut for the **Voice@Work Application Transfer** standalone application. This file should reside in the directory in which you have installed your Voice@Work application files.
  3. Click the **Voice@Work Application Transfer** option.

     NOTE:
     The standalone version of the Application Transfer tool does not have or display any of the Code Generation tool options.

Setting Application Transfer Options

The Application Transfer Options dialog box allows you to select and set options for Application Transfer. You can choose to transfer your application files between floppy disks and the target system, or over a network directly to the target system. You can also elect to have Voice@Work notify you before overwriting files at the transfer destination and when the transfer is complete. Finally, you can elect whether to use your settings as the default settings for future applications.

To set the Application Transfer Options, do the following:

1. Open the **Code Generation / Application Transfer** tool window.
2. Click the **Application Transfer** tab.
3. From the **Options** menu, select **Application Transfer**.

   Voice@Work displays the Application Transfer Options dialog box (Figure 108).

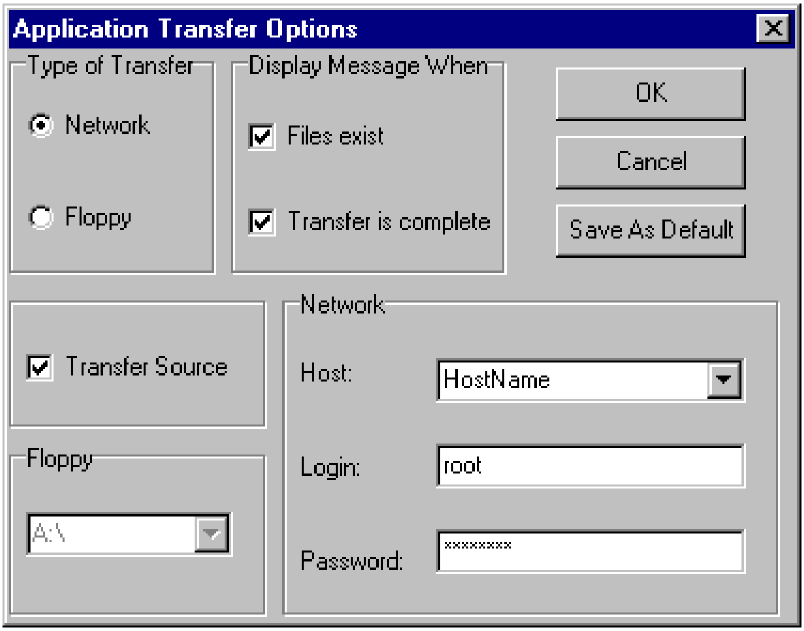


Figure 108. Application Transfer Options

4. In the **Type of Transfer** box, select the method for transferring your application from your PC to the target system:

   - **Network** allows you to transfer the application using a TCP/IP network connection. Selecting this option activates the **Network** information area on the window.

     NOTE:
     It is possible to use the **Network** option to transfer your application using a modem connection. In order to transfer files using a modem connection, you must first set up the necessary PPP connections between your PC and the target system. For information on setting up the necessary PPP connections, see Appendix H, "Transferring Files Using a Modem".

   - **Floppy** allows you to transfer the application files to floppy disks, which can then be used to transfer the files to the target system. Selecting this option activates the **Floppy** information area on the window.

5. If you want Voice@Work to notify you before overwriting any files with matching filenames, click the **Files exist** checkbox.
   - If you click the **Files exist** checkbox, Voice@Work checks the destination system to see if files using the same application name exist. If they do, you get a warning when you try to transfer the application. If you choose to proceed with the transfer, Voice@Work overwrites any files with matching filenames.
   - If you do *not* click the **Files exist** checkbox, Voice@Work overwrites any files at the destination with matching filenames, *without* notifying you that it will do so.
6. If you want Voice@Work to notify you when the transfer process has been completed, click the **Transfer is complete** checkbox.
   - If you click the **Transfer is complete** checkbox, Voice@Work opens an alert box after the transfer is finished stating that the transfer has completed.
   - If you do not click the **Transfer is complete** checkbox, Voice@Work does not notify you when the transfer is complete.
7. If you want Voice@Work to transfer application source files along with the generated code, check the **Transfer Source** checkbox. For more information, see “Transferring Application Source Files” below.
8. If you are transferring:
   - Over a network, enter the following information about the target system in the **Network** information fields:
     a. Enter the name or the IP address of the destination host in the **Host:** field.

        If you have a **Hosts** file defined in your **\Windows** directory (for Windows 95) or **\Winnt\system32\drivers\etc** directory (for Windows NT 4.0), the **Host** pulldown menu displays the names of the hosts listed in that file.

        Otherwise, the pulldown menu is blank, and you must provide the name of the host you want.
     b. Type the user login ID in the **Login:** field.
     c. Type the user password in the **Password:** field.

        For security reasons, only asterisks (*) appear in the **Password:** field as you type in the password.

- To floppy disks, select the drive you want from the **Floppy** pulldown menu.

  NOTE:
  All available removable media drives appear in the pulldown menu, but do not select anything other than a floppy drive (or at least make sure that the target system also supports the type of removable media for the drive you want to use).

9. If you want Voice@Work to transfer application source files along with the generated code, check the **Transfer Source** checkbox.
   - For more information about transferring application source files, see "Transferring Application Source Files" below.
10. If you want to make the options you have selected the default settings for future application transfers, click the **Save As Default** button.

    NOTE:
    Unless you first click the **Save As Default** button, clicking **OK** sets the options only for the current application. The next application reverts to the previous defaults.

11. Click **OK**.

Transferring an Application

Before you can transfer your application to the target system, you must first:

- Generate the application code files using the Code Generation tool (see Chapter 23, "The Code Generation Tool").
- Set the Application Transfer Options (see "Setting Application Transfer Options" above).

When transferring an application with a name matching an application that already exists on the target voice response system, Lucent Technologies recommends that you first remove the old application files. To do this, use the **scrm** command on the target system. See Appendix A, "Using the SCI Tools".

For example, suppose you had an application **abc_auto** on the target system that you later replaced with a new, totally different **abc_auto** application that does not use the same FlexWord lists and database tables. When you transfer the new **abc_auto** application to the target system, only the files with matching filenames are replaced. The old FlexWord lists and database tables remain, however, and can cause conflicts when you use the new version.

To transfer your application to the target voice response system:

1. Open the **Code Generation/Application Transfer** window. See “Opening the Application Transfer Window” above for the procedure.
2. Click the **Application Transfer** tab.

   The window changes to a display similar to the one shown in Figure 109.

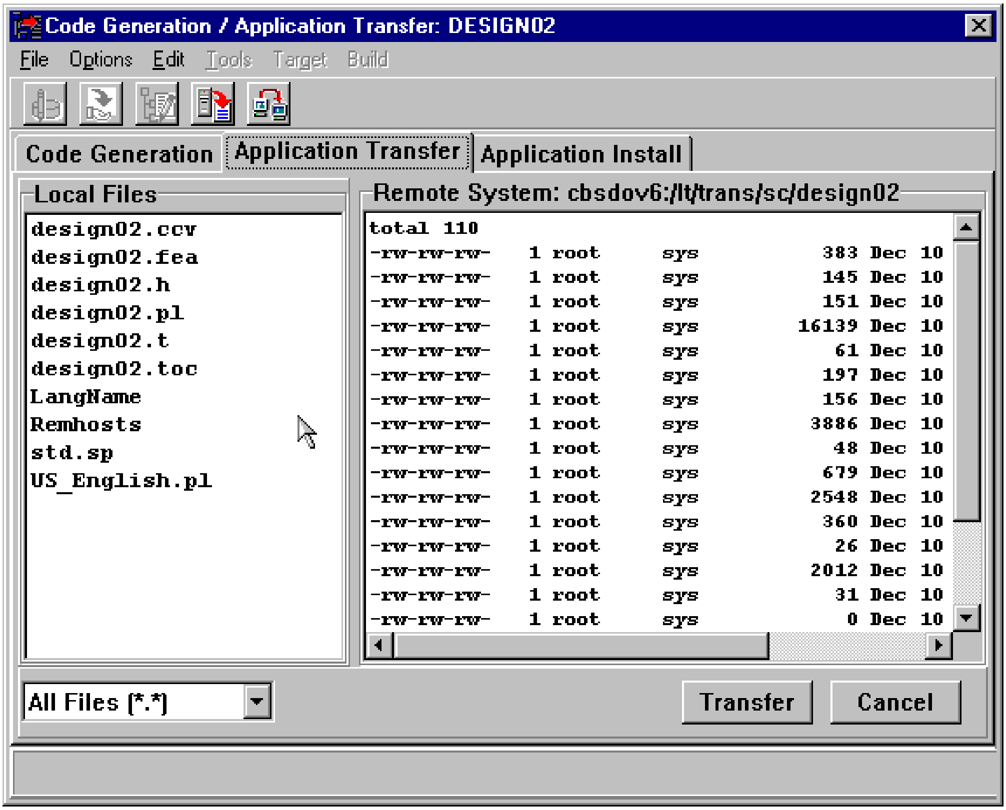


Figure 109. Code Generation/Application Transfer Window

3. Select the files you want to transfer.

   You can choose to look at or select only certain *types* of files (such as script files or database files). Use the pulldown menu below the **Local Files** pane to select the file types. (For details about the types of files generated by Voice@Work for application transfer, see “Code Generation/Application Transfer File Types” in Chapter 23, “The Code Generation Tool”.)

To select:

- A *single* file, click the filename in the **Local Files** pane.
- *Multiple contiguous* files, click the first filename, and shift-click the last filename.
- *Multiple discontiguous* files, click the first topic you wish to select, and control-click each other filenames you wish to select.
- *All* displayed files, do one of the following:
  - — Click the **Edit** menu, and then click **Select All.**
  - — Press CONTROL A.

4. If you want to view the files in the destination area before transferring your application, click the  button on the Toolbar.

   - If you are transferring over a network, Voice@Work displays all filenames on the remote host.
   - If you are transferring to floppy disks, Voice@Work displays all filenames on the current floppy disk.

5. Click the **Transfer** button in the lower right corner of the window.

   - If you are transferring to floppy disks and you selected the **Display Message When Files exist** option in the Application Transfer Options window, Voice@Work displays the filenames of any files on the current floppy disk and then displays a warning message (Figure 110).
     - — To cancel the Application Transfer operation, select **No**.
     - — To have Voice@Work writes the files to the floppy disk, overwriting any files that have matching filenames, select **Yes**.

Figure 110. Application Transfer Warning Message

NOTE:
If you are transferring to floppy disks, you must still transfer the files from floppy to the target system. See "Transferring Files from Floppy Disks to the Target System" below.

- If you are transferring over a network and you selected the **Display Message When Files exist** option in the Application Transfer Options window, Voice@Work displays the filenames of all files on the target system and checks to see if the application name of the application you wish to transfer already exists there. If the name:
  - — Does not exist, Voice@Work completes the Application Transfer by writing the files to the target system.
  - — Exists, Voice@Work displays a warning message similar to the one shown in Figure 110.

If you selected the **Display Message When Transfer is complete** option in the Application Transfer Options window, Voice@Work displays the following alert when the transfer is completed (Figure 111).

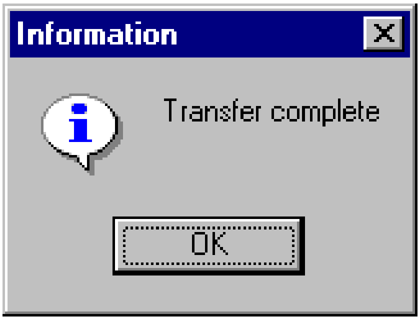


Figure 111. Information Window

Transferring Files from Floppy Disks to the Target System

Once you have transferred the application source files to floppy disks, you must transfer them to the target system.

NOTE:
If you selected **Network** as the **Type of Transfer** in the **Application Transfer Options** window, you do not need to perform this procedure.

To transfer files from floppy disk to the target system, do the following:

1. Log in to the target system as *root*.
2. Insert the floppy disk containing the application transfer files into the floppy disk drive.
3. At the system prompt, enter some form of the following command:

   scat [-o] ***<application_name>***

   where:

   - **[-o]** is an option that overwrites any files with matching filenames without warning you.
   - ***<application_name>*** is the name of the application.

   For example, if you want to transfer the files for an application named **acct_bal** from floppy disks to a target system, and you do not want the target system to warn you before overwriting files with matching filenames, enter: **scat -o acct_bal**

Code Generation/Application Transfer File Types

For information about the types of files available for application transfer, see "Code Generation Application Transfer File Types" in Chapter 23, "The Code Generation Tool".

Transferring Application Source Files

Voice@Work provides the option to transfer your applications' source files to the target system. Even though the files cannot be used by the target system, this can be helpful should you need assistance from technical support personnel in troubleshooting your applications.

To transfer your application source files to the target system, do the following:

1. Open the Code Generation / Application Transfer window. See "Opening the Application Transfer Window" above for the procedure.
2. Click the **Application Transfer** tab.
3. From the **Options** menu, select **Application Transfer**.

   Voice@Work displays the Application Transfer Options window (Figure 112).

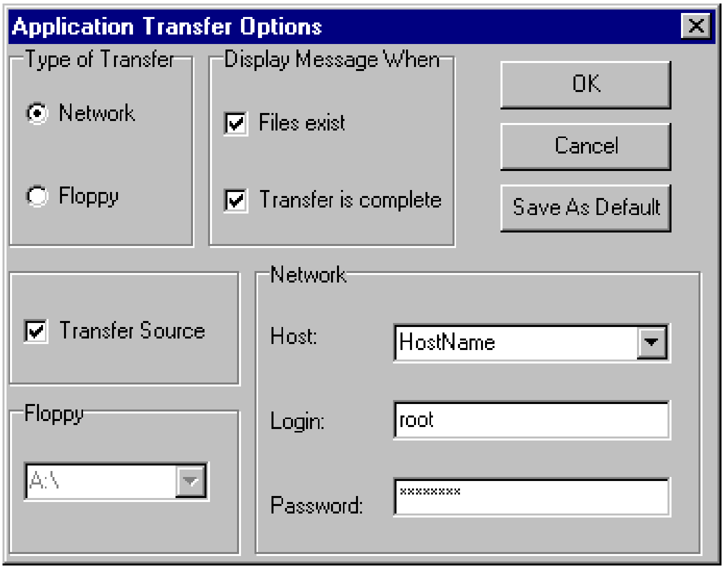


Figure 112. Application Transfer Options Window

4. Check the **Transfer Source** checkbox.
5. Click **OK**.

   The Application Transfer Options window closes and the Application Transfer window's appearance changes (Figure 113). Notice that, in addition to the **Local Files** and **Remote Files** panes, there is now a **Folders** pane.

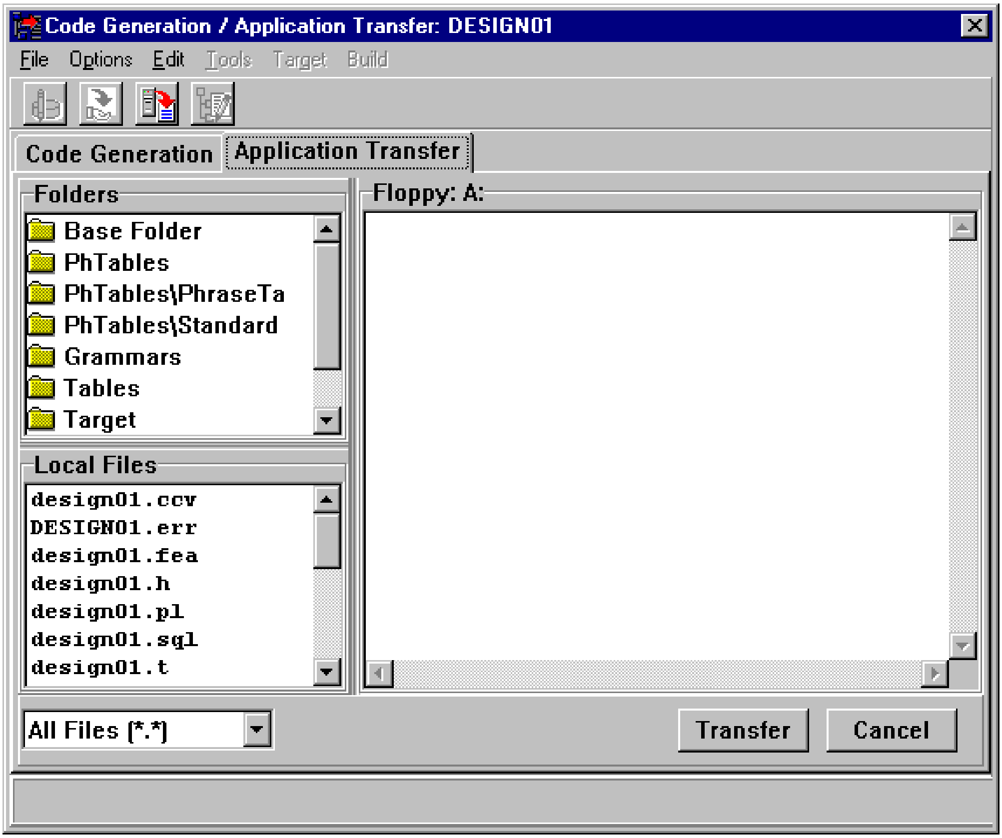


Figure 113. Application Transfer Window with Transfer Source Option Checked

6. Click on a folder to select the type of source files you want to transfer.

   NOTE:
   You can transfer the contents of only one folder at a time.

   The contents of the folder display in the **Local Files** pane.

7. In the **Local Files** pane, select the files you want to transfer.

   NOTE:
   You cannot transfer application source files by simply selecting the folder. Once you select the folder, you must also select the files you want to transfer from that folder.

8. Click the **Transfer** button.

Repeat steps 6 – 8 for each additional set of application source files you want to transfer.

Application Transfer Window Menus

The Code Generation/Application Transfer tools window has its own set of menus. Table 74 describes the Application Transfer menus and their options.

Table 74. Application Transfer Window Menus

Menu	Option	Comments/Descriptions
File	Open	Allows you to temporarily open another application *for application transfer purposes only.* You can: ■ Select the application of your choice from a standard **Open** dialog box. ■ View files for the selected application in the **Local Files** pane. ■ Work with the newly selected application as if it were the application currently open on the Voice@Work design pad. **NOTE:** This option remains active only as long as you are actually in the Application Transfer tab. When you leave the Application Transfer tab, Voice@Work reverts to the currently open application for Application Transfer.
	Exit	Closes the **Code Generation / Application Transfer** dialog box.
Options	Code Generation	Is not available from the Application Transfer tab view. Does not exist in the standalone version of the Application Transfer tool.
	Application Transfer	Opens the Application Transfer Options window.
Edit	Select All OR CONTROL A	Allows you to select all available Local Files for transfer to the target destination.
Tools	Inactive in the Applications Transfer tab view	
Target		
Build		

The Events String Editor

25

Overview

The Events String Editor allows you to determine which variables will be treated as call data events on the target voice response system.

The Events String Editor displays a list of all the variables defined for the application and a list of the variables selected to be treated as call data events.

Purpose

The purpose of this chapter is to describe the Events String Editor. Topics include:

- Opening the Events String Editor
- Selecting Variables to be treated as Call Data Events
- Removing Variables from the Selected Variables list
- Moving Variables on the Selected Variables List
- Events String Editor Menus

Opening the Events String Editor

To open the Events String Editor, do one of the following:

- From the **Tools** menu, select **Events String Editor**.
- Click the  button on the Toolbar.
- Press ALT T, then press E.

Voice@Work displays the Events Editor window (Figure 114).

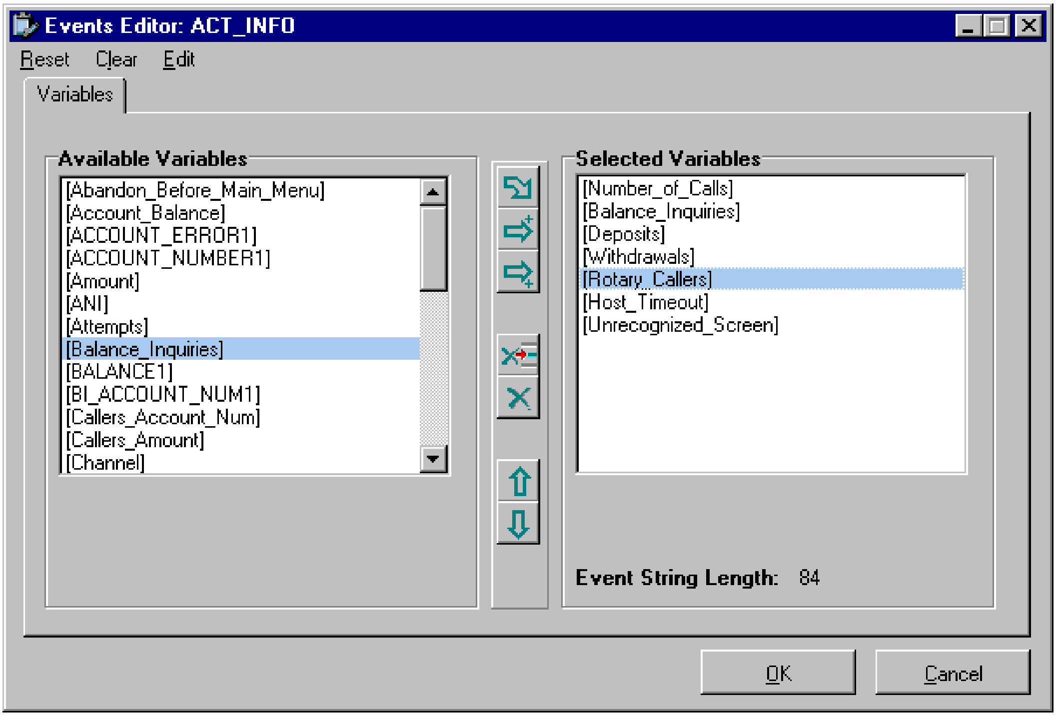


Figure 114. Events Editor Window

Selecting Variables

There are three methods for adding a variable to the Selected Variables list:

- Appending the variable to the end of the Selected Variables list
- Inserting the variable *before* a particular variable on the Selected Variables list
- Inserting the variable *after* a particular variable on the Selected Variables list

NOTE:
When you add a variable to the Selected Variables list, the Event String Length value at the bottom of the window increases by the length defined for that variable.

Appending a Variable

To append a variable to the end of the Selected Variables list, do one of the following:

- Double-click the name of the variable you want to add from the Available Variables list.
- Highlight the variable you want to add from the Available Variables list, and then select **Append Variable** from the **Edit** menu.
- Highlight the variable you want to add from the Available Variables list, and then click the **Append Variable** button.

Adding a Variable *Before* a Selected Variable

To insert a variable *before* a variable on the Selected Variables list, do the following:

1. Highlight the variable you want to insert from the Available Variables list.
2. Highlight the variable on the Selected Variable list you want it to go before.
3. Do one of the following:
   - Click the **Insert Variable Before** button.
   - From the **Edit** menu, select **Insert Variable**.

Adding a Variable *After* a Selected Variable

To insert a variable *after* a variable on the Selected Variables list, do the following:

1. Highlight the variable you want to insert from the Available Variables list.
2. Highlight the variable on the Selected Variable list you want it to go after.
3. Click the **Insert Variable After** button.

Removing Variables

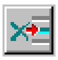
There are two methods for removing variables from the Selected Variables list:

- Removing a single variable
- Removing all variables

> **NOTE:**
> When you remove variables from the Selected Variables list, the Event String Length value at the bottom of the window decreases by the length defined for those variables.

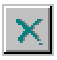
Removing a Single Variable

To remove a single variable from the Selected Variables list, do one of the following:

- Double-click the name of the variable you want to remove from the Selected Variables list.
- Click to select the variable you want to remove, and then select **Delete Variable** from the **Edit** menu.
- Click to select the variable you want to remove, and then click the **Delete Variable**  button.

Removing All Variables

To remove all variables from the Selected Variables list, do one of the following:

- Click the **Delete All Variables**  button.
- Click the word **Clear** on the menu bar.

Moving Variables on the Selected Variables List

To move a variable up or down on the Selected Variables list, do the following:

1. Click to select the variable you want to move.
2. To move the variable:
   - Up, click the **Move Up**  button or select **Raise Variable** from the **Edit** menu.
   - Down, click the **Move Down**  button or select **Lower Variable** from the **Edit** menu.

Events String Editor Menus

Table 75 summarizes the menu bar in the Events String Editor.

Table 75. Events String Editor Menu Options

Menu/Options	Description
Reset	Resets the Events String Editor to the selections and status it had when it was first opened.
Clear	Removes all variables from the Selected Variables list.
Edit	Contains the following options: ■ Add Variable – Inactive in the current version of Voice@Work. ■ Append Variable – Appends a variable selected in the Available Variables list to the *end* of the Selected Variables list. ■ Insert Variable – Inserts a variable from the Available Variables list *before* a selected variable in the Selected Variable list. ■ Delete Variable – Removes a selected variable from the Selected Variables list. ■ Lower Variable – Moves a variable downward on the Selected Variables list. ■ Raise Variable – Moves a variable upward on the Selected Variables list.

The Node Inspector 26

Overview

The Node Inspector displays different information and editing options for each node, depending on the node's predetermined attributes. It provides a convenient way for you to edit node attributes.

Purpose

The purpose of this chapter is to provide information about features common to all Node Inspector displays. For more information regarding the specifics of each node, see Chapter 31, "Node Descriptions".

Topics in this chapter include:

- Opening the Node Inspector
- Selecting a Node, to View or Edit Attributes
- Selecting a Set of Node Attributes
- Writing Node Notes
- Right-Click Menu Options for the Node Inspector

Opening the Node Inspector

To open the Node Inspector, do one of the following:

- From the **Tools** menu, select **Node Inspector**.
- Click the  button on the Toolbar.
- Press ALT T, and then press N.

Voice@Work displays the Node Inspector window (Figure 115).

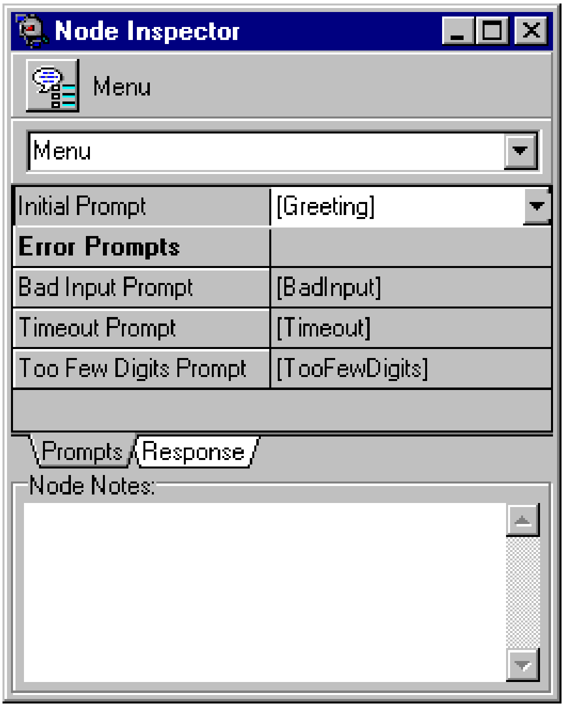


Figure 115. Node Inspector Window

Table 76 describes the elements of a typical Node Inspector window.

Table 76. Node Inspector Window Elements

Element	Description
Node selection pulldown menu	The pulldown menu at the top of the Node Inspector allows you to select which node to edit.
Node attribute fields	The fields below the Node selection pulldown menu allow you to set attributes for the selected node.
Node attribute tabs	When available, the tabs at the bottom of the Node attributes area allow you to select groups of attributes for editing.
Node Notes area	The Node Notes area at the bottom of the Node Inspector allows you to type in comments on the selected node.

Selecting a Node, to View or Edit Attributes

To view or edit a node's attributes, do the following:

1. Open the Node Inspector.
2. Select the desired node by doing one of the following:
   - From the active call flow, highlight the node in the design pad.
   - Use the Node Select pulldown menu, located near the top of the Node Inspector window—see Figure 115).

     NOTE:
     Only the nodes used in the active call flow are available from this pulldown menu.

   Once this menu is selected, you can view and edit the node's attributes. For information about editing a particular type of node, see Chapter 26, "The Node Inspector".

   To view additional information or hidden text, resize the Node Inspector window.

   Voice@Work displays the icon and name of the node type you are viewing or editing just above the Node Select pulldown menu.

Selecting a Set of Node Attributes

Many types of nodes organize sets of attributes on a single window. These attributes are related by function or type. If there is more than one set of attributes, each set is designated by a tab near the bottom of the Node Inspector window.

To select the set of attributes you want to view or edit, click its tab.

Writing Node Notes

The Node Notes area is located at the bottom of the Node Inspector window, if the Show Node Notes option is checked in the right-click menu. See the "Right-Click Menu Options for the Node Inspector" section below.

To write and save notes about a particular node, place the cursor in the text area and type the desired note text.

These notes stay with the node — for instance, if you copy and paste it — and can be viewed whenever the Node Inspector is active for that node. They also appear in the status bar at the bottom of the main window whenever the cursor rests on the node.

NOTE:
Node notes do *not* print when you print a report for the node using the Print command.

Right-Click Menu Options for the Node Inspector

Table 77 describes the right-click menu options for the Node Inspector.

NOTE:
The last four element options are available only with nodes for which variable elements are defined. These are the Set and Test node, the Trace Variables node, and the Load Variables node.

Table 77. Right Click Menu Options

Menu Option	Description
Show Hints	Toggles on/off. When checked (toggled on), allows a help window to pop up whenever the cursor rests over a field in the Node Inspector window.
Show Node Notes	Toggles on/off. When checked (toggled on), displays the Node Notes area of the Node Inspector window and allows you to enter, add, or edit comments. When unchecked (toggled off), hides the Node Notes area.
Stay on Top	Toggles on/off. When checked (toggled on), causes the Node Inspector window to remain on top of other open Voice@Work windows.
New Element	Allows you to add a new element to the variable definition.
Delete Element	Allows you to delete an element from a variable definition.
Move Element Up	Allows you to move an element upward in the variable definition table.
Move Element Down	Allows you to move an element downward in the variable definition table.

The Preference Editor

27

Overview

The Preference Editor allows you to set application and toolwide preferences for using Voice@Work. The Preference Editor has three tabs (sets of options), each of which controls a different set of preference options.

Purpose

The purpose of this chapter is to provide information and ideas for setting personal preferences in using Voice@Work. Topics include:

- Opening and Using the Preference Editor
- The Application Parameters Tab
- The Environment Tab
- The Printing Tab
- The Custom WholeWord Grammars Tab

Opening and Using the Preference Editor

To open the Preference Editor, do one of the following:

- From the **Tools** menu, select **Preference Editor**.
- Click the  icon on the Toolbar.
- Press ALT T, and then press P.

Voice@Work displays the Preference Editor window (Figure 116).

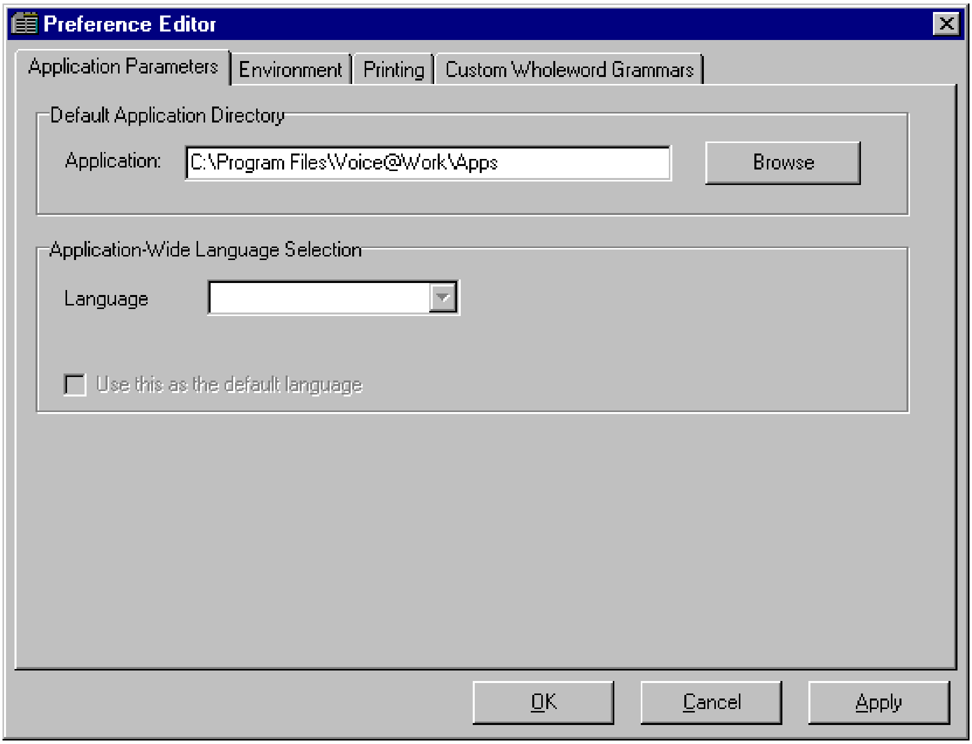


Figure 116. Application Parameters Tab of the Preference Editor

To set your preferences, apply the changes, and close the Preference Editor window, click **OK**.

To cancel any changes you have made and revert to the previous settings, click the **Cancel** button.

To apply any changes you have made without closing the window, click the **Apply** button.

NOTE:
The **Apply** button is only active if you have actually made changes in the Preference Editor.

The Application Parameters Tab

The Application Parameters tab (Figure 116) contains options for setting various parameters. Table 78 summarizes the options available on the Application Parameters tab.

Table 78. Application Parameters Tab Options

Parameter	Description/Comments
Default Application Directory	This parameters allows you to define the full pathname for the directory where applications are to be saved by default. You must include the drive letter. NOTE: When you reset the default pathname, the change does not become effective until you exit and restart Voice@Work. If you do not know the pathname, click the **Browse** button to search for the directory you want. NOTE: This default pathname is used by the **New**, **Open**, and **Save** commands on the File menu *when you first open Voice@Work only*. Once you select a different pathname, Voice@Work remembers and uses that pathname until you close it.

Continued on next page

Table 78. Application Parameters Tab Options — *Continued*

Parameter	Description/Comments
Application-Wide Language Selection	This parameter allows you to select from the pulldown menu the language you want to use in your application. If you want to use the language as the default for future applications, click the **Use this as the default language** checkbox. If you do not check this box, Voice@Work returns to the previously specified default language when you start a new application. The default is the language you selected as the default language when you first installed Voice@Work on your PC or whatever other language you have since set as the default language using the Preference Editor. **NOTE:** If you try to select a language that does not support Text-to-Speech options or FlexWord or WholeWord speech recognition, and you have used these resources in your application, Voice@Work displays a message instructing you to remove all uses of or references to these resources before you can select the new language. If you see any of these messages, you must remove any the appropriate resources before continuing with the language change.

Continued on next page

Table 78. Application Parameters Tab Options — *Continued*

Parameter	Description/Comments
Application-Wide Language Selection (cont.)	When you change the language for an application, Voice@Work automatically changes the following: ■ The EBS phrase table ■ WholeWord grammars ■ Variables to defaults When changing an application's language, you are responsible for: ■ Importing custom phrase tables for the new language ■ Modifying FlexWord lists **NOTE:** All resources used in the application (Text-to-Speech, WholeWord grammars, FlexWord lists, and EBS) must be available in the language you want to select. If not, Voice@Work does not allow you to change the language.

The Environment Tab

The Environment tab (Figure 117) contains options for setting your working environment in Voice@Work. Table 79 describes the options available on the Environment tab.

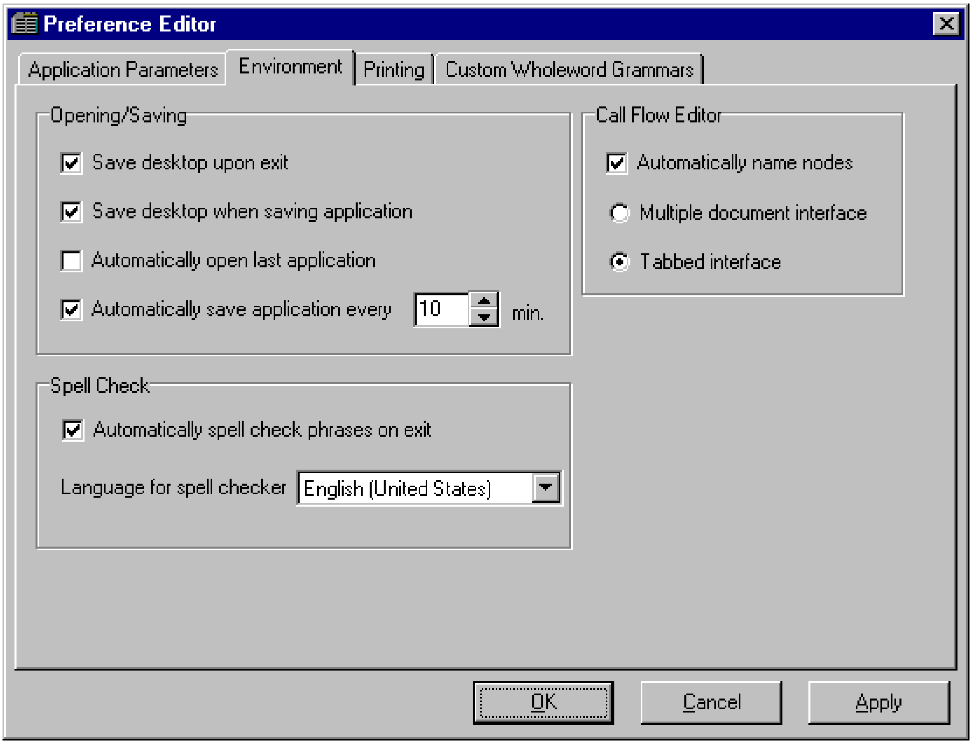


Figure 117. Environment Tab of the Preference Editor

Table 79. Environment Tab Options

Option Group	Option (and Type)	Description/Comments
Opening/Saving	Save desktop upon exit (checkbox)	When checked, saves your design pad (desktop) options so they are used whenever you reopen Voice@Work.
	Save desktop when saving application (checkbox)	When checked, saves your design pad (desktop) options whenever you save an application.
	Automatically open last application (checkbox)	Upon starting Voice@Work, automatically opens whatever application you were working on the last time you had Voice@Work open, when checked.
	Automatically save application (checkbox and pulldown menu)	When checked, saves your application automatically at the time interval you specify. Offers a range of 1–179 minutes. Setting this number to zero has the same result as unchecking the box.
Spell Check	Automatically Spell Check Phrases on Exit (checkbox)	When checked, uses the Microsoft Word Spell Checker to spell check phrase text whenever you exit the Phrase Editor. Voice@Work uses its own interface in accessing the spell checker tool. **NOTE:** This feature works only if you have both Microsoft Word and the Word Spell Checker installed on your PC.
	Language for Spell Checker (pulldown menu)	Defaults to the designated language for the Microsoft Word Spell Checker.

Continued on next page

Table 79. Environment Tab Options — *Continued*

Option Group	Option (and Type)	Description/Comments
Call Flow Editor	Automatically name nodes (checkbox)	When checked, suggests a name for each new node whenever you drag an icon onto the design pad. You may then either accept the suggested name or type in another name.
	Multiple Document Interface (radio button)	When selected, provides a different (nonmodal) window for each call flow in the application. This allows you to see more than one call flow at a time. **NOTE:** You must restart Voice@Work before this option takes effect.
	Tabbed Interface (radio button)	When selected, allows you to select call flows by clicking the call flow name tabs at the bottom of the design pad window. Each call flow, when selected, occupies the entire design pad window space. **NOTE:** You must restart Voice@Work before this option takes effect.

The Printing Tab

The Printing tab (Figure 118) contains options for printing reports in Voice@Work.

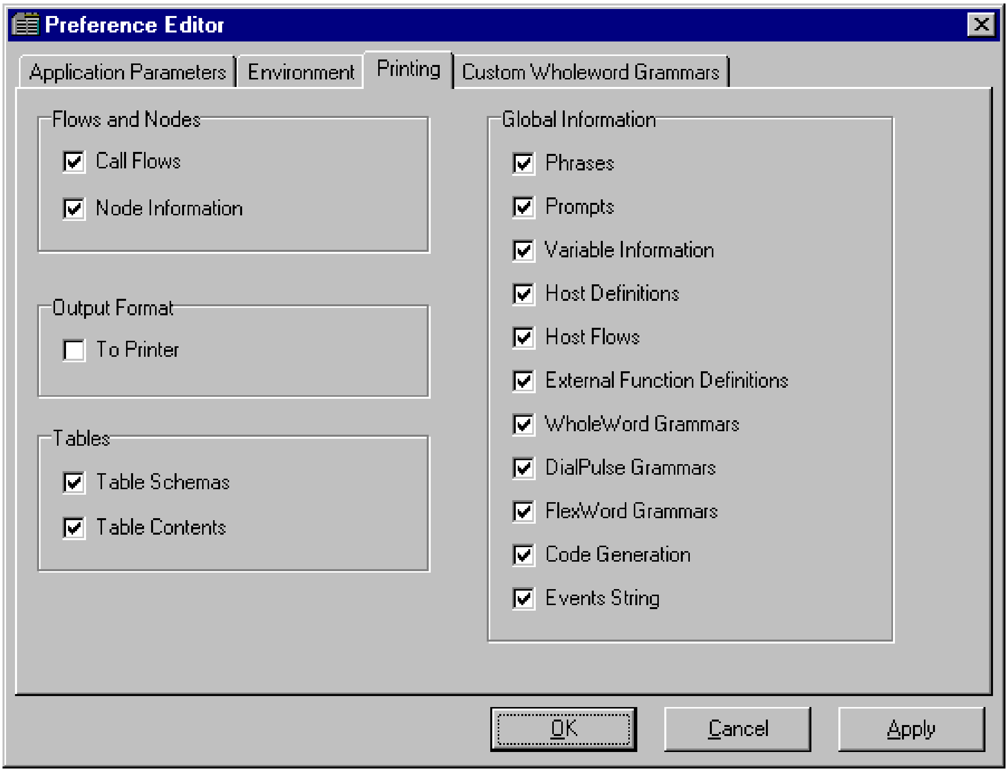


Figure 118. Printing Tab of the Preference Editor

All the options on this tab, except the Output Format **To Printer** option, toggle on/off printing options.

- When checked, the option is available for printing *if it exists in the application*. If nothing exists for that option, it does not appear in the Print Reports window, even if you have checked it here.
- When unchecked, the designated option is not available for printing even if it exists in the application.

The Output Format **To Printer** checkbox determines the default **Destination** in the Print Reports window.

- When checked, the default output format is to the **Default Printer**.
- When unchecked, the default output format is to a document (*.doc) file.

For more information about printing reports, see “The Print Reports Window” in Chapter 5, “The File Menu”.

The Custom WholeWord Grammars Tab

The Custom WholeWord Grammars tab (Figure 119) contains options for selecting, adding, and removing custom WholeWord grammars. Table 80 describes the options available on the Custom WholeWord Grammars tab.

For more information about custom WholeWord grammars in Voice@Work, see Appendix I, “Custom WholeWord Grammars”.

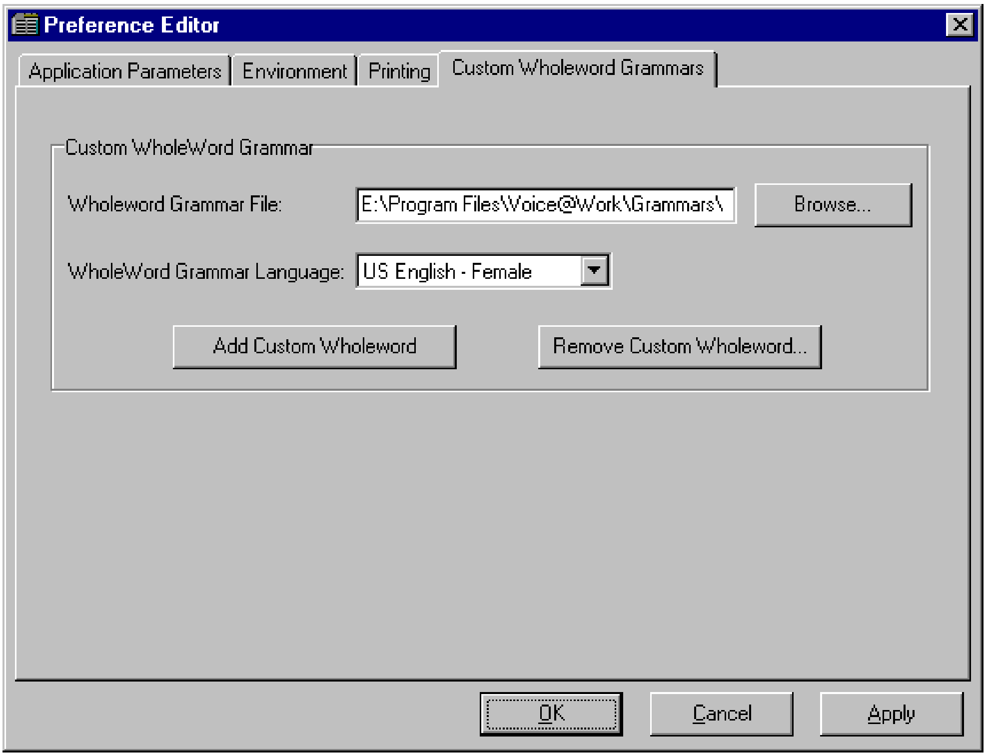


Figure 119. Custom WholeWord Grammars Tab of the Preference Editor

Table 80. Custom WholeWord Grammars Tab Options

Field/Option	Description/Comments
WholeWord Grammar File:	This field displays the full path and filename for a custom WholeWord grammar selected using the **Browse** button. Alternately, you can type in this field the full pathname for the custom WholeWord grammar you want to add or remove. **NOTE:** The designated custom WholeWord grammar file must have a **.grm** extension, or Voice@Work does not recognize it as a valid grammar.
Browse	This button is used to locate and select the specific custom WholeWord grammar you want to add or remove.
WholeWord Grammar Language:	This field displays the language being used for an application if one is open. If no application is open, this field displays the default language as set by the user during installation or by using The Application Parameters Tab. The dropdown menu contains a list of all the languages currently installed with Voice@Work. You can select a language other than the default language displayed. Whatever language you select in this field is the language that the selected custom WholeWord grammar file is to be associated with.
Add Custom WholeWord	This button adds the selected custom WholeWord grammar to the available Voice@Work application resources. When this button is clicked, the specified custom grammar is verified for the correct format and copied to the toolwide grammar directory for the designated language. Once you have added a custom grammar, it is available to any application that uses the designated language.

Continued on next page

Table 80. Custom WholeWord Grammars Tab Options — *Continued*

Field/Option	Description/Comments
Remove Custom WholeWord	This button removes the selected custom WholeWord grammar from the appropriate Voice@Work language directory. **NOTE:** Any applications already using this custom WholeWord grammar still have the grammar available, because each application makes a local copy of the grammar.

The Simulation Tool

28

Overview

The Simulation tool runs your application in a simulated environment. The Simulation tool is often helpful in finding problems within the application, thus reducing the amount of time required to test your application on the target system.

Purpose

The purpose of this chapter is to describe the Simulation tool and how to use it. Topics include:

- Opening the Simulation Tool
- Running an Application Simulation
- Examining the Simulation Script
- Right-Click Menu Options for the Simulation Tool

Opening the Simulation Tool

To start the Simulation tool, do one of the following:

- From the **Tools** menu, select **Simulation**.
- Click the  button on the Toolbar.
- Press ALT T, and then press S.

Voice@Work displays the Application Simulation window, and the simulation immediately starts running (Figure 120) from the currently highlighted node.

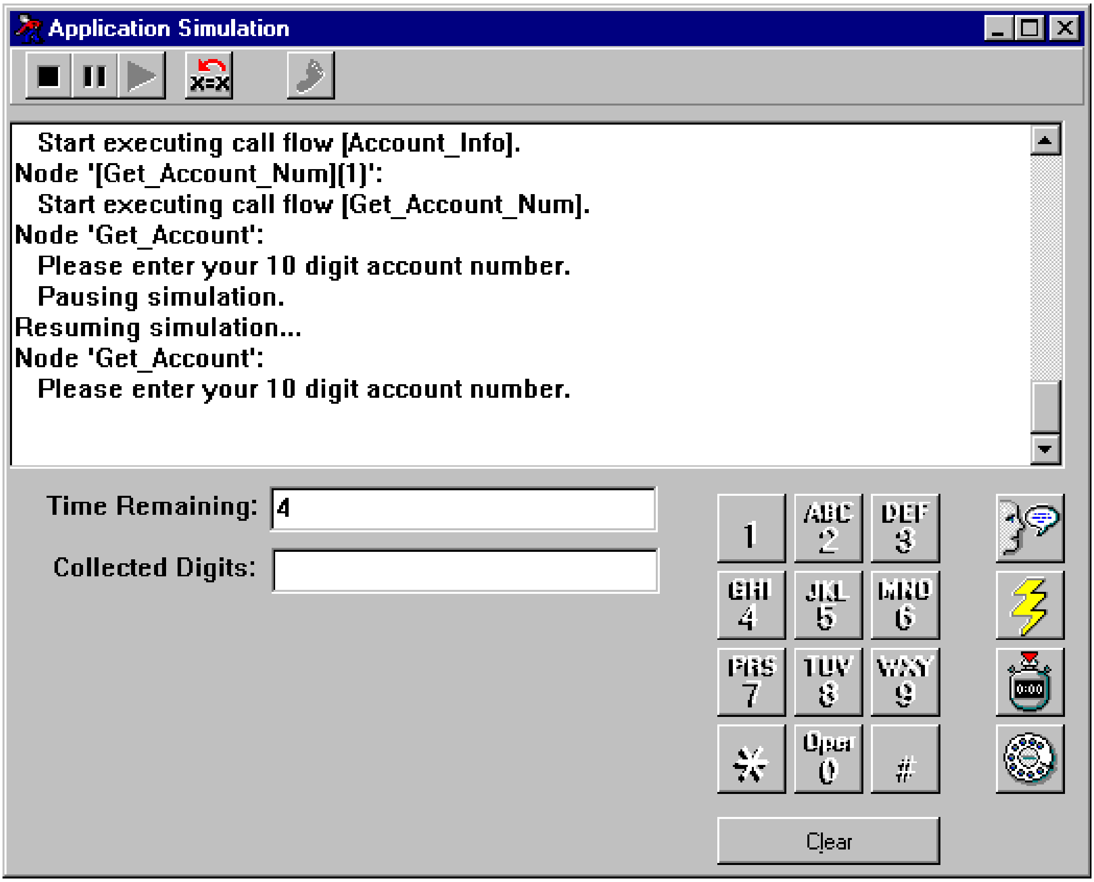


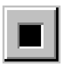
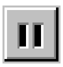
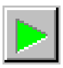
Figure 120. Application Simulation Window

NOTE:
The Simulation tool will not work if you attempt to start it from an empty branch. If an empty branch is highlighted and you try to run the Simulation tool, Voice@Work displays the following dialog box (Figure 121).

Figure 121. Empty Branch Start Simulation Dialog Box

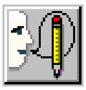
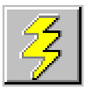
Table 81 describes the elements of the Application Simulation window.

Table 81. Application Simulation Window Elements

Element	Icon (if applicable)	Description
Stop button		Stops the simulation
Pause button		Causes the simulation to pause
Play button		Resumes running a simulation that has been paused or stopped, or runs a simulation that has ended
Reset Variables button		Resets all variables to their default values
Step-Through button		Allows you to “step through” a simulation, one node at a time

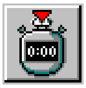
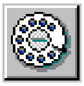
Continued on next page

Table 81. Application Simulation Window Elements — *Continued*

Element	Icon (if applicable)	Description
Simulation script field		Displays a script of all simulated events while the simulation is running
Time Remaining field		Displays a “countdown” for each node that uses a timed delay for responses.
Collected Digits field		Contains any digits that have been collected (using touchtone input or data from a table) and not used.
Touchtone keypad		Simulates caller input from a touchtone telephone.
Clear button		Clears any digits in the **Collected Digits** field. **NOTE:** You may also enter alphanumeric touchtone input from your computer keypad or keyboard. You cannot enter touchtone input from your telephone for the Simulation tool.
ASR button		Allows you to simulate spoken responses for FlexWord and WholeWord speech recognition features. **NOTE:** You must have an Initial Prompt defined for the node for this feature to work.
Asynchronous Event button		Simulates an asynchronous event. To simulate an asynchronous event (such as a customer hanging up in the middle of a node), press the Asynchronous Event button during a simulation. **NOTE:** If you have defined a routine to handle asynchronous events, pressing this button calls that routine.

Continued on next page

Table 81. Application Simulation Window Elements — *Continued*

Element	Icon (if applicable)	Description
Force Timeout button		Enables you to skip the “countdown”, or delay, that is programmed into many nodes, and go on to the next node.
DPR button		Simulates a Dial Pulse Recognition (DPR) response from the caller.

Running an Application Simulation

To run a simulation of an application or part of an application, do the following:

1. On the design pad, highlight the node or object you want the simulation to start from.
2. Open the Simulation tool.

   When you open the Simulation tool, the simulation starts running immediately.

   NOTE:
   You can use the Set Breakpoint option (from the right-click popup menu for each node) to “step through” a simulation or temporarily stop the simulation at points where you might want to look at what is going on within the application, check variable values, and so forth.

You do not have to start a simulation at the very beginning of your application. If you only want to test part of the application, you can choose any node within the application from which to start.

While the simulation is running, the script field displays a script that logs all simulated events.

If recorded speech it installed on your system and you have the appropriate hardware, the recorded speech plays at the appropriate place during the simulation.

The **Time Remaining:** field displays the amount of time left for the current node, where there is a built-in time limit.

To force a timeout for a node, press the **Force Timeout**  icon. Forcing a timeout enables you to skip the countdown, or delay, that is programmed into many nodes.

Using the **Force Timeout** button can lessen the time it takes to move through a simulation.

3. Respond to any prompts according to what you want to test.

   Most of the time, the response a caller is prompted for involves pressing a touchtone button on a telephone keypad. Use the touchone keypad in the Application Simulation window to simulate this input. Use the mouse to click the desired keys.

   To simulate a response from a dial pulse telephone (that is, from a caller who does not have touch tone response available), click the DPR icon when you encounter a prompt in which a touchtone or dial pulse response is required from the caller.

   - To cause the simulation to pause, click the **Pause** button.
   - To stop the simulation, click the **Stop** button.
   - To resume running the simulation, click the **Play** button.

   NOTE:
   When you resume a simulation, it starts at the beginning of the node that you stopped or paused on.

   - To step through the simulation one node at a time, click the **Step-Through** icon.

   When the simulation is finished, the message “End of simulation” appears in the script window.

4. (Optional) Do one of the following:

   - To reset the variables to their defaults — so you can run the simulation again, for example — press the **Reset Variables** button.
   - To clear Collected Digits, press the **Clear** button.
   - To run the simulation again, select the node you want to start from, and then click the **Play** button.

Examining the Simulation Script

While a simulation is running, the script field displays a script that logs all simulated events. This script is kept until you close the Application Simulation window.

To view parts of the script that have “scrolled up” in the field and are no longer visible, use the scrollbar at the right.

There are times when it is helpful to see the entire script at once. To do this, you can print the script.

To print a copy of the script, do the following:

1. Right-click in the script window and select **Select All** from the popup menu.

   Voice@Work highlights the script text.

2. Do one of the following:

   - Press CONTROL C on your keyboard.
   - Right-click in the script window again, and select **Copy** from the popup menu.

3. Paste the text into a document in the text processor of your choice, such as Notepad or Wordpad.
4. Use the Print command in the text processor to print out the script.

Right-Click Menu Options for the Simulation Tool

Table 82 describes the right-click menu options for the Simulation tool.

Table 82. Right-Click Menu Options for the Simulation Tool

Option	Action
Undo	Inactive in the current version.
Cut	Removes the highlighted text and copies it to the clipboard.
Copy	Copies the highlighted text to the clipboard, leaving the original where it is.
Paste	Pastes a selection previously copied to the clipboard to the current cursor location.
Delete	Removes the highlighted text completely, that is, does not copy it to the clipboard.
Select All	Highlights all text in the script window so you can cut, copy, or delete it.

The Verify Design Tool 29

Overview

The Verify Design tool allows you to check your application design for possible errors or omissions. When you open the Verify Design tool, it automatically searches through each call flow of your application. The results are displayed in the Verification Results window.

Purpose

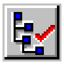
The purpose of this chapter is to describe the Verify Design tool and its use. Topics include:

- Running the Verify Design Tool
- Using Verification Results
- Right-Click Menu Options for the Verify Design Tool

Running the Verify Design Tool

The Verify Design tool runs automatically when you open it.

To open the Verify Design tool, do one of the following:

- From the **Tools** Menu, select **Verify Design**.
- Click the  button on the Toolbar.
- Press [ALT] [T], and then press [Y].

Voice@Work performs the design verification actions and displays the results in the Verification Results window (Figure 122).

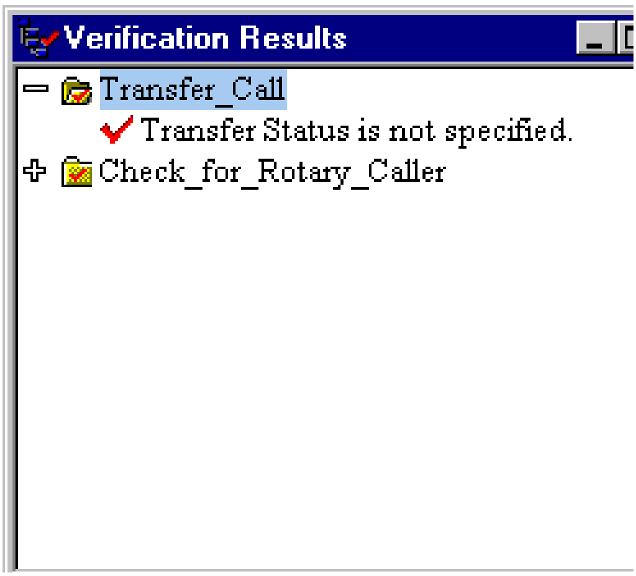
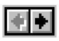


Figure 122. Verification Results Window

Table 83 describes the elements of the Verification Results window.

Table 83. Verification Results Window Elements

Element/Icon	Description
Results display pane	Displays the results of the verification
	Closed directory symbol—indicates that possible errors at this node are hidden To view the possible errors in this node, double-click the icon to expand it.
	Open directory symbol—indicates that possible errors at this node are displayed below the node name
	Indicates a possible error or omission. The node where the possible error or omission is located is listed just above this description.
Call flow tabs	Used to select the call flow to view verification results. The results for each call flow are contained in a separate display pane.
	Allows you to scroll to different call flows to view possible errors

If the Verify Design tool finds no errors or omissions, the following message box displays (Figure 123).

Figure 123. Verification Results “No Error” Message

Using Verification Results

When you run the Verify Design tool, Voice@Work displays the results of the search in the Verification Results window (Figure 122).

NOTE:
The results of the search indicate only *possible* errors or omissions. You must make the final determination whether a “flagged” node really is an error or omission.

- The icons designate the names of nodes where the Verify Design tool found possible errors or omissions.
- If the directory symbol is open (default), the lines immediately below marked with the icon describe the possible errors or omissions.

To locate the node within its call flow, double-click the name or the description of the node.

To view another call flow's search results, click that call flow's tab at the bottom of the window.

Right-Click Menu Options for the Verify Design Tool

Table 84 describes the right-click menu options available from the Verification Results window.

NOTE:
The right-click popup menu only appears when you right-click an entry in the Verification Results window.

Table 84. Right-Click Menu Options for the Verification Results Window

Menu Option	Description/Action
Edit	Locates the node in the design pad, from where you can edit it.
Verify Node	Verifies only the selected node for possible errors or omissions.
Verify Call Flow	Verifies only the current call flow.
Verify Design	Verifies the entire application design.
Expand	Expands a collapsed node entry, indicated by the icons.
Collapse	Collapses an expanded node entry, indicated by the icons.
Expand All	Expands all node entries in the current call flow.
Collapse All	Collapses all node entries in the current call flow.

Working with Nodes 30

Overview

Nodes are the basic "building blocks" used to create an application. You use nodes to build your call flows by dragging node objects or icons from the Node Palette to the design pad.

Voice@Work offers 24 different nodes, each with its own attributes and behaviors.

Purpose

The purpose of this chapter is to provide general information about nodes and their use in Voice@Work applications. Nodes are listed and described in detail in Chapter 31, "Node Descriptions". Topics include:

- Placing a Node in a Call Flow
- About Node Branches (including inserting and renaming branches)
- Editing Node Attributes
- Right-Click Menu Options

Placing a Node in a Call Flow

To place a node in a call flow, drag the node's icon from the Node Palette to the design pad. Once you place the node where you want it, the Node Inspector opens. You can use it to view, define, and edit the node's attributes.

Drop the icon onto any open branch (see the "About Node Branches" below). Or, if you do not want to connect the node to the call flow, drop the icon below the last node.

You can also drop the node *before* an existing node, including at the top of a call flow, by dropping it on top of an existing first-level node (Figure 124). Voice@Work does not allow you to drop a node on top of a node of any other level.

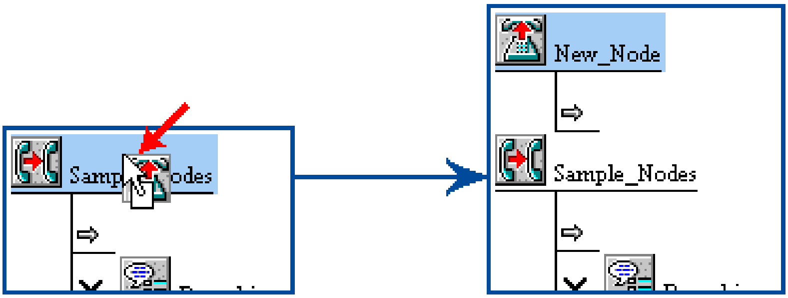


Figure 124. Dropping a Node onto an Existing Node (with the Result)

While you are dragging a node, the appearance of your cursor changes to let you know whether you can paste the node where the cursor is. When the cursor appears as an arrow (Figure 124), you can drop the node where you want it — either above the node or onto the empty branch. When the cursor appears as a circle with a slash across it (Figure 125), if you drop the node, Voice@Work places it at the end of the active call flow. Voice@Work also places the node at the end of the active call flow if the cursor is located after the end of the last node in the active call flow.

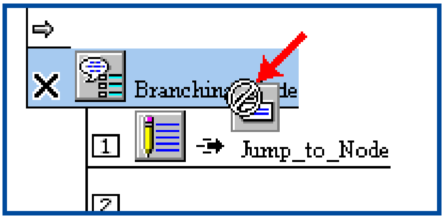


Figure 125. Cursor Change Showing Invalid Drop Status

You can also drag and drop nodes (and all subordinate nodes) to move them within a single call flow. When you place your cursor on the node you want to move, the cursor changes as described above to let you know where you can drop it. This is identical to a cut-and-paste operation.

Voice@Work also allows you to select multiple nodes within a call flow, but they must be contiguous nodes. That is, they must have the same “parent” node and be adjacent to each other on the call flow. As with single nodes, all subordinate nodes are copied or moved with the parent nodes. For more information about selecting, copying, and pasting multiple contiguous nodes, see “More about Cut/Copy/Paste” in Chapter 6, “The Edit Menu”.

For more details on moving and manipulating nodes in a call flow, see “Right-Click Menu Options” below.

About Node Branches

Node branches are indicated by lines leading down and to the right from a node (Figure 126).

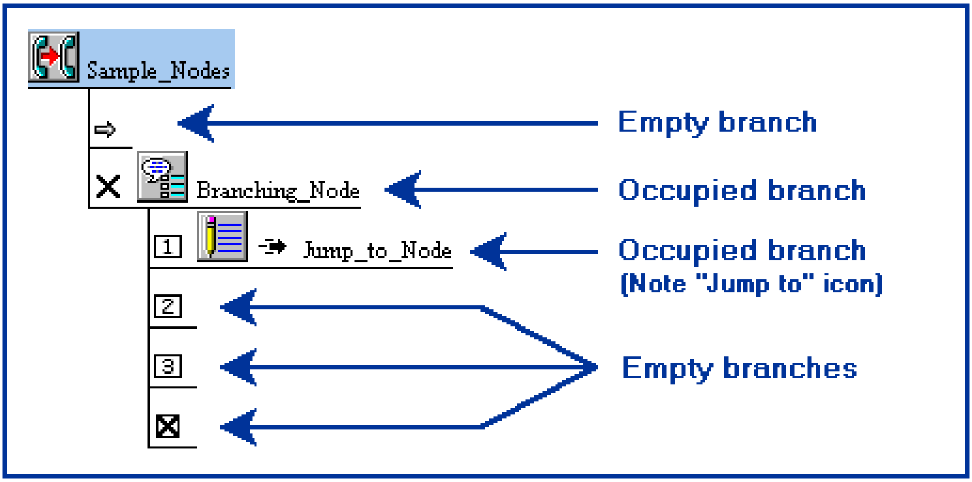


Figure 126. Node Branch Example

You can drag-and-drop a new node icon onto any empty branch. An empty branch is indicated by a horizontal line that does not connect to any node.

Inserting and Renaming Branches

Some node types allow you to insert new branches under them or to rename existing default branches.

When you select **Insert Branch** or **Rename Branch** from a right-click popup menu, the results depend on what type of node you are working with. Nodes that allow you to insert and rename branches include:

- The Menu Node
- The Automenu Node
- The Prompt and Collect Node
- The Time Branch Node
- The Call Flow Node
- The Set and Test Node

Editing Node Attributes

Use the Node Inspector to view and edit the attributes and elements of a node. See Chapter 26, "The Node Inspector", for additional information.

In general, to enter or edit values for a node's attributes, you must do one of the following:

- Type the literal value in the appropriate field yourself.
- Make a selection from a pulldown menu.
- Drag and drop a resource from the appropriate resource manager.

For details specific to each node, see the appropriate node description in Chapter 31, "Node Descriptions".

Right-Click Menu Options

The following are general descriptions of the right-click menu options available for nodes and for empty node branches.

NOTE:
Not all options are available for each node type. These are general descriptions only. For the specific details of how these options work with specific types of nodes, see the appropriate node description in Chapter 31, "Node Descriptions".

Right-Click Menu Options for Nodes

Table 85 describes the menu options available by right-clicking a node.

NOTE:
Parentheses ( ) around a menu option name indicate that the option is not available for all node types.

Table 85. Right-Click Menu Options for Nodes

Menu Option	Action
Cut	Removes the selected node and any subordinate nodes and copies it to the clipboard.
Copy	Copies the selected node and any subordinate nodes to the clipboard.

Continued on next page

Table 85. Right-Click Menu Options for Nodes — *Continued*

Menu Option	Action
Paste	Inserts any nodes on the clipboard at the point where the cursor is, subject to the paste limitations described in Chapter 6, ‘‘The Edit Menu’’. This option is only available once you have done a cut or copy action.
Branch Notes	Opens the Branch Note dialog box (Figure 127), where you can enter a short description for the branch. Branch notes are displayed in the main window status bar whenever the cursor is over that branch.
(Delete Branch)	When available, this option allows you to delete the entire branch from the selected node. **NOTE:** This action cannot be undone.
(Insert Branch)	When available, this option allows you to insert a new branch into the call flow. See ‘‘Inserting and Renaming Branches’’ below.
(Rename Branch)	When available, this option allows you to rename the branch. See ‘‘Inserting and Renaming Branches’’ below.
(Receive Field Mappings)	When available (Host Send and Host Send/Receive nodes only), this option maps fields received from the remote host to the appropriate fields in the selected Host Definition. This option is only available if: ■ At least one Host Definition Editor --> Field Definition tab --> Direction column field value is set to **From Host** or **Both**. ■ A new branch is inserted under the appropriate node.
Delete Node	Removes the selected node and any subordinate nodes from the call flow. **NOTE:** This action cannot be undone.
Disconnect Node	Separates the selected node and any subordinate nodes from the call flow.

Continued on next page

Table 85. Right-Click Menu Options for Nodes — *Continued*

Menu Option	Action
(Edit Flow)	(Active only with the Call Flow node) Opens the “encapsulated node” call flow, making it the active call flow.
Edit Node	Opens or activates the Node Inspector, where you can edit the selected node's attributes.
(Find Jumped To Node)	(Active only for nodes that have the icon following the node icon) Allows you to find the node that the selected node jumps to.
Encapsulate	Converts the selected node to a Call Flow node and encapsulates that and any subordinate nodes, making them into a separate call flow.
(Expand Node)	Reverses the Encapsulate command, expanding an encapsulated node and moving the components from separate call flow status to incorporate them into the parent call flow.
(Hide / Show Execution)	When available, hides or shows the execution of the node during Simulation.
Print Node	Prints out a summary of the selected node's attributes.
Rename Node	Allows you to rename the node.
Set / Clear Breakpoint	Stops the execution of a Simulation at the node where it is set. A breakpoint is designated in a call flow with the icon.
Verify Node	Opens and runs the Verify Design tool on the selected node only.
Print Call Flow	Prints out the active call flow similar to the design pad display of the call flow.
Verify Call Flow	Opens and runs the Verify Design tool on the active call flow only.

Continued on next page

Table 85. Right-Click Menu Options for Nodes — *Continued*

Menu Option	Action
Close	Displays a list of all the currently open call flows. Selecting a call flow from the list closes it. **NOTE:** This option is available only when "Tabbed Interface" is selected on the Environment tab of the Preference Editor.
Open	Displays a list of all the currently closed call flows. Selecting a call flow from the list opens it. **NOTE:** This option is available only when "Tabbed Interface" is selected on the Environment tab of the Preference Editor.

Right-Click Menu Options for an Empty Branch

Table 86 describes the menu options available by right-clicking on an empty branch under a node.

NOTE:
Parentheses ( ) around a menu option name indicate that the option is not available for all node types.

Table 86. Right-Click Menu Options for an Empty Branch

Menu Option	Action/Description
Cut	Grayed out — inactive on an empty branch.
Copy	Grayed out — inactive on an empty branch.
Paste	Inserts any nodes on the clipboard at the point where the cursor is, subject to the paste limitations described in Chapter 6, ''The Edit Menu''. This option is only available once you have done a cut or copy action.

Continued on next page

Table 86. Right-Click Menu Options for an Empty Branch — *Continued*

Menu Option	Action/Description
Branch Notes	Opens the Branch Note dialog box (Figure 127), where you can enter a short description for the branch. Branch notes are displayed in the main window status bar whenever the cursor is over that branch.
(Delete Branch)	When available, allows you to delete the entire branch from the selected node. **NOTE:** This action cannot be undone.
(Insert Branch)	When available, allows you to insert a new branch into the call flow (see "Inserting and Renaming Branches" below).
(Rename Branch)	When available, allows you to rename the branch (see "Inserting and Renaming Branches" below).
Connect Node	Opens a dialog box displaying a list of any nodes in the call flow that are currently disconnected. Selecting a node from the list connects it to the current call flow at the node where this option was selected.
Copy Call Flow	Opens a dialog box displaying a list of all call flows currently defined for the application. Selecting a call flow from the list copies it to the current branch where this option was selected. **NOTE:** The [Main] call flow and [Host Flow] call flows cannot be copied.
Edit Node	Opens the Node Inspector, where you can edit the selected node's attributes.
Jump to Node	Opens a dialog box displaying a list of all nodes in the call flow to which you may jump from the currently selected node.
Reuse Call Flow	Similar to **Copy Call Flow**, except that the currently active call flow is not displayed and cannot be selected. Provides a link to the selected call flow, but does *not* make a copy of it

Continued on next page

Table 86. Right-Click Menu Options for an Empty Branch — *Continued*

Menu Option	Action/Description
Print Call Flow	Prints out the active call flow similar to the design pad display of the call flow.
Verify Call Flow	Opens and runs the Verify Design tool on the active call flow only.
Close	Displays a list of all the currently open call flows. Selecting a call flow from the list closes it. **NOTE:** This option is available only when "Tabbed Interface" is selected on the Environment tab of the Preference Editor.
Open	Displays a list of all the currently closed call flows. Selecting a call flow from the list opens it. **NOTE:** This option is available only when "Tabbed Interface" is selected on the Environment tab of the Preference Editor.

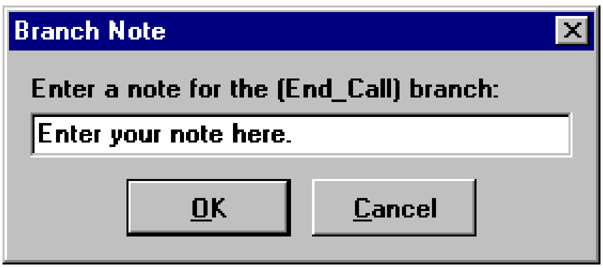


Figure 127. Branch Notes Dialog Box

Node Descriptions 31

Overview

Nodes are the basic “building blocks” used to create an application. Voice@Work offers 24 different types of nodes, each with its own attributes and behaviors.

You can use the Node Inspector to view and edit the attributes and elements of a node.

Purpose

The purpose of this chapter is to list and describe the nodes used in Voice@Work applications. Topics include:

- List of Nodes
- Control Nodes
- Telephony Nodes
- IVR (Interactive Voice Response) Nodes
- Database Nodes
- Host Nodes
- Miscellaneous Nodes

List of Nodes

The following is a list of all the nodes in Voice@Work. To learn more about a particular node, see the section for that node in this chapter.

- Control Nodes:
  - Call Flow
  - Return
  - Set and Test
  - Time Branch
  - Asynchronous Event
- Telephony Nodes:
  - Answer Call
  - Transfer Call
  - Disconnect Call
- IVR (Interactive Voice Response) Nodes:
  - Announcement
  - Menu
  - Prompt and Collect
  - Automenu
  - Voice Capture
- Database Nodes:
  - Query Table
  - Update Table
  - Add Record
  - Delete Record
- Host Nodes:
  - Host Connection
  - Host Send
  - Host Send/Receive
- Miscellaneous Nodes:
  - Inline Code
  - Trace Variables
  - Save and Exec
  - Load Variables

Control Nodes

The following sections detail the control nodes.

The Call Flow Node

The Call Flow node is used to create a new call flow or to encapsulate a set of nodes.

- Use the Call Flow node to create a new call flow by dragging the Call Flow node icon to any open branch of a call flow. You can also create a "disconnected" call flow by dragging the Call Flow node to a call flow window without dropping it onto an open branch.
- Use the Call Flow node to encapsulate a portion of a call flow and make a new, separate call flow out of it. The new call flow can then be viewed in its own call flow window on the design pad.

You can reuse or copy the encapsulated node anywhere in the application, as if it were a single node. This node is useful when you want to perform the same operation in more than one place in the application, much like a subroutine.

You can also encapsulate a large call flow within an application. By encapsulating a large call flow, several nodes can be collapsed into one node until you want to expand it. Encapsulating nodes can also improve the readability of large applications.

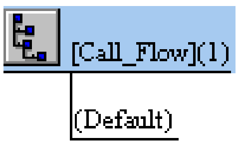


Figure 128. Call Flow Node

Behavior

- When this node is reached, the encapsulated call flow is executed.
- Like a function in a high-level language, the Call Flow node returns a value. Each return value is represented by a Return node in the call flow definition and by a branch on an instance of the call flow.

- All branches from nodes contained in the encapsulated node return to the Default branch unless additional branches are inserted. If a branch is added to the encapsulated node, it appears as a Return node in the call flow.
- Any branch of the call flow that does not return to a specified branch within its parent call flow returns to the Default branch.

Attributes

Inserting and Renaming Branches

When you insert a Call Flow node in a call flow, it creates one branch named (by default) "Default." This branch cannot be renamed or deleted.

When you select **Insert Branch** or **Rename Branch** on a Call Flow node, Voice@Work displays the Name Exit Condition dialog box (Figure 129).

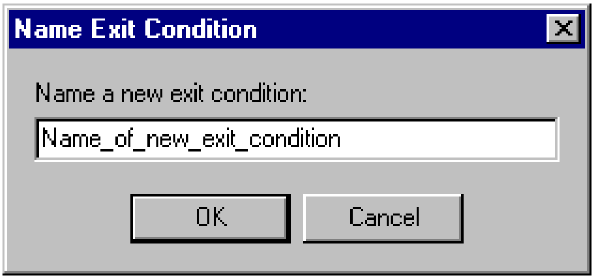


Figure 129. Name Exit Condition Dialog Box

Type the name you want for the branch, then click **OK**.

When naming branches, keep the following in mind:

- Names can be any length. You should, however, try to keep branch names as short as possible.
- Do not use spaces in names. Voice@Work ignores and deletes spaces when creating the branch name.

NOTE:
You may want to use an underscore (_) symbol in place of a space.

- Branch names are case sensitive. Voice@Work preserves cases as you assign them when naming the branch. The one exception is that the first letter of a name is always converted to upper case, regardless of how you type it in. For example, if you typed in **new_Branch** as the branch name, Voice@Work converts it to **New_Branch** when creating the branch.

  NOTE:
  Although Voice@Work preserves cases when saving names and transfers them to the target system with cases preserved, you cannot have two names with the same letters. If, for example, you have a branch called **New_Branch**, you cannot also create a branch called **NEW_BRANCH**.

Other Attributes

Other attributes are assigned and edited using the Node Inspector (Figure 130).

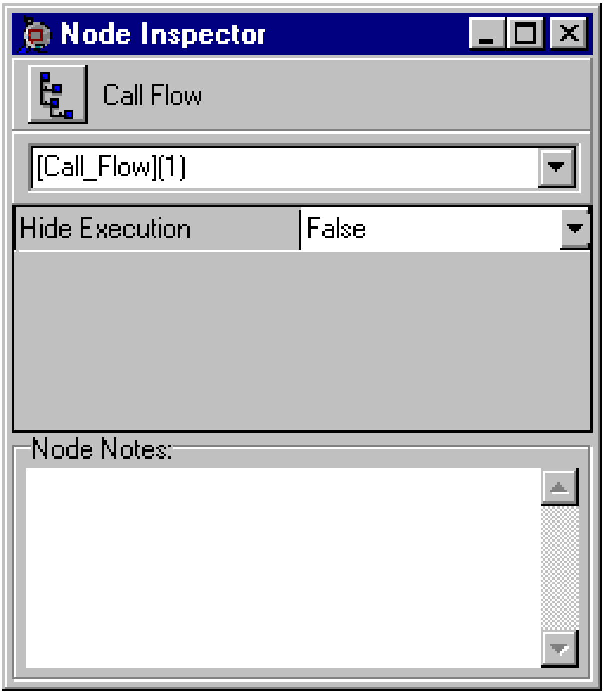


Figure 130. Node Inspector — Call Flow Node

- The number in parentheses after the Call Flow name represents each instance of the call flow used in the application. For example, the first time you use the node in an application, (1) follows the call flow node name. The second instance is appended with (2), and so on.

- **Hide Execution** – When set to **True**, this attribute hides the execution of the encapsulated node during Simulation.
- To locate and edit the call flow in the encapsulated node, click the node to highlight it, and then select **Edit Flow** from the popup menu. Voice@Work displays the encapsulated call flow on the design pad.

The Return Node

The Return node stops the execution of the call flow that contains it.

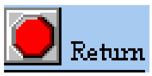


Figure 131. Return Node

Behavior

- In the [Main] call flow, the Return node stops the execution of the application.
- In any other call flow, the Return node results in the call flow returning and taking the branch with the same name as the Return node.
- The Return node terminates both the call (disconnects) and the application script.

Attributes

Attributes are assigned and edited using the Node Inspector (Figure 132). There are no user-configurable attributes for the Return node.

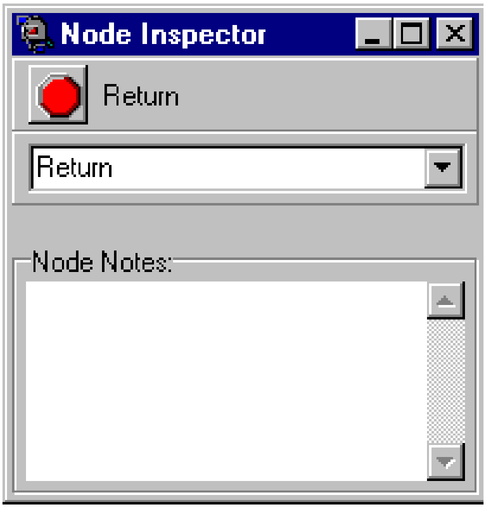


Figure 132. Node Inspector — Return Node

The Set and Test Node

The Set and Test node allows you to manipulate and evaluate a wide range of variable expressions and branch conditions. This node has two basic uses: It allows you to set variables and to test conditions for variables' use. The kinds of variable expressions you can set and test include the following:

- Arithmetic
- Assign
- Concat
- ItemCount
- ItemOf
- Length
- NextRecord
- Parse
- PrevRecord
- Substring

NOTE:
Before you can set test conditions for any of these expressions, you must first add a branch to the node.

Example of Use

ABC Automotive wants to perform a telephone customer survey. The survey consists of six questions. The company wants to track how many participants were involved in the survey. Callers must respond to at least four questions to be considered a participant.

The Set and Test node is used to set the value of a variable to the number 4. Each time the caller responds to a question, the variable is decremented. At the end of the survey, if the variable's value is 0 or less, the caller is added to the number of participants in the survey.

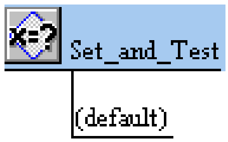


Figure 133. Set and Test Node

Behavior

- The behavior of the Set and Test node depends largely on what expressions and conditions are set for the variable.
- If no logical expression turns out to be true, the **Default** branch is taken.
- Expressions are always executed first, in the order in which they appear on the Expressions tab. The Branch Conditions, if any, are then executed.

Attributes

Inserting and Renaming Branches

When you select **Insert Branch** or **Rename Branch** from the right-click popup menu on a Set and Test node, Voice@Work displays the Set Test Branch dialog box (Figure 134).

Figure 134. Name Set Test Branch Dialog Box

Type the name you want for the branch.

Click **OK**.

Other Attributes

Other attributes are assigned and edited using the Node Inspector (Figure 135).

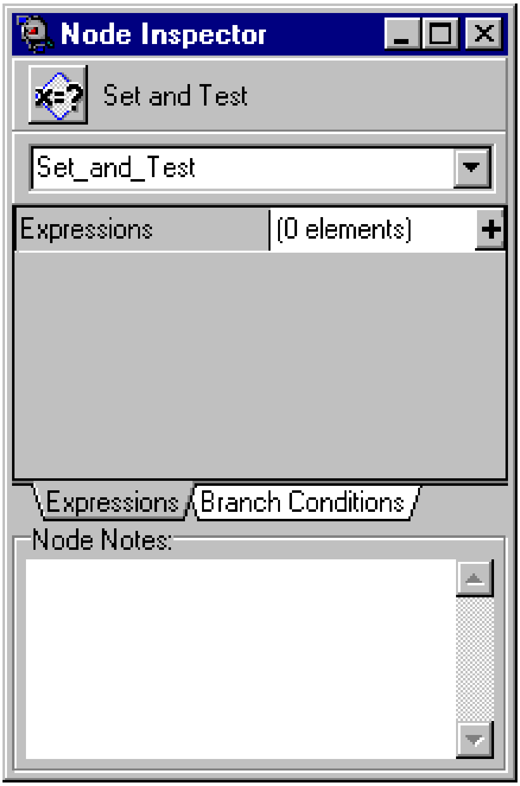


Figure 135. Node Inspector — Set and Test Expressions

Expressions Tab

- **Expressions**
  - — To add an expression, click the button.
  - — To set an expression's attributes, select an element, and then click the button (Figure 136). Voice@Work displays the Set Test Expression Editor window. For more information about using this editor, see "Set Test Expression Editor" below.

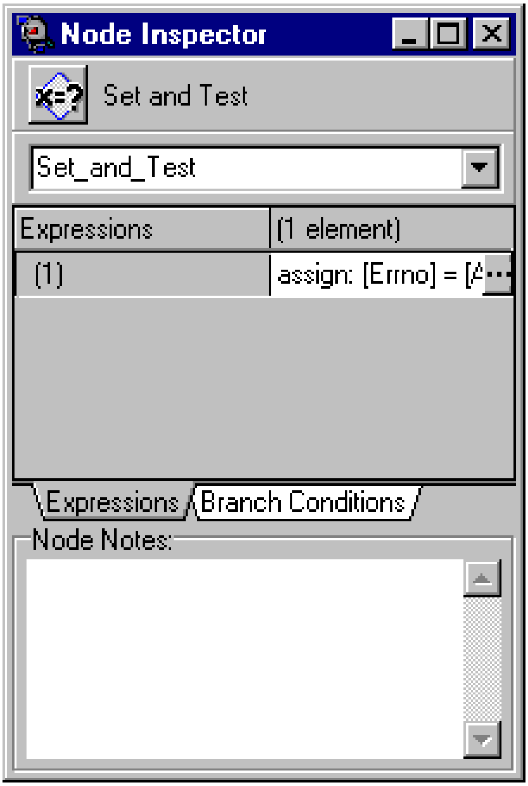


Figure 136. Node Inspector — Set and Test Node, Expressions Tab

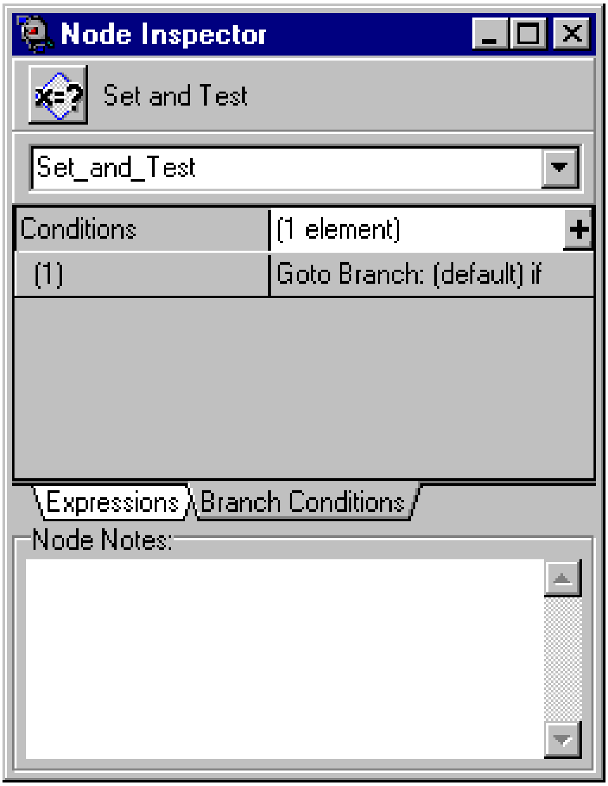


Figure 137. Node Inspector — Set and Test Node, Branch Conditions Tab

Branch Conditions Tab

- **Conditions**
  - — To add a Branch Conditions statement, click the button.

NOTE:
You must add at least one branch to this node in the call flow before you can set Branch Conditions elements. If you attempt to set branch conditions for an element and there is no branch other than the default branch on the node, Voice@Work displays the following error message shown in Figure 138.

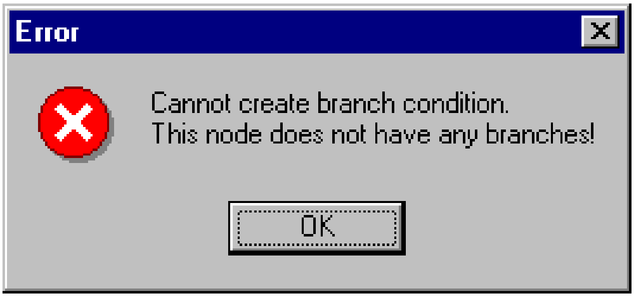


Figure 138. Branch Condition Error Window

— To set branch conditions, click the button for any element other than the default branch condition. You cannot change the attributes of the default branch condition.

Voice@Work displays the Set Test Branch Condition Editor window. For more information about using this editor, see "Set Test Branch Condition Editor" below.

Set Test Expression Editor

The Set Test Expression Editor opens whenever you click the button for an expression already defined in the selected Set and Test node (Figure 139).

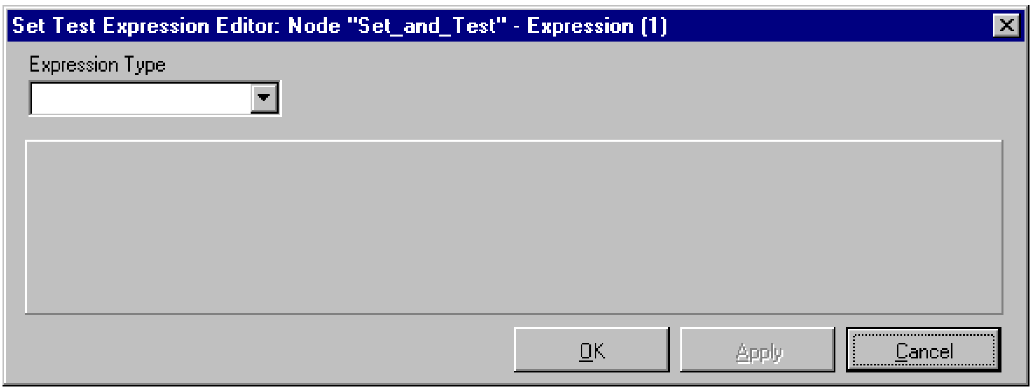


Figure 139. Set Test Expression Editor

From the **Expression Type** pulldown menu, select the type of expression you want to create and set.

NOTE:
You can create new variables in the Set Test Expression Editor in any field that uses variable values. To create a new variable, type the name for the variable enclosed in square brackets [ ] in the appropriate field. When you move to another field, Voice@Work automatically creates the new variable, places it in the Variables Manager, and assigns the variable the appropriate type and default attributes.

The following sections describe the types of test expressions.

Arithmetic

The Arithmetic expression performs arithmetic operations on two selected operands and returns the resulting value (Figure 140).

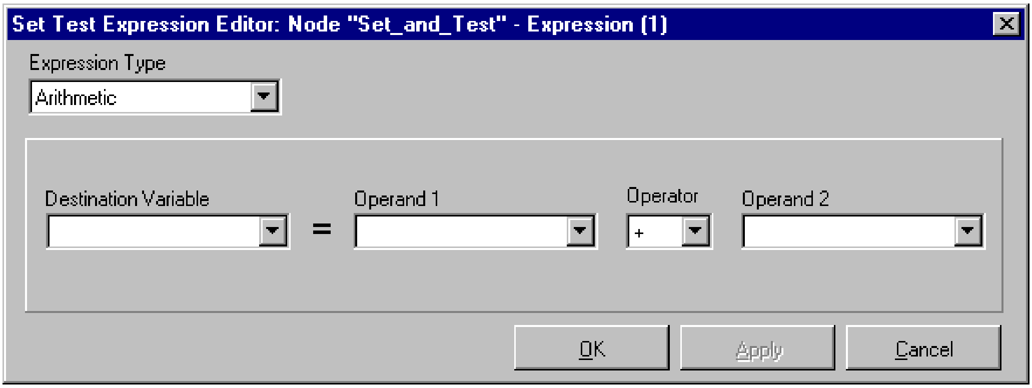


Figure 140. Set Test Expression Editor – Arithmetic Expression Type

Table 87 describes the fields for the **Arithmetic** expression type.

Table 87. Set Test Expression Editor – Arithmetic Type Fields

Field	Action/Description
Destination Variable	Type in or from the pulldown menu select the name of the variable to which you want to assign the resulting value.
Operand 1	Type in a literal value or a variable name, or from the pulldown menu select the variable containing the value for the first operand.
Operator	From the pulldown menu, select the operator to use. Options include: **+**, **-**, *****, **/**, and **mod**.
Operand 2	Type in the literal value, or from the pulldown menu select the variable containing the value for the second operand.

NOTE:
For more information about arithmetic operations with variables, see "Variables and Arithmetic Operations" in Appendix D, "Manipulating Variables".

Assign

The Assign expression assigns the value of a variable or a literal value to another variable (Figure 141).

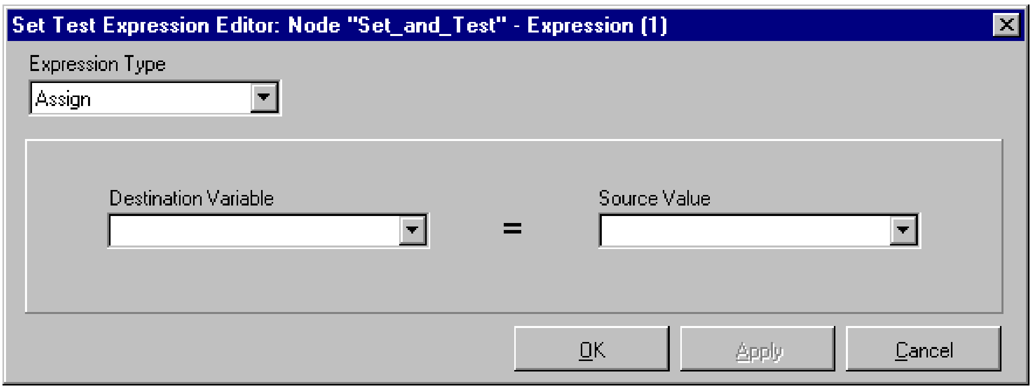


Figure 141. Set Test Expression Editor – Assign Expression Type

Table 88 describes the fields for the **Assign** expression type.

Table 88. Set Test Expression Editor – Assign Type Fields

Field	Action/Description
Destination Variable	Type in or from the pulldown menu select the name of the variable to which you want to assign the resulting value.
Source Value	Type in the literal value, or from the pulldown menu select the variable containing the value, to be assigned to the variable.

Concat

The Concat expression concatenates two character strings into a single string (Figure 142).

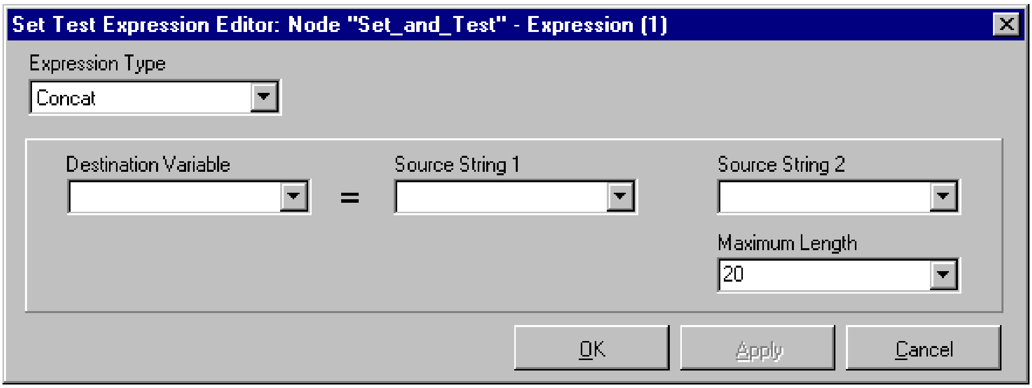


Figure 142. Set Test Expression Editor – Concat Expression Type

Table 89 describes the fields for the **Concat** expression type.

Table 89. Set Test Expression Editor – Concat Type Fields

Field	Action/Description
Destination Variable	Type in or from the pulldown menu select the name of the variable to which you want to assign the resulting value.
Source String 1	Type in the character string, or from the pulldown menu select the variable containing the string to be assigned as the first string.
Source String 2	Type in the character string, or from the pulldown menu select the variable containing the string to be assigned as the second string.
Maximum Length	Type in the number of characters, or from the pulldown menu select the variable containing the number of characters, to control the maximum number of characters permitted in the concatenated string. If this field is set to 0 (zero), no length check is performed. The length of the destination variable determines the maximum length.

ItemCount

The ItemCount expression counts the number of items in a list (a character string with list items separated by commas) and returns the total count as a number (Figure 143).

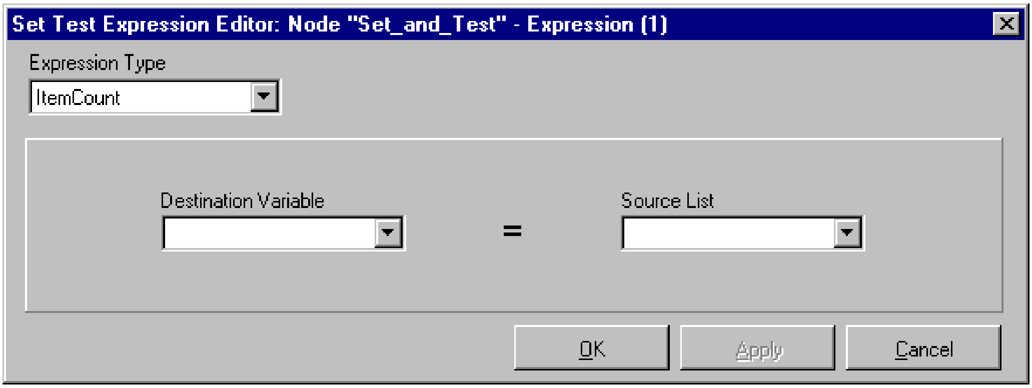


Figure 143. Set Test Expression Editor – ItemCount Expression Type

Table 90 describes the fields for the **ItemCount** expression type.

Table 90. Set Test Expression Editor – ItemCount Type Fields

Field	Action/Description
Destination Variable	Type in or from the pulldown menu select the name of the variable to which you want to assign the resulting value.
Source List	Type in or from the pulldown menu select the variable containing a list of items to be counted.

ItemOf

The ItemOf expression selects a particular item from a list (a character string with list items separated by commas) and returns its value (Figure 144).

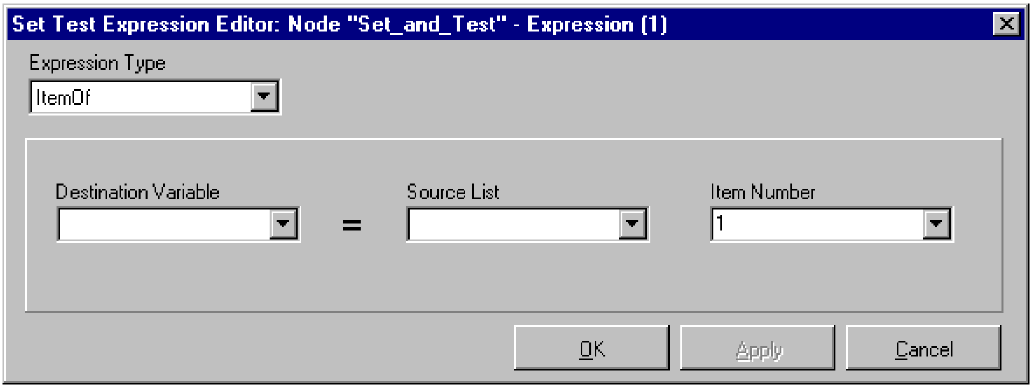


Figure 144. Set Test Expression Editor – ItemOf Expression Type

Table 91 describes the fields for the **ItemOf** expression type.

Table 91. Set Test Expression Editor – ItemOf Type Fields

Field	Action/Description
Destination Variable	Type in or from the pulldown menu select the name of the variable to which you want to assign the resulting value. **NOTE:** If the item number is out of range or otherwise invalid, or if the data source does not exist, a null string is returned.
Source List	Type in or from the pulldown menu select the variable containing a list of items.
Item Number	Type in a literal value or from the pulldown menu select the variable containing the item number of the item you want.

Length

The Length expression counts the number of characters in a string and returns the count as a number (Figure 145).

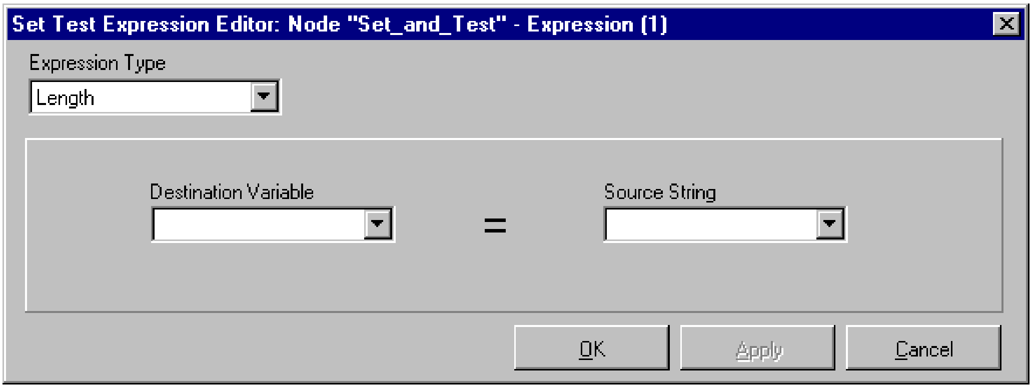


Figure 145. Set Test Expression Editor – Length Expression Type

Table 92 describes the fields for the **Length** expression type.

Table 92. Set Test Expression Editor – Length Type Fields

Field	Action/Description
Destination Variable	Type in or from the pulldown menu select the name of the variable to which you want to assign the resulting value. **NOTE:** This must be a number variable.
Source String	Type in the character string, or from the pulldown menu select the variable containing the string whose characters are to be counted.

NextRecord

The NextRecord expression returns the value of the index for the next record in a previously selected list of records and updates the variables populated by the most recent database query (Figure 146).

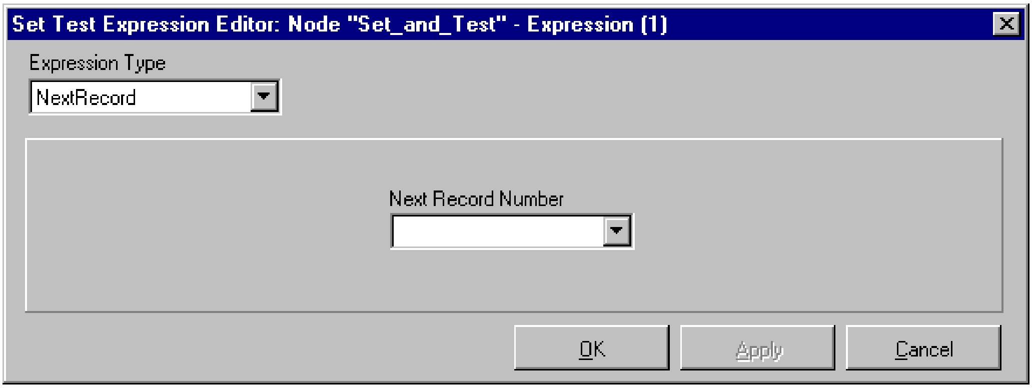


Figure 146. Set Test Expression Editor – NextRecord Expression Type

NOTE:
The return value for this field is incremented before the record is retrieved and represents the number of the current record. In other words, the NextRecord number is actually the current record number after the node is executed.

Table 93 describes the fields for the **NextRecord** expression type.

Table 93. Set Test Expression Editor – NextRecord Type Fields

Field	Action/Description
Next Record Number	Type in or from the pulldown menu select the name of the variable to assign the next record number to. If there is no next record, returns a value of 0 (zero).

Parse

The Parse expression breaks a field or string into two smaller subfields (Figure 147). The source field is typically structured to consist of two or more subfields separated by one or more separator characters such as commas or spaces. When this expression is used in an application, the first subfield of the source string is cut and copied to the destination and any separators are skipped until the next non-separator character is encountered. The remaining characters are then moved to the start of the source field.

NOTE:
This expression alters the original source field or string.

For example, suppose a source field contains the characters “Columbus, Ohio 43213". The separators, as defined in the Parse expression, are commas, periods, and spaces. The first time this expression is used then, the application removes “Columbus” and assigns it to the destination variable. The comma and space are stripped out, and the new source string is “Ohio 43213". The return code contains the number 8, for the number of characters that were reassigned.

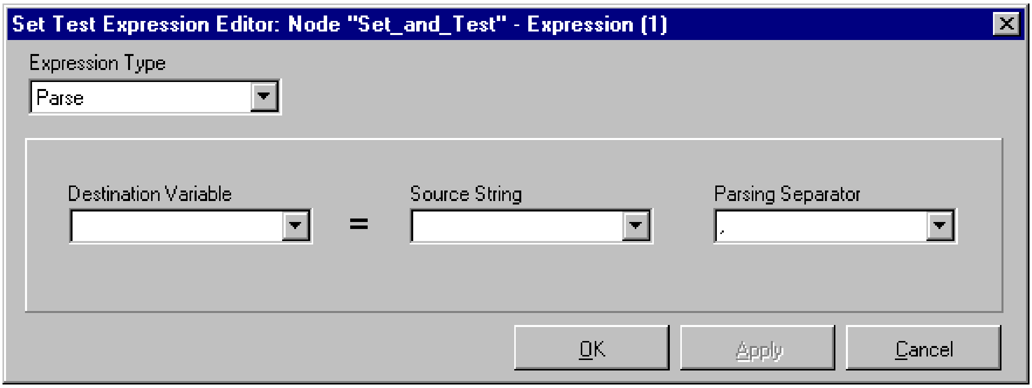


Figure 147. Set Test Expression Editor – Parse Expression Type

Table 94 describes the fields for the **Parse** expression type.

Table 94. Set Test Expression Editor – Parse Type Fields

Field	Action/Description
Destination Variable	Type in or from the pulldown menu select the name of the variable to which you want to assign the resulting value.
Source String	Type in the character string, or from the pulldown menu select the variable containing the string to be parsed.
Parsing Separator	Type in the characters or from the pulldown menu select the variable containing the characters to be used in separating the parsed elements of the string.

PrevRecord

The PrevRecord expression returns the value of the index for the previous record in a previously selected list of records and updates the variables populated by the most recent database query (Figure 148).

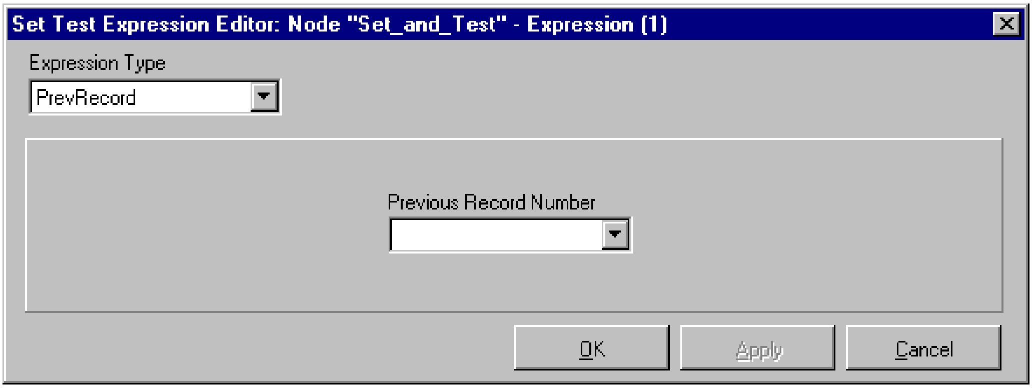


Figure 148. Set Test Expression Editor – PrevRecord Expression Type

NOTE:
The return value for this field is decremented before the record is retrieved and represents the number of the current record. In other words, the PrevRecord number is actually the current record number after the node is executed.

Table 95 describes the fields for the **PrevRecord** expression type.

Table 95. Set Test Expression Editor – PrevRecord Type Fields

Field	Action/Description
Previous Record Number	Type in or from the pulldown menu select the name of the variable to assign the previous record number to. If there is no previous record, returns a value of 0 (zero).

Substring

The Substring expression returns a selected portion of a character string (Figure 149).

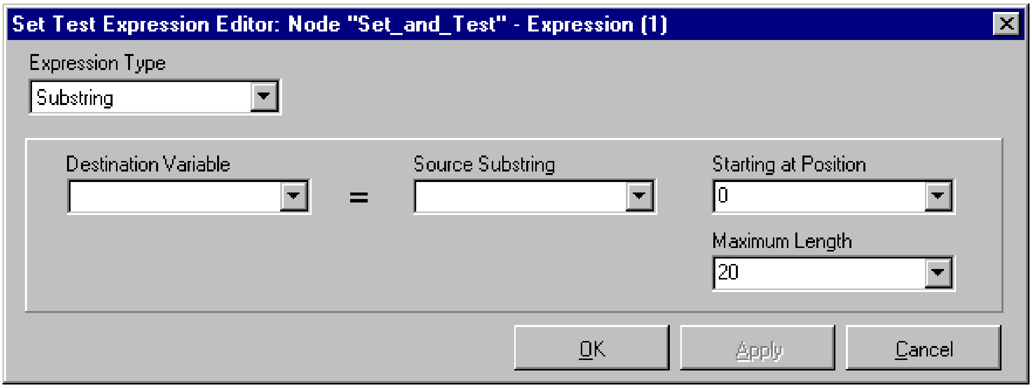


Figure 149. Set Test Expression Editor – Substring Expression Type

Table 96 describes the fields for the Substring expression type.

Table 96. Set Test Expression Editor – Substring Type Fields

Field	Action/Description
Destination Variable	Type in or from the pulldown menu select the name of the variable to which you want to assign the resulting value.

Continued on next page

Table 96. Set Test Expression Editor – Substring Type Fields — *Continued*

Field	Action/Description
Source Substring	Type in the substring, or from the pulldown menu select the variable containing the substring to be returned.
Starting at Position	Type in or from the pulldown menu select the variable containing the character position to use as the starting position for the returned substring. If the source string does not contain enough characters to reach the specified starting position, the return value is null. If the number specified here is a negative number, the resulting substring is defined by counting backwards from the *end* of the source string.
Maximum Length	Type in the number of characters, or from the pulldown menu select the variable containing the number of characters, to use in setting the maximum number of characters to be used from the selected substring. A zero in this field instructs the application to return all characters after the specified starting position. EXAMPLES: Assuming that the source substring value is “Columbus”, then if: ■ “Starting at Position” value = 2, and “Maximum Length” value = 3, the result is “lum”. ■ “Starting at Position” value = 2, and “Maximum Length” value = 8, the result is “lumbus”. ■ “Starting at Position” value = 3, and “Maximum Length” value = 0, the result is “umbus”. ■ “Starting at Position” value = -4, and “Maximum Length” value = 3, the result is “bus”. ■ “Starting at Position” value = -4, and “Maximum Length” value = 0, the result is “mbus”. ■ “Starting at Position” value = 9 or -9, the result is “” (null string).

Set Test Branch Condition Editor

The Set Test Branch Condition Editor opens whenever you click the button for a branch condition already defined in the selected Set and Test node (Figure 150).

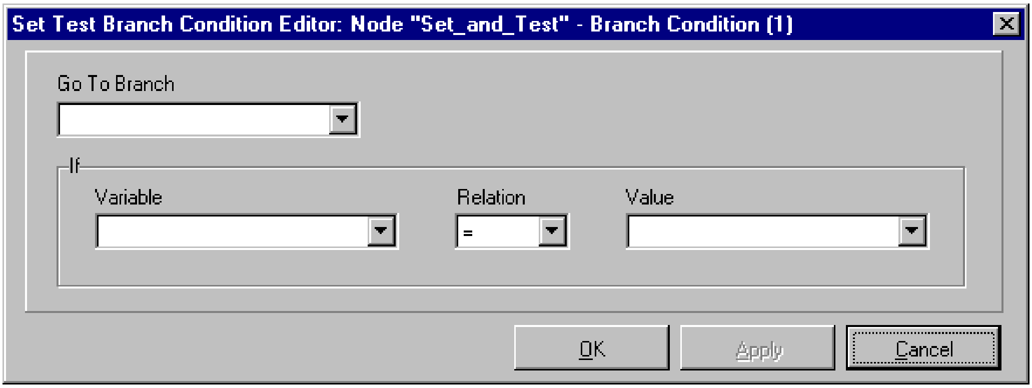


Figure 150. Set Test Branch Condition Editor

Table 97 describes the fields for the Set Test Branch Condition Editor fields.

Table 97. Set Test Branch Condition Editor Fields

Field	Action/Description
Go To Branch	Type in or from the pulldown menu select the name of the branch for which you want to set conditions.
Variable	Type in a variable name or from the pulldown menu select the variable you want to use as the condition test variable.
Relation	From the pulldown menu select the relation operator to use in testing the variable. Options include: **=**, **<>**, **<**, **<=**, **>**, and **>=**.
Value	Type in a literal value or variable name, or from the pulldown menu select the variable containing the value to use in testing the variable.

The Time Branch Node

The Time Branch node is used to direct the call based on time of day, day of the week, or specified date (Figure 151).

Example of Use

ABC Automotive wants to transfer callers to an sales representative during normal business hours only. If the caller calls at any time outside of these hours, an announcement is played asking the caller to call back during normal business hours.

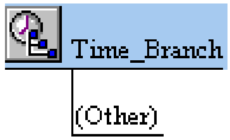


Figure 151. TimeBranch Node

Behavior

- The branch that matches the actual time of day, day of the week, or specified date is taken.
- If there are multiple branches that match the actual time, day or date, the first matching branch is taken.
- Select the branch labeled **(Other)** to insert a branch or branches that have the time conditions you want to set.
- If there are no branches that match the actual time, day, or date, the default **(Other)** branch is taken.

Attributes

When you insert a Time Branch node in a call flow, it creates one branch named (by default) “Other.” This branch cannot be renamed or deleted.

To use a Time Branch node, you must insert additional branches to set the time intervals for which you want the node to check.

Inserting Branches

To insert a branch on a time node, do the following:

1. Right-click the branch where you want the new branch to go (the branch you click moves down when the new branch is created).
2. Click the **Insert Branch** option.

   Voice@Work displays the Time Branch Events dialog box (Figure 152).

By clicking the **Time** tab, you can assign a **Start Time** and an **End Time** for the branch to use in routing calls (Figure 152).

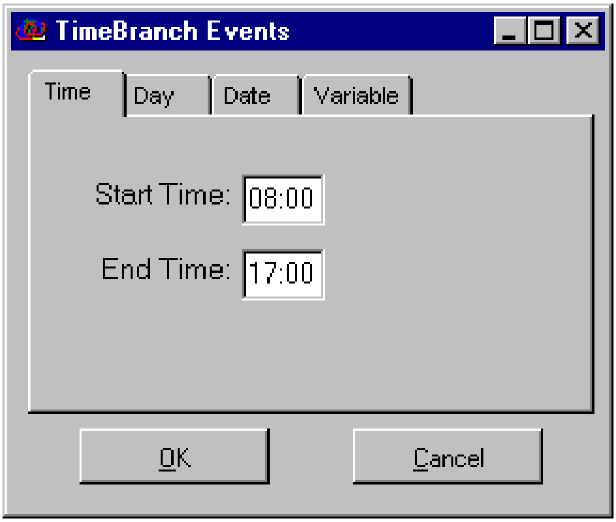


Figure 152. TimeBranch Events Window

By clicking on the **Day** tab, you can assign a **Start Day** and an **End Day** for the branch to use in routing calls (Figure 153).

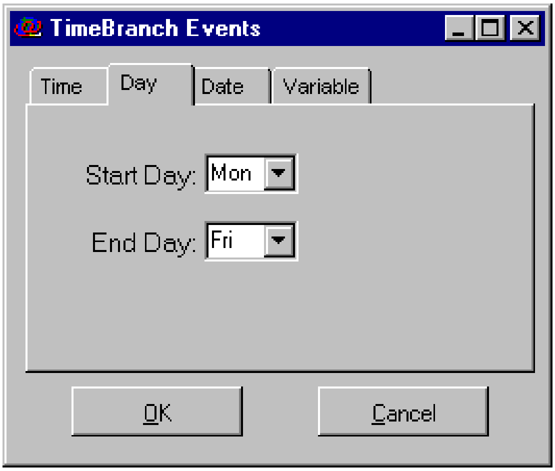


Figure 153. TimeBranch Events — Day Tab

By clicking on the **Date** tab, you can assign a **Start Date** and an **End Date** for the branch to use in routing calls (Figure 154).

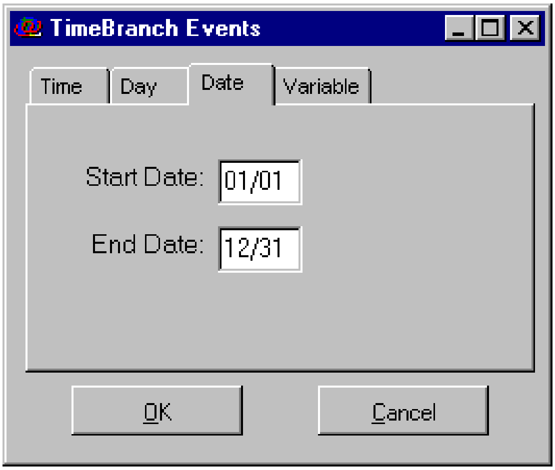


Figure 154. TimeBranch Events — Date Tab

By clicking on the **Variable** tab, you can set your own **Branch Name**, **Start Variable**, and **End Variable** for the branch to use in routing calls (Figure 155).

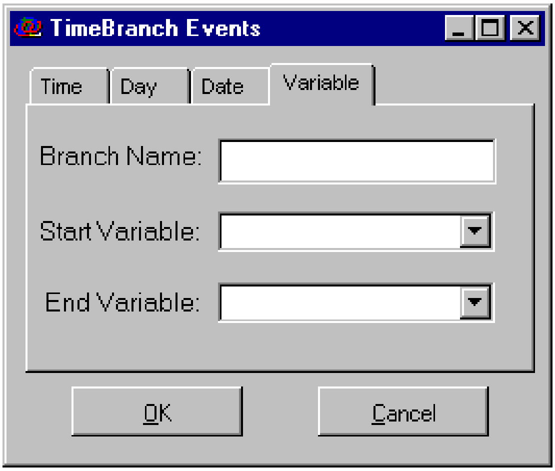


Figure 155. TimeBranch Events — Time Tab

Other Attributes

Other attributes are assigned and edited using the Node Inspector (Figure 156).

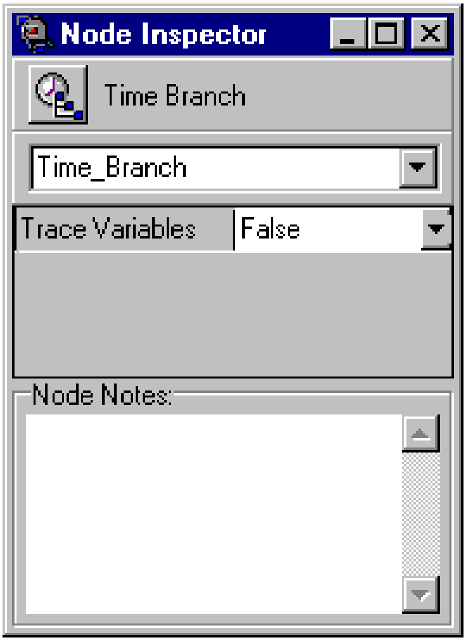


Figure 156. Node Inspector — Time Branch Node

- **Trace Variables** — Embeds a command telling the target system to use the trace function on this node when running the application. This is useful primarily as a debugging tool after installing your application on the target system.

The Asynchronous Event Node

The Asynchronous Event node defines the call flow to be executed if an asynchronous event, such as a caller hanging up, is detected (Figure 157).

Example of Use

A caller has called the ABC Automotive Inquiry Line to get some information but hangs up before completing the call. Once the caller hangs up, the Asynchronous Event node is used to:

- Clear a host screen
- Reset the host to the transaction screen
- Increment a record counter variable
- Terminate the script so it is ready for a new caller

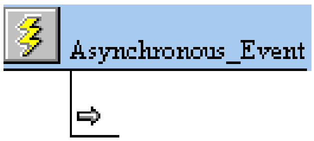


Figure 157. Asynchronous Event Node

Behavior

- The Asynchronous Event node specifies the call flow that is executed when an Asynchronous Event is detected.

  NOTE:
  The only Asynchronous Event currently supported is **Inbound Call Termination**.

- When the system detects an asynchronous event, the following sequence occurs. The system:
  1. Drops the call
  2. Jumps to the defined call flow
  3. Executes the call flow
  4. Terminates application processing of the call

Attributes

Attributes are assigned and edited using the Node Inspector (Figure 158).

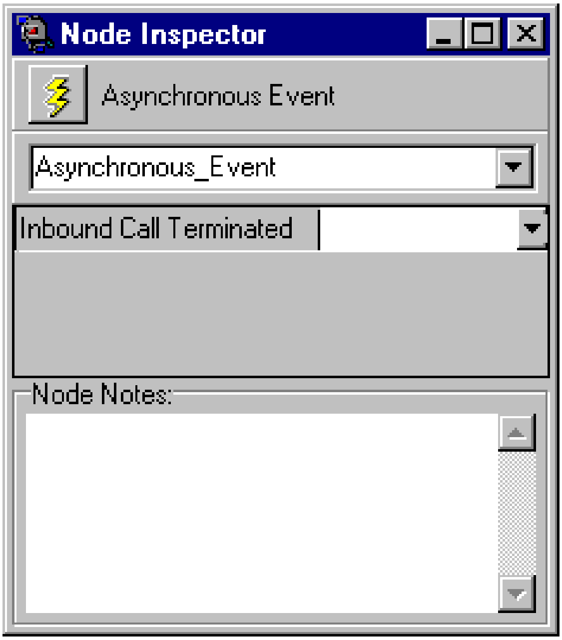


Figure 158. Node Inspector — Asynchronous Event

- **Inbound Call Terminated** — From the pulldown menu, select the name of the call flow to be executed, or the **[none]** option.

> **NOTE:**
> Using the example above, once the caller has reached the point in the script where a hangup might be expected, then you would use the [none] option to negate a previously set asynchronous event routine.

Telephony Nodes

The following sections describe the telephony nodes.

The Answer Call Node

The Answer Call node answers the incoming call. This is typically one of the first nodes of any application.

Example of Use

A customer calls the ABC Automotive Inquiry Line. The system detects the incoming call and answers it.

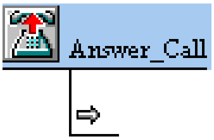


Figure 159. Answer Call Node

Behavior

- The Answer Call node answers incoming calls and passes them along to the next node.
- The application will work in Simulation mode without an Answer Call node. When the application is downloaded to the target system, however, no incoming calls are answered without the code that is generated from the Answer Call node.

NOTE:
If you have not used an Answer Call node in your application, the Verify Design tool, when used, will notify you that it is missing. See Chapter 29, "The Verify Design Tool", for additional information.

Attributes

Attributes are assigned and edited using the Node Inspector (Figure 160). There are no user-configurable attributes for the Answer Call node.

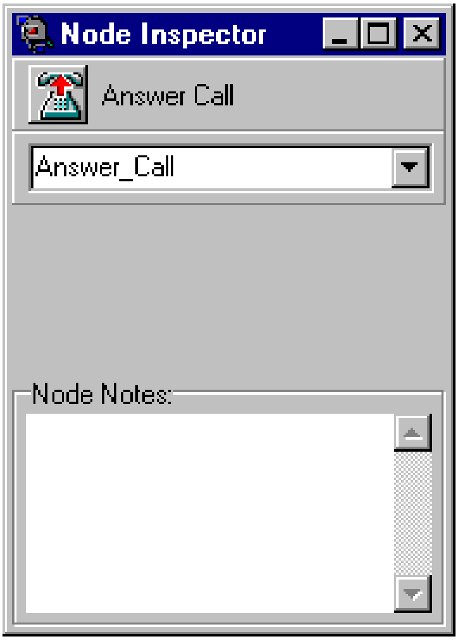


Figure 160. Node Inspector — Answer Call Node

The Transfer Call Node

The Transfer Call node is used to transfer a call to another number (Figure 161).

Example of Use

The ABC Automotive Inquiry Line has a menu giving the caller the option to transfer to the service department or the parts department. When the caller makes a choice by entering the proper touchtone, the call is transferred to the appropriate department.

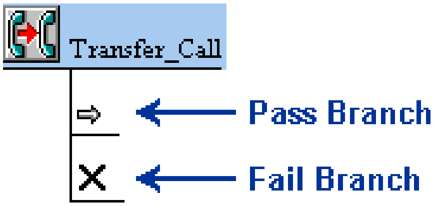


Figure 161. Transfer Call Node

Behavior

- When this node is executed, the call is transferred to the selected telephone number.
- If the transfer is completed successfully, the Pass branch is taken.
- If the transfer is not completed within the time specified, the Fail branch is taken.
- The node can be set to play a message to the caller while the transfer is being executed, depending upon the type of transfer you have selected.
- Specific node behavior for this node depends on the type of transfer you have selected. There are three types of transfers: Blind (default), Full CCA, and Intelligent (see **Transfer Type** below).

Attributes

Attributes are assigned and edited using the Node Inspector (Figure 162).

The Transfer Options Tab

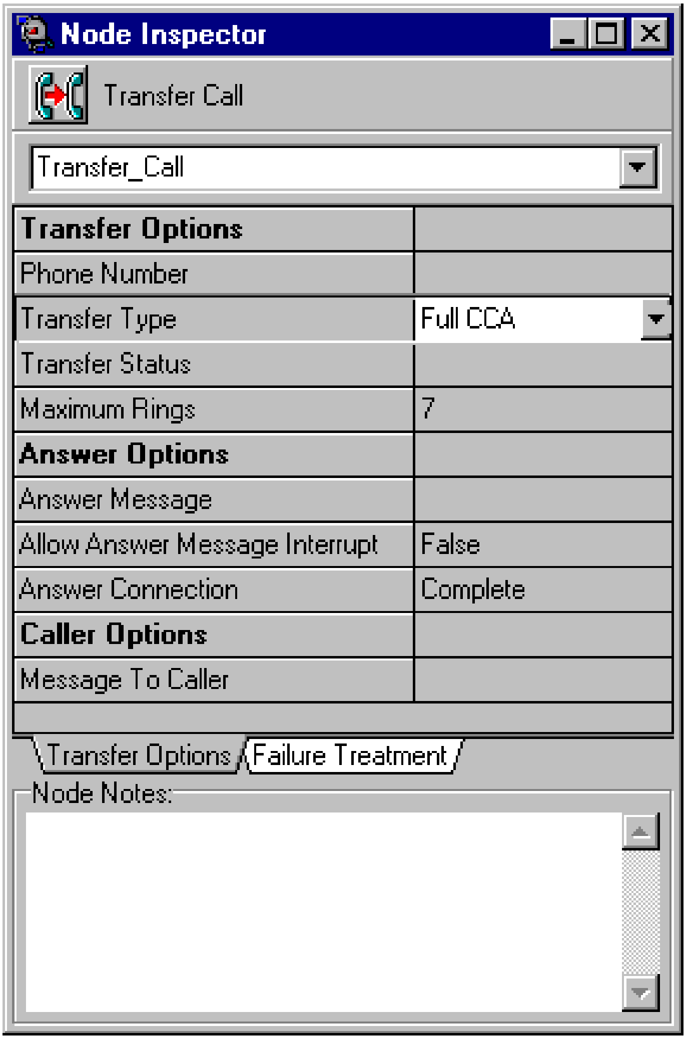


Figure 162. Transfer Call Node — Transfer Options Tab

Transfer Options

- **Phone Number** — Enter the telephone number or from the pulldown menu select the variable containing the telephone number to transfer the caller to.

  NOTE:
  You can use alphanumeric characters in entering this telephone number.

- **Transfer Type** — From the pulldown menu select the type of transfer to make. There are three options:
  - — **Blind** (Default): The application dials the third-party number to start the transfer, and then relinquishes all call-handling responsibilities. In other words, in a blind transfer, the transfer call is placed, and the caller is released. The caller is then left to deal with a busy signal or a no-answer.
  - — **Intelligent** (CCA): The application dials the third-party number to start the transfer. It then monitors call progress signals to determine if the line is busy, is ringing, or has been answered. In this case, the application continues to monitor the transfer until it is successfully completed. If the call cannot be completed successfully, control of the call returns to the application and the Fail branch of the node is taken.
  - — **Full CCA** (Call Classification Analysis) This is a more advanced form of transfer type (see above). Full CCA provides more complete interpretation of network progress tones than does basic intelligent CCA. It includes busy, answered, ring no-answer, and reorder detections.

  NOTE:
  For more information about call transfer types, see the *INTUITY™ CONVERSANT® System Description* documentation for your system.

- **Transfer Status** — From the pulldown menu select the variable, if any, you want to use to store the transfer status result.

  NOTE:
  For more information about transfer status, see the *INTUITY™ CONVERSANT® System Description* documentation for your system.

- **Maximum Rings** — (Inactive when Transfer Type **Blind** is selected) Enter a number, or from the pulldown menu select the variable that specifies the maximum number of rings to allow before directing the call to the Fail branch.
  (Range = 0-20, default = 7)

Answer Options

NOTE:
These options are inactive when Transfer Type Blind is selected.

- **Answer Message** — From the pulldown menu select the prompt to play *to the third party* when the message is answered.

- **Allow Answer Message Interrupt** — When set to **True**, this attribute allows the third party to interrupt the Answer Message and connect with the caller more quickly.
- **Answer Connection** — From the pulldown menu select the option that determines how the call is handled after an answer is detected. There are two options:
  - **Complete** — (Default) Connects the caller to a third party when a specific condition exists, such as third-party Answer, third-party NoAnswer, or third-party Busy
  - **Hold** — Leaves the caller on hold when a specific condition exists, such as third-party Answer, third-party NoAnswer, or third-party Busy

Caller Options

- **Message to Caller** — From the pulldown menu, select the prompt to play to the caller prior to transferring the call.

The Failure Treatment Tab

Figure 163 shows the Failure Treatment tab.

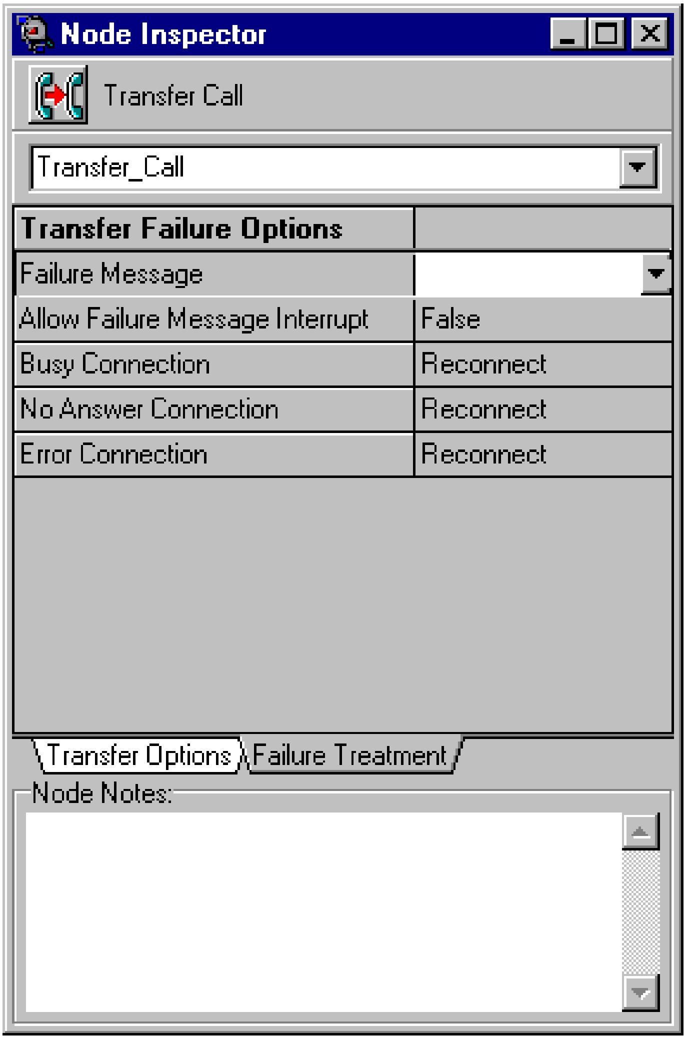


Figure 163. Transfer Call Node — Failure Treatment Tab

Transfer Failure Options

NOTE:
These options are inactive when Transfer Type **Blind** is selected.

- **Failure Message** — From the pulldown menu, select the prompt or message to play if the call cannot be completed successfully.

- **Allow Failure Message Interrupt** — When set to **True**, this attribute allows the caller to interrupt the Failure Message by hanging up or selecting another option.
- **Busy Connection** — From the pulldown menu, select the option that determines how the call is handled when a busy signal is detected. There are three options:
  - **Reconnect** — (Default) Drops the third party and reconnects the caller to the script so that other action can be taken, when a specific condition exists, such as third-party Answer, third-party NoAnswer, or third-party Busy
  - **Complete** — Connects the caller to a third party when a specific condition exists, such as third-party Answer, third-party NoAnswer, or third-party Busy
  - **Hold** — Leaves the caller on hold when a specific condition exists, such as third-party Answer, third-party NoAnswer, or third-party Busy
- **No Answer Connection** — From the pulldown menu select the option that determines how the call is handled when no answer is detected. There are three options:
  - **Reconnect** — (Default) Drops the third party and reconnects the caller to the script so that other action can be taken, when a specific condition exists, such as third-party Answer, third-party NoAnswer, or third-party Busy
  - **Complete** — Connects the caller to a third party when a specific condition exists, such as third-party Answer, third-party NoAnswer, or third-party Busy
  - **Hold** — Leaves the caller on hold when a specific condition exists, such as third-party Answer, third-party NoAnswer, or third-party Busy
- **Error Connection** — From the pulldown menu select the option that determines how the call is handled when an erroneous connection is detected. There are three options:
  - **Reconnect** — (Default) Drops the third party and reconnects the caller to the script so that other action can be taken, when a specific condition exists, such as third-party Answer, third-party NoAnswer, or third-party Busy
  - **Complete** — Connects the caller to a third party when a specific condition exists, such as third-party Answer, third-party NoAnswer, or third-party Busy
  - **Hold** — Leaves the caller on hold when a specific condition exists, such as third-party Answer, third-party NoAnswer, or third-party Busy

The Disconnect Call Node

The Disconnect Call node disconnects the incoming call and executes the next node in the call flow.

Example of Use

A customer calling ABC Automotive gets the desired information and hangs up. The system disconnects and resets all the variables to be ready for the next caller (Figure 164).

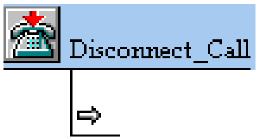


Figure 164. Disconnect Call Node

Behavior

The Disconnect node:

- Disconnects incoming calls and executes the next node
- Allows the call flow to be terminated at specific user-defined points
- Is *not* required to successfully terminate the dialog with the caller

Attributes

Attributes are assigned and edited using the Node Inspector (Figure 165).There are no user-configurable attributes for the Disconnect Call node.

Figure 165. Node Inspector — Disconnect Call Node

IVR (Interactive Voice Response) Nodes

This section describes the IVR nodes.

The Announcement Node

The Announcement node plays a simple announcement to the caller (Figure 166).

Example of Use

ABC Automotive has a customer hotline. The caller who reaches the hotline is greeted with the following announcement:

> "Thank you for calling the ABC Automotive Customer Service hotline."

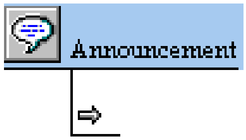


Figure 166. Announcement Node

Behavior

- When executed, a simple announcement is played to the caller.
- No data is collected from the caller.

Attributes

Attributes are assigned and edited using the Node Inspector (Figure 167).

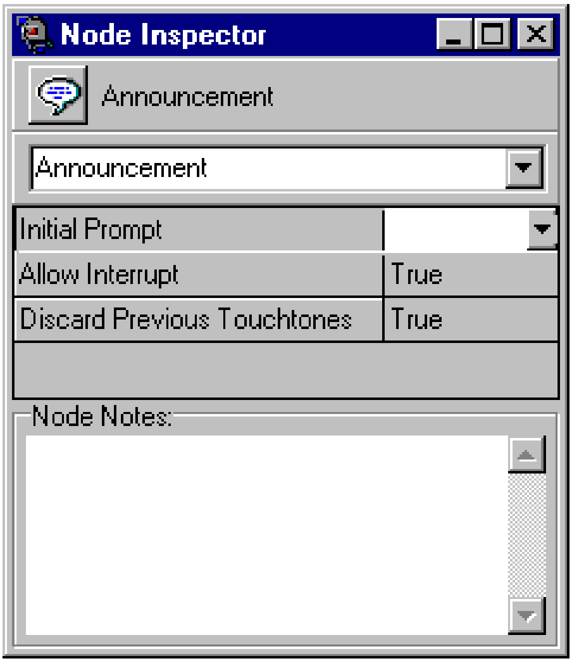


Figure 167. Node Inspector — Announcement Node

- **Initial Prompt** — From the pulldown menu, select the prompt you want to use as the announcement, or double-click to create a new one. See Chapter 19, "Working with Prompts", for additional information.
- **Allow Interrupt** — To allow the caller to interrupt the Initial Prompt, select **True** from the pulldown menu.

  To play the Initial Prompt without interruption, select **False** from the pulldown menu.
- **Discard Previous Touchtone** — To ignore any touchtones the caller may have entered before the Initial Prompt is played, select **True** from the pulldown menu.

  To process any touchtones the caller may have entered before the Initial Prompt is played, select **False** from the pulldown menu. This allows you to take advantage of the "Type Ahead" feature, which works only with touchtones.

The Menu Node

The Menu node presents a list of options from which the caller can choose (Figure 168).

Example of Use

The ABC Automotive Inquiry Line gives the caller a menu of options to choose from:

> "For the Service Department, press 1.
>
> For the Parts Department, press 2.
>
> To find out the status of an order, press 3."

A valid response from the caller would be a touchtone of 1, 2, or 3.

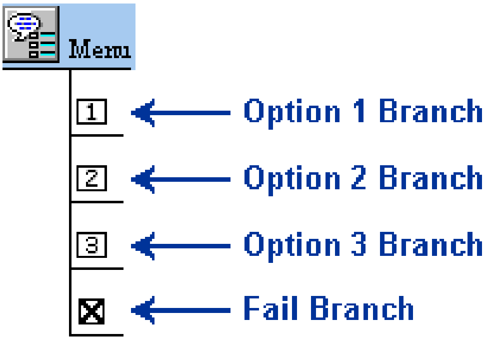


Figure 168. Menu Node

Behavior

- This node plays the Initial Prompt as specified in the Node Inspector (see Prompts tab below).
- The Initial Prompt presents a predetermined menu of choices to the caller.
- This node directs the call based upon the caller's response to the menu choices.
- If the caller enters an invalid response, the Bad Input Prompt is played. If the **Replay Initial Prompt** field is set to **True**, the Initial Prompt also plays after the Bad Input Prompt.
- If the caller does not respond with enough digits, the Too Few Digits Prompt is played. If the **Replay Initial Prompt** field is set to **True**, the Initial Prompt also plays after the Too Few Digits Prompt.
- If the caller enters an invalid response and the **Tries Available** field value has been met, the Fail branch is executed.
- If the caller does not respond within the time set in the **First Digit Timeout** field, the Timeout Prompt is played. If the **Replay Initial Prompt** field is set to **True**, the Initial Prompt also plays.
- If the caller does not respond on the last try, the Timeout branch is taken if one has been defined. Otherwise, the Fail branch is taken by default.
- Valid responses from the caller are saved in the appropriate variable.

Attributes

When you insert a Menu node in a call flow, Voice@Work creates four branches and assigns each branch a default identifying number. You can insert more branches, rename branches, or delete branches as you need to.

Inserting Branches

To insert a new branch on a Menu node, do the following:

1. Right-click the branch where you want the new branch to be inserted.

   NOTE:
   When the new branch is inserted, it moves the selected branch down.

   Voice@Work displays the Touchtones keypad window (Figure 169).

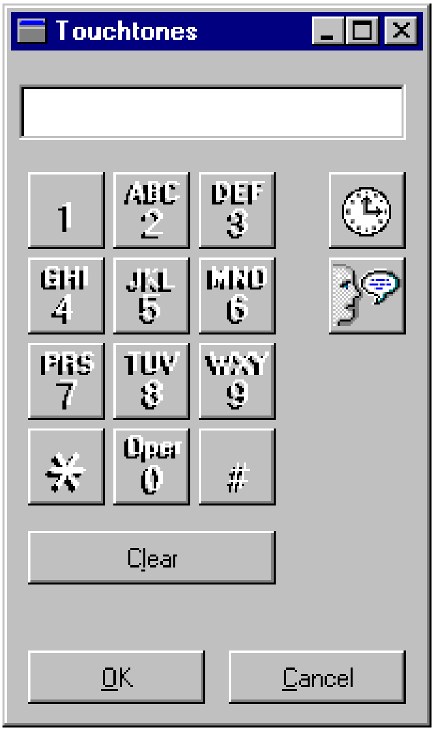
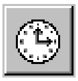
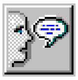


Figure 169. Touchtones Keypad Window

2. Do one of the following:
   - Click the digit (or digits) you want to assign to label the branch.
   - To define the branch as a "timeout" branch, click the  icon.

- To define the branch using an WholeWord grammar enumerated list item, click the  icon.

  Clicking this icon opens a Grammar Word List window. This window displays the list items enumerated for the WholeWord grammar designated on the Response tab for this node.

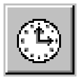
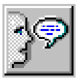
3. Click **OK**.

Renaming Branches

To rename (assign a different identifier to) a branch on a Menu node, do the following:

1. Right-click the branch you want to rename, then select **Rename Branch** from the popup menu.

   Voice@Work displays the Touchtones keypad window (Figure 169).

2. Do one of the following:
   - Click the digit or digits you want to assign to label the branch.
   - To rename the branch as a “timeout” branch, click the  icon.
   - To rename the branch using an WholeWord grammar enumerated list item, click the  icon.

     Clicking this icon opens a Grammar Word List window. This window displays the list items enumerated for the WholeWord grammar designated on the Response tab for this node.

3. Click **OK**.

Deleting Branches

To delete a branch from a Menu node, do the following:

1. Right-click the branch you want to delete.
2. Select Delete from the popup menu.

Other Attributes

Other attributes are assigned and edited using the Node Inspector (Figure 170).

The Prompts Tab

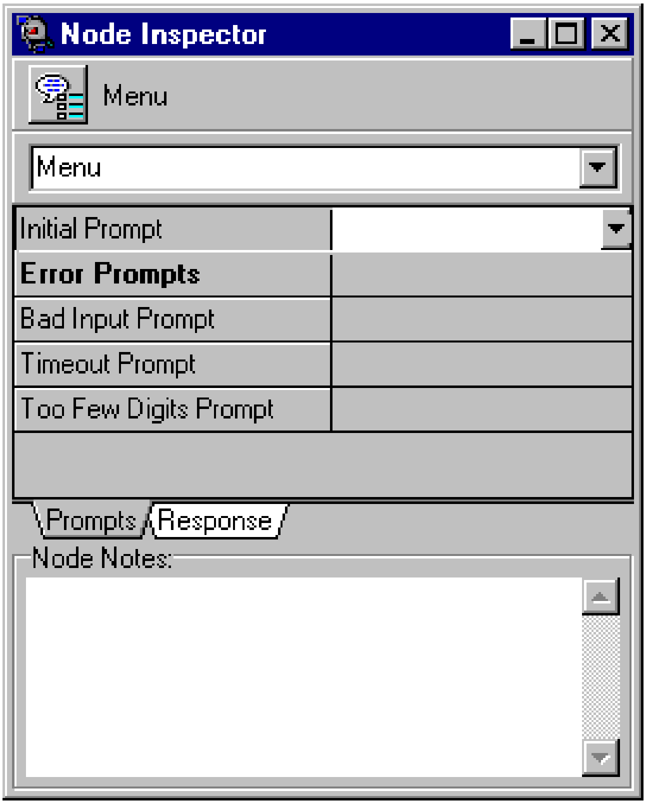


Figure 170. Menu Node — Prompts Tab

- **Initial Prompt** — From the pulldown menu, select the prompt you want to play when the Menu node begins to execute. The Initial Prompt must be defined using the Prompt Editor window. See "The Prompt Editor" in Chapter 19, "Working with Prompts", for additional information.

Error Prompts

- **Bad Input Prompt** — From the pulldown menu, select the prompt to play when the caller enters invalid input, such as pressing a touchtone which is not defined as one of the choices or saying the wrong word in response.
- **Timeout Prompt** — From the pulldown menu, select the prompt to play when the caller waits too long to respond to the Initial Prompt.
- **Too Few Digits Prompt** — From the pulldown menu, select the prompt to play when the caller does not enter enough digits.

The Response Tab

Figure 171 shows the Response tab.

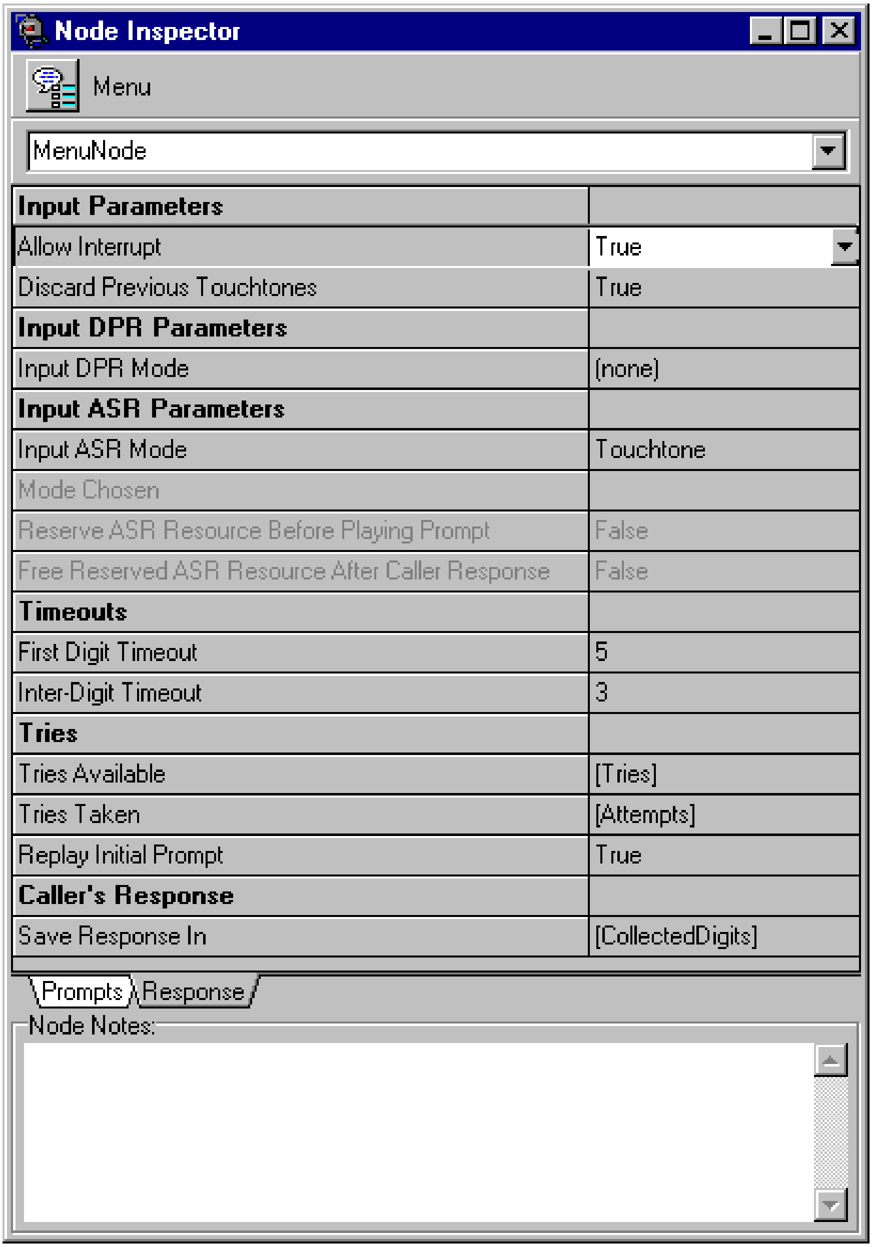


Figure 171. Menu Node — Response Tab

Input Parameters

- **Allow Interrupt** — To allow the caller to interrupt while prompts are playing, select **True** from the pulldown menu.

  To play prompts without interruption, select **False** from the pulldown menu.

  NOTE:
  You can use this option only if the **Input ASR Mode** option is set to Touchtones or a WholeWord speech recognition grammar. Even then, you can use a WholeWord speech recognition grammar only if you are using an SP or SSP circuit card. For details, see the *INTUITY™ CONVERSANT® System Speech Development, Processing, and Recognition* documentation for your system.

- **Discard Previous Touchtone** — To ignore any touchtones the caller may have entered before the Initial Prompt is played, select **True** from the pulldown menu.

  To process any touchtones the caller may have entered before the Initial Prompt is played, select **False** from the pulldown menu. This allows you to take advantage of the “Type Ahead” feature, which works only with touchtones.

Input DPR Parameters

- **Input DPR Mode** – From the pulldown menu, select the DPR grammar you want to use to collect input from the caller.

Input ASR Parameters

- **Input Mode** — From the pulldown menu, select whether you want to use Touchtones (default), WholeWord speech recognition, or FlexWord speech recognition as input from the caller.
- **Mode Chosen** — (Not available if Touchtone is selected as the Input Mode) From the pulldown menu, select the variable that holds the value indicating how the caller responded to the Initial Prompt.

  NOTE:
  If WholeWord or FlexWord speech recognition is chosen here, and the caller enters a response using Touchtones, the value returned to the variable is a negative number. Otherwise, the value returned is a positive number.

- **Reserve ASR Resource Before Playing Prompt** — (Not available if Touchtone is selected as the Input Mode) To make sure that the ASR resource is available before obtaining spoken caller input, select True.
- **Free Reserved ASR Resource After Caller Response** — (Not available if Touchtone is selected as the Input Mode) To free the ASR resource as soon as spoken input is retrieved from the caller, select **True**.

Timeouts

- **First Digit Timeout** — Do one of the following:
  - — Enter the number of seconds to allow a caller to respond to the Initial Prompt.
  - — From the pulldown menu, select a variable to determine the number of seconds to allow a caller to respond to the Initial Prompt.
- **Inter-Digit Timeout** — Do one of the following:
  - — Enter the number of seconds to allow a caller for entering each subsequent digit in a multiple-digit response.
  - — From the pulldown menu, select a variable to determine the number of seconds to allow a caller for entering each subsequent digit in a multiple-digit response.

Tries

- **Tries Available** — Do one of the following:
  - — Enter the number of tries to allow a caller to respond to the Initial Prompt.
  - — From the pulldown menu, select a variable to determine the number of tries to allow a caller to respond to the Initial Prompt.
    (Default = 3)
- **Tries Taken** — From the pulldown menu, select a variable to hold the number of tries it takes a caller to enter a correct response.
- **Replay Initial Prompt** — If you want the Initial Prompt to replay each time the caller enters an invalid response (after any error messages when appropriate), select **True**.

Caller’s Response

- **Save Response In** — From the pulldown menu, select the variable in which to save the caller's unformatted response. This must be a character variable.

The Prompt and Collect Node

The Prompt and Collect node is used to prompt a caller for input such as account numbers or credit card numbers (Figure 172).

Example of Use

A customer calls the ABC Automotive Inquiry Line to find out about the status of an order. The caller is prompted for the order number:

> “The order number is located in the upper left corner of your order form, just underneath your name and address. Please enter your eight-digit order number, followed by the pound sign.”

After the caller enters the requested order number and pound sign, the following prompt plays:

> “You entered your order number as: 3159 8433 Is this correct? If so, say 'Yes' at the tone.”

When the caller responds “Yes,” the application validates the number against a database table field (Order Numbers) and moves on to the next prompt.

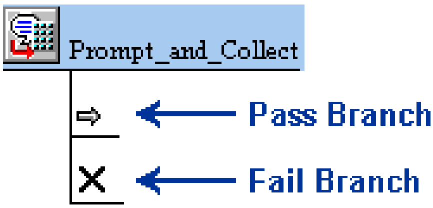


Figure 172. Prompt and Collect Node

Behavior

- The Initial Prompt plays, requesting the caller to enter input.
- The caller's response is collected for future use.
- The caller may be asked to confirm the response entered.
- The caller's input can be validated against a variable, set, or table value.

Attributes

When you insert an Prompt and Collect node, Voice@Work creates two default branches which cannot be renamed or deleted. You can, however, insert additional branches which can later be renamed.

Inserting Branches

To insert a new branch on an Prompt and Collect node, do the following:

1. To insert an additional branch, right-click where you want the new branch.
2. Select **Insert Branch** from the popup menu.

   Voice@Work displays the Touchtones dialog box (Figure 173).

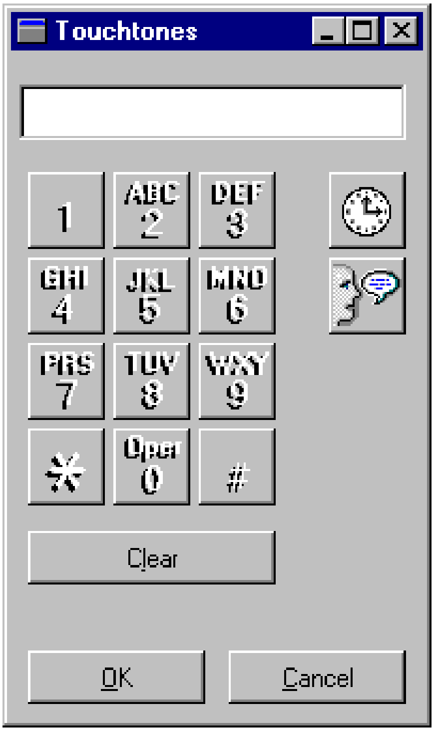
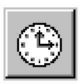
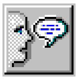


Figure 173. Touchtones Keypad Window

3. Do one of the following:
   - Enter the digit (or digits) you want to assign to the branch.
   - To define the branch as a “timeout” branch, click the  icon.

- To define the branch using an WholeWord grammar enumerated list item, click the  icon.

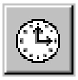
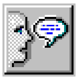
  Clicking this icon opens a Grammar Word List window. This window displays the list items enumerated for the WholeWord grammar designated on the Response tab for this node.

4. Click **OK**.

Renaming Branches

To rename (assign a different number to) a branch, do the following:

1. Right-click the branch you want to rename.
2. Select **Rename Branch** from the popup menu.

   Voice@Work displays the Touchtones dialog box (Figure 173).
3. Do one of the following:
   - Enter the digit (or digits) you want to assign to the branch.
   - To rename the branch as a “timeout” branch, click the  icon.
   - To rename the branch using an WholeWord grammar enumerated list item, click the  icon.

     Clicking this icon opens a Grammar Word List window. This window displays the list items enumerated for the WholeWord grammar designated on the Response tab for this node.
4. Click **OK**.

Other Attributes

Other attributes are assigned and edited using the Node Inspector (Figure 174).

The Prompts Tab

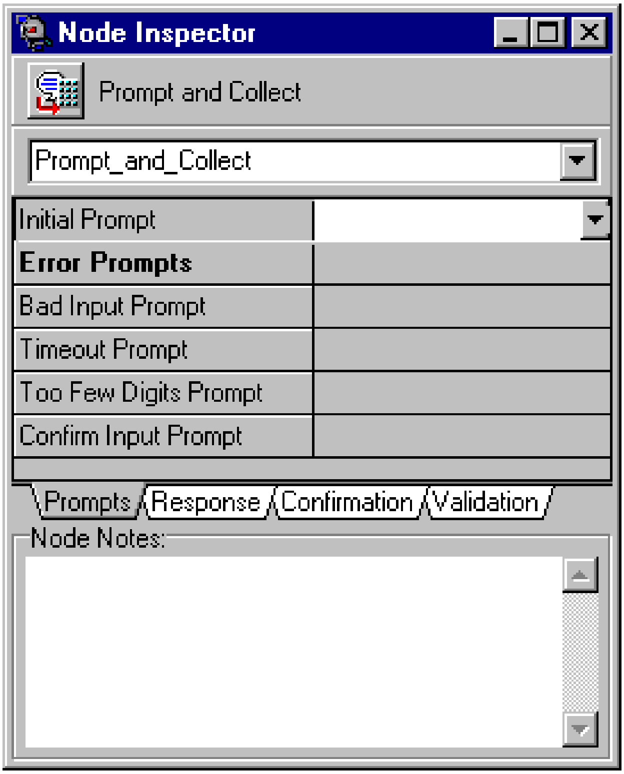


Figure 174. Prompt and Collect Node — Prompts Tab

- **Initial Prompt** — From the pulldown menu, select the name of the prompt to play when execution of the node is started.

Error Prompts

- **Bad Input Prompt** — From the pulldown menu, select the prompt to play when the caller enters invalid input, such as entering an invalid account number.
- **Timeout Prompt** — From the pulldown menu, select the prompt to play when the caller waits too long to respond to the Initial Prompt.
- **Too Few Digits Prompt** — From the pulldown menu, select the prompt to play when the caller does not enter enough digits.
- **Confirm Input Prompt** — From the pulldown menu, select the prompt to play after the caller enters a response. This prompt is used to get feedback as to whether the caller's initial response was correct.

The Response Tab

Figure 175 show the Response tab.

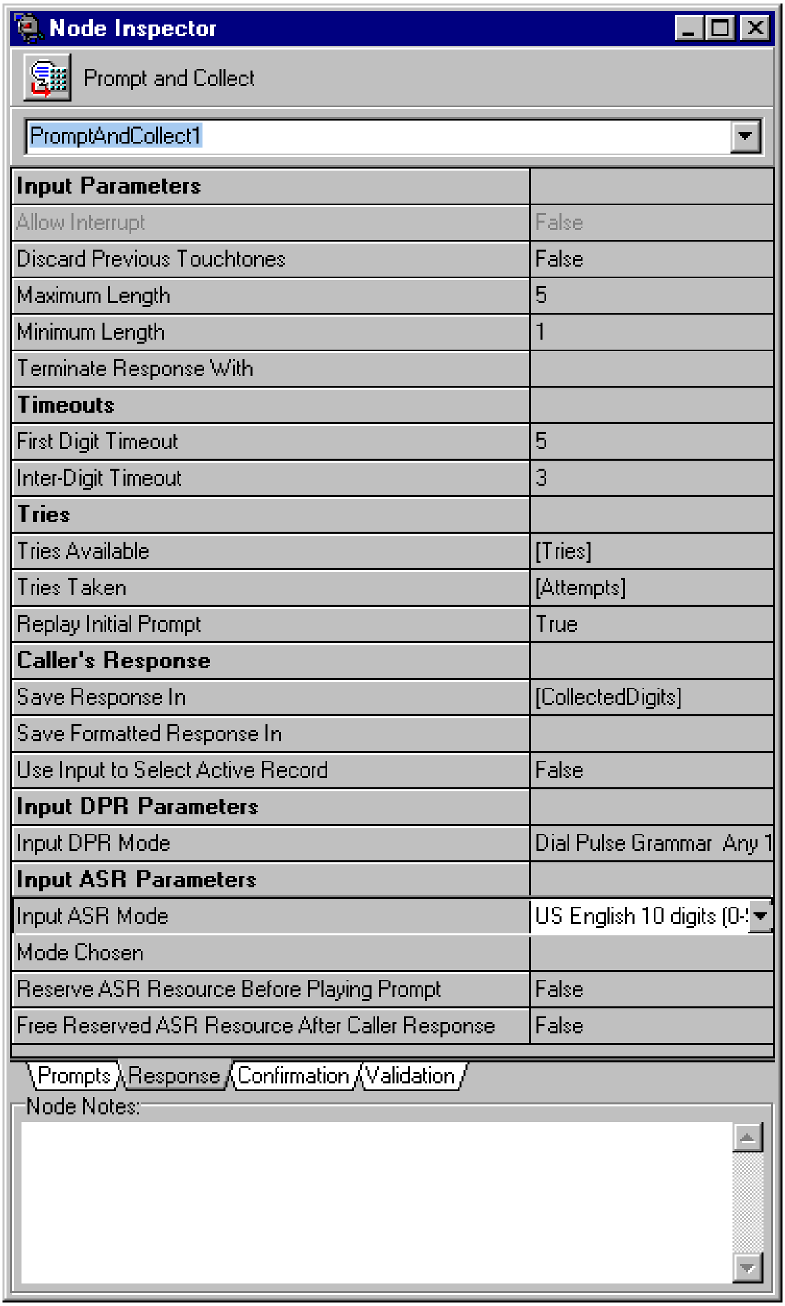


Figure 175. Prompt and Collect Node — Response Tab

Input Parameters

- **Allow Interrupt** — To allow the caller to interrupt while prompts are playing, select **True** from the pulldown menu.

  To play prompts without interruption, select **False** from the pulldown menu.

  NOTE:
  You can use this option only if the **Input ASR Mode** option is set to Touchtones or a WholeWord speech recognition grammar. Even then, you can use a WholeWord speech recognition grammar only if you are using an SP or SSP circuit card. For details, see the *INTUITY™ CONVERSANT® System Speech Development, Processing, and Recognition* documentation for your system.

- **Discard Previous Touchtone** — To ignore any touchtones the caller may have entered before the Initial Prompt is played, select **True** from the pulldown menu.

  To process any touchtones the caller may have entered before the Initial Prompt is played, select **False** from the pulldown menu. This allows you to take advantage of the "Type Ahead" feature, which works only with touchtones.

- **Maximum Length** — Enter the maximum number of digits to accept from the caller. This may be a variable from the pulldown menu.
- **Minimum Length** — Enter the minimum number of digits to accept from the caller. This may be a variable from the pulldown menu.
- **Terminate Response With** — Enter the characters to be entered by the caller to indicate the end of a response. This may be a variable from the pulldown menu.

Timeouts

- **First Digit Timeout** — Do one of the following:
  - Enter the number of seconds to allow a caller to respond to the Initial Prompt.
  - From the pulldown menu, select a variable to determine the number of seconds to allow a caller to respond to the Initial Prompt.
- **Inter-Digit Timeout** — Do one of the following:
  - Enter the number of seconds to allow a caller for entering each subsequent digit in a multiple-digit response.
  - From the pulldown menu, select a variable to determine the number of seconds to allow a caller for entering each subsequent digit in a multiple-digit response.

Tries

- **Tries Available** — Do one of the following:
  - — Enter the number of tries to allow a caller to respond to the Initial Prompt.
  - — From the pulldown menu, select a variable to determine the number of tries to allow a caller to respond to the Initial Prompt.
    (Default = 3)
- **Tries Taken** — From the pulldown menu, select a variable to hold the number of tries it takes a caller to enter a correct response.
- **Replay Initial Prompt** — If you want to replay the Initial Prompt, after any error messages, each time the caller enters an invalid response, select **True**.

Caller's Response

- **Save Response In** — From the pulldown menu, select the variable in which to save the caller's unformatted response.
- **Save Formatted Response In** — From the pulldown menu, select the variable in which to save the caller's formatted response.
- **Use Input to Select Active Record** — When set to **True**, this causes the selected table record's values to be stored into the variables specified in the previous Query Table node.

Input DPR Parameters

- **Input DPR Mode** – From the pulldown menu, select the DPR grammar you want to use to collect input from the caller.

Input ASR Parameters

- **Input Mode** — From the pulldown menu, select whether you want to use Touchtones, a WholeWord speech recognition, or a FlexWord speech recognition to collect input from the caller.
- **Mode Chosen** — (Not available if Touchtone is selected as the Input Mode) From the pulldown menu, select the variable that holds the value indicating how the caller responded to the Initial Prompt.

- **Reserve ASR Resource Before Playing Prompt** — (Not available if Touchtone is selected as the Input Mode) To make sure that the ASR resource is available before obtaining spoken caller input, select **True**.

  NOTE:
  To reserve ASR resources, you must specify a node for the **Fail** branch. If you do not, you can set this value to **True**, but no code is generated to allocate ASR resources during the code generation process.

- **Free Reserved ASR Resource After Caller Response** – (Not available if Touchtone is selected as the Input Mode) To free the ASR resource as soon as spoken input is retrieved from the caller, select **True**.

The Confirmation Tab

Figure 176 shows the Confirmation tab.

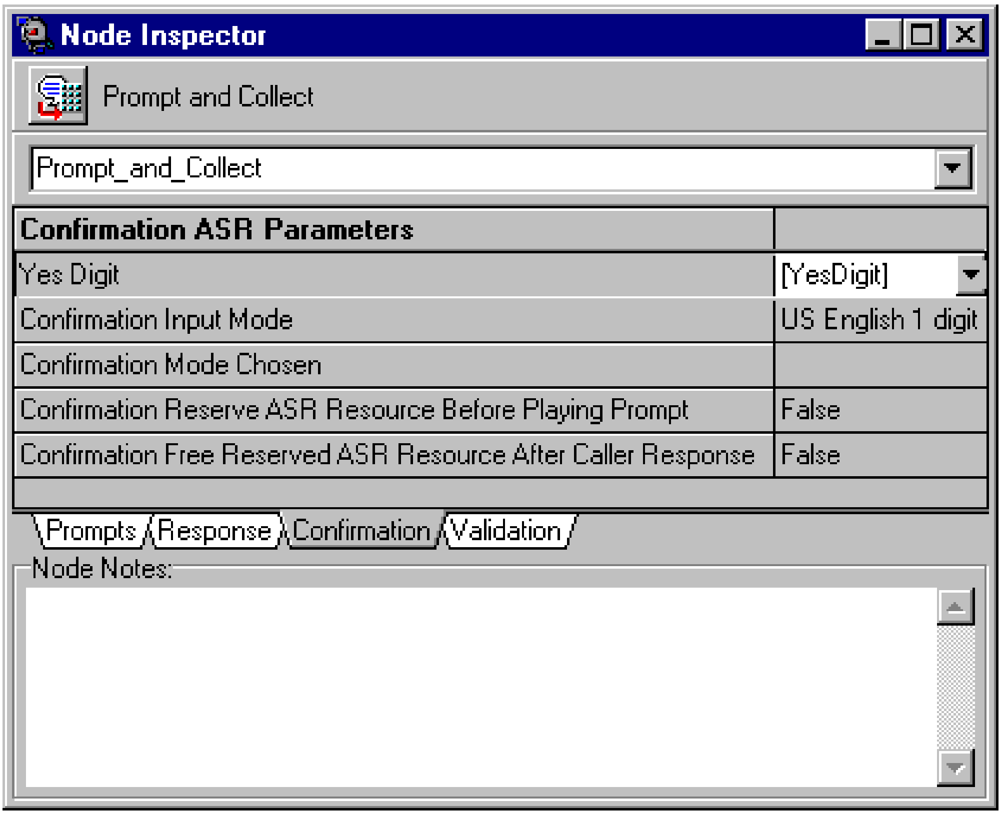


Figure 176. Prompt and Collect — Confirmation Tab

Confirmation ASR Parameters

- **Yes Digit** — Enter the digit, using a variable name or a literal value, to be used to confirm the caller's response. (Default = [YesDigit])
- **Confirmation Input Mode** — From the pulldown menu, select whether you want to use Touchtones, WholeWord speech recognition, or FlexWord speech recognition as confirmation input from the caller.
- **Confirmation Mode Chosen** — (Not available if Touchtone is selected as the Input Mode) From the pulldown menu, select the variable that holds the value indicating how the caller responded to the Confirmation Prompt.
- **Confirmation Reserve ASR Resource Before Playing Prompt** — (Not available if Touchtone is selected as the Input Mode) To make sure that the ASR resource is available before obtaining spoken confirmation, select **True**.
- **Confirmation Free Reserved ASR Resource After Caller Response** — (Not available if Touchtone is selected as the Input Mode) To free the ASR resource as soon as spoken confirmation is retrieved from the caller, select **True**.

The Validation Tab

Figure 177 shows the Validation tab.

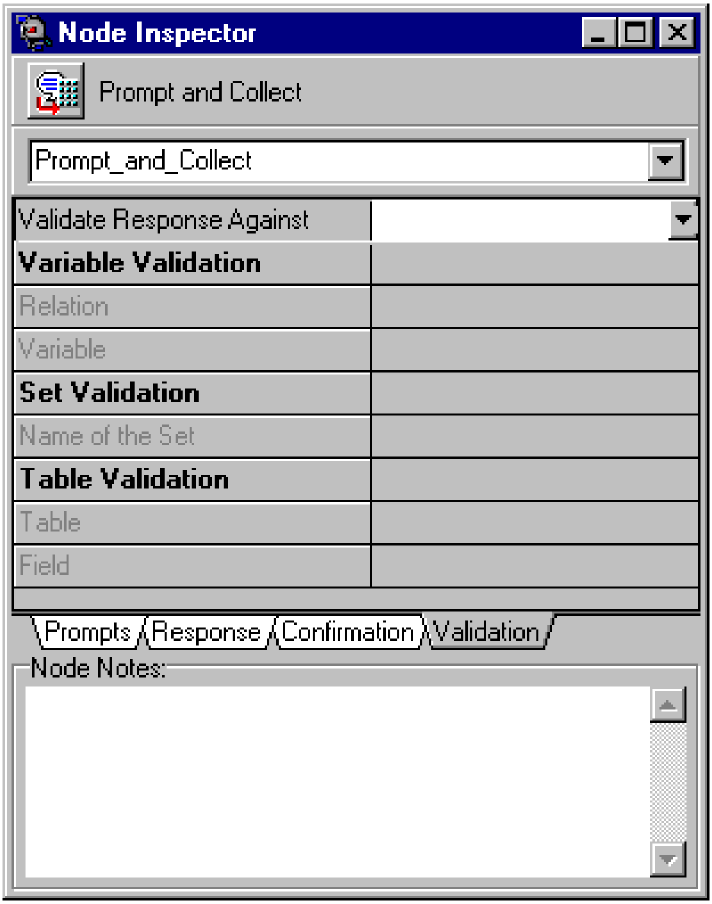


Figure 177. Prompt and Collect Node — Validation Tab

- **Validate Response Against** — From the pulldown menu, select the entity to be used for validating the caller's input. There are four options: **Variable**, **Set**, **Table**, or **(none)**.

Variable Validation

NOTE:
The following attributes are available only if Variable is selected in the **Validate Response Against** field above.

- **Relation** — Enter or from the pulldown menu select the relation symbol (**=**, **<>**, **<**, **<=**, **>**, **>=**) to use in comparing the caller's response to the test variable.
- **Variable** — Enter or from the pulldown menu select the variable containing the value to compare with the data collected from the caller.

Set Validation

NOTE:
The following attribute is available only if Set is selected in the **Validate Response Against** field above.

- **Name of the Set** — Enter the name of the set of values used to validate the caller's response. You must then enter a set of valid values, such as **1-999** or **1,2,7,0**.

Table Validation

NOTE:
The following attributes are available only if Table is selected in the **Validate Response Against** field above.

- **Table** — Enter or from the pulldown menu select the table that contains the field value to be used to validate the caller's response.
- **Field** — Enter or from the pulldown menu select the field within the selected table that contains the value to be used to validate the caller's response.

The Automenu Node

The Automenu node dynamically creates a menu of choices derived from a list of previously retrieved database records (Figure 178).

Example of Use

A customer calling the ABC Automotive Inquiry Line can request to hear a list of this week's sale items. Since the items that are on sale change weekly, the information can be stored in a database table. Items in the database table can be marked as sale items and the sale price of the item can be updated. Only the data in the database table needs to be updated. The call flow information does not have to change.

Upon requesting the list of sale items, the customer hears the following:

> “The following items are on sale during the week of December 12, 1997.
>
> > “For information regarding Super Charger Batteries, press 1.
> >
> > “For information regarding Grason Spark Plugs, press 2.
> >
> > “For information regarding Generex Oil, press 3.
> >
> > “For information regarding APEX Windshield Wiper Fluid, press 4.
>
> “Please make your selection now.”

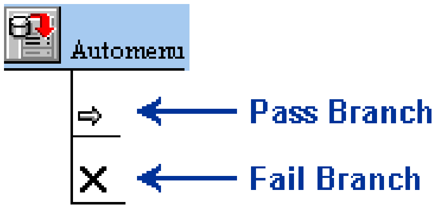


Figure 178. Automenu Node

Behavior

- The Automenu node plays the Initial Prompt as specified in the Node Inspector (see **Prompts Tab** below).
- The Automenu node dynamically creates a menu of choices for the caller, based upon a list of previously retrieved database records.
- The Automenu node directs the call based upon the caller's response to the menu choices.
- If the caller enters an invalid response, the Bad Input Prompt is played. If the **Replay Initial Prompt** field is set to **True**, the Initial Prompt also plays.
- If the caller enters an invalid response and the **Tries Available** field value has been exceeded, the **Fail** branch is executed.
- If the caller does not respond on the last try, the Timeout branch is taken if one has been defined. Otherwise, the Fail branch is taken by default.
- If the caller does not respond within the time set in the **First Digit Timeout** field, the Timeout prompt is played. If multiple digits are being collected from the caller and the caller does not enter a subsequent digit within the time set in the **Inter-Digit Timeout** field, the Timeout prompt is played. If the **Replay Initial Prompt** field is set to **True**, the Initial Prompt also plays.
- If the caller does not respond with enough digits, the Too Few Digits prompt is played. If the **Replay Initial Prompt** field is set to **True**, the Initial Prompt also plays.
- Valid responses from the caller are saved in the appropriate variable.

Attributes

When you insert an Automenu node, Voice@Work creates two default branches which cannot be renamed or deleted. You can, however, insert additional branches which can later be renamed.

Inserting Branches

To insert a new branch on an Automenu node, do the following:

1. To insert an additional branch, right-click where you want the new branch.
2. Select **Insert Branch** from the popup menu.

   Voice@Work displays the Touchtones dialog box (Figure 179).

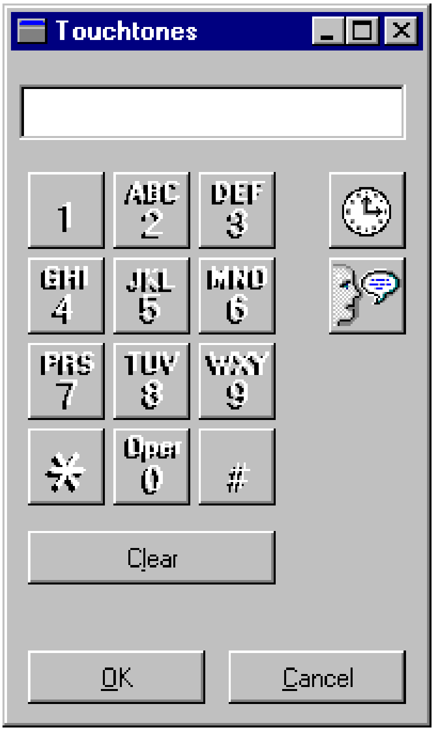
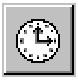
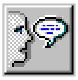


Figure 179. Touchtones Keypad Window

3. Do one of the following:
   - Enter the digit (or digits) you want to assign to the branch.
   - To define the branch as a “timeout” branch, click the  icon.

- To define the branch using an WholeWord grammar enumerated list item, click the  icon.

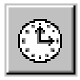
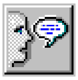
  Clicking this icon opens a Grammar Word List window. This window displays the list items enumerated for the WholeWord grammar designated on the Response tab for this node.

4. Click **OK**.

Renaming Branches

To rename (assign a different number to) a branch, do the following:

1. Right-click the branch you want to rename.
2. Select **Rename Branch** from the popup menu.

   Voice@Work displays the Touchtones dialog box (Figure 179).
3. Do one of the following:
   - Enter the digit or digits you want to assign to the branch.
   - To rename the branch as a “timeout” branch, click the  icon.
   - To rename the branch using an WholeWord grammar enumerated list item, click the  icon.

     Clicking this icon opens a Grammar Word List window. This window displays the list items enumerated for the WholeWord grammar designated on the Response tab for this node.
4. Click **OK**.

Other Attributes

Other attributes are assigned and edited using the Node Inspector (Figure 180).

The Prompts Tab

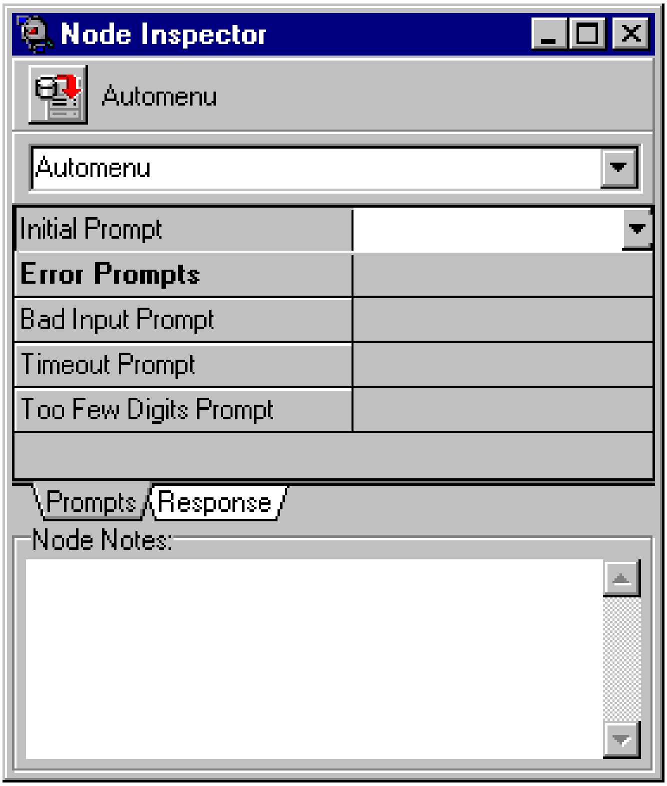


Figure 180. Automenu Node — Prompts Tab

- **Initial Prompt** — From the pulldown menu, select the prompt you want to play when the Automenu begins to execute. The Initial Prompt must be defined using the "Use All Selected Records in Prompt" option in the Prompt Editor window (see "The Use All Selected Records in Prompt Option (Prompt Editor)" in Chapter 19, "Working with Prompts").

Error Prompts

- **Bad Input Prompt** — From the pulldown menu, select the prompt to play when the caller enters invalid input, such as pressing a touchtone which is not defined as one of the choices.
- **Timeout Prompt** — From the pulldown menu, select the prompt to play when the caller waits too long to respond to the Initial Prompt.
- **Too Few Digits Prompt** — From the pulldown menu, select the prompt to play when the caller does not enter enough digits.

The Response Tab

Figure 181 shows the Response tab.

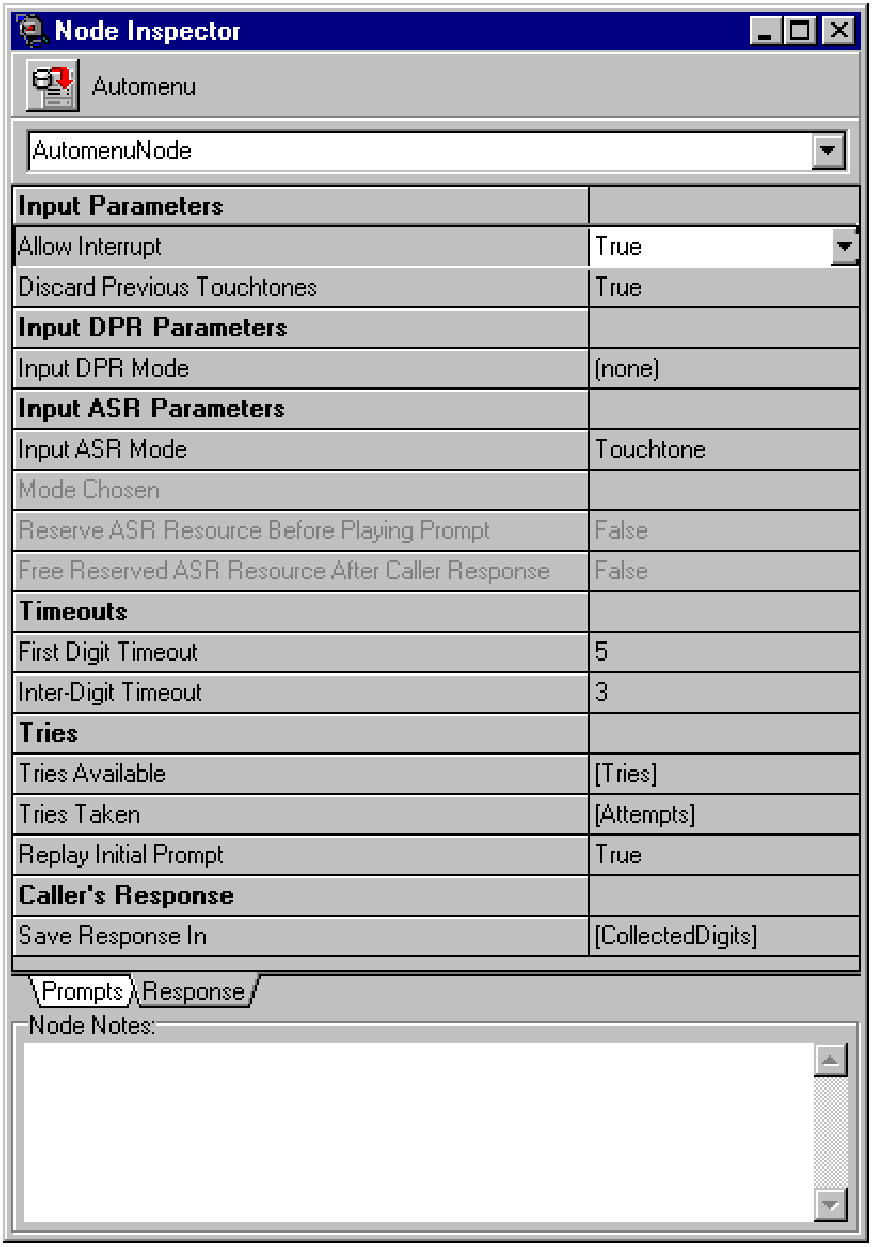


Figure 181. Automenu Node — Response Tab

Input Parameters

- **Allow Interrupt** — To allow the caller to interrupt while prompts are playing, select **True** from the pulldown menu.

  To play prompts without interruption, select **False** from the pulldown menu.

  NOTE:
  You can use this option only if the **Input ASR Mode** option is set to Touchtones or a WholeWord speech recognition grammar. Even then, you can use a WholeWord speech recognition grammar only if you are using an SP or SSP circuit card. For details, see the *INTUITY™ CONVERSANT® System Speech Development, Processing, and Recognition* documentation for your system.

- **Discard Previous Touchtone** — To ignore any touchtones the caller may have entered before the Initial Prompt is played, select **True** from the pulldown menu.

  To process any touchtones the caller may have entered before the Initial Prompt is played, select **False** from the pulldown menu. This allows you to take advantage of the "Type Ahead" feature, which works only with touchtones.

Input DPR Parameters

- **Input DPR Mode** – From the pulldown menu, select the DPR grammar you want to use to collect input from the caller.

Input ASR Parameters

- **Input ASR Mode** — From the pulldown menu, select whether you want to use Touchtones, WholeWord speech recognition, or FlexWord speech recognition as input from the caller.
- **Mode Chosen** — (Not available if Touchtone is selected as the Input Mode) From the pulldown menu, select the variable that holds the value indicating how the caller responded to the Initial Prompt.
- **Reserve ASR Resource Before Playing Prompt** — (Not available if Touchtone is selected as the Input Mode) To make sure that the ASR resource is available before obtaining spoken caller input, select **True**.
- **Free Reserved ASR Resource After Caller Response** — (Not available if Touchtone is selected as the Input Mode) To free the ASR resource as soon as spoken input is retrieved from the caller, select **True**.

Timeouts

- **First Digit Timeout** — Do one of the following:
  - Enter the number of seconds to allow a caller to respond to the Initial Prompt.
  - From the pulldown menu, select a variable to determine the number of seconds to allow a caller to respond to the Initial Prompt.
- **Inter-Digit Timeout** — Do one of the following:
  - Enter the number of seconds to allow a caller for entering each subsequent digit in a multiple-digit response.
  - From the pulldown menu, select a variable to determine the number of seconds to allow a caller for entering each subsequent digit in a multiple-digit response.

Tries

- **Tries Available** — Do one of the following:
  - Enter the number of tries to allow a caller to respond to the Initial Prompt.
  - From the pulldown menu, select a variable to determine the number of tries to allow a caller to respond to the Initial Prompt.
    (Default = 3)
- **Tries Taken** — From the pulldown menu, select a variable to hold the number of tries it took a caller to enter a correct response.
- **Replay Initial Prompt** — To replay the Initial Prompt, after any error messages, each time the caller enters an invalid response, select **True**.

Caller’s Response

- **Save Response In** — From the pulldown menu, select the variable in which to save the caller's unformatted response.

The Voice Capture Node

The Voice Capture node is used primarily to record a caller's spoken response. Use this node to gather information that cannot be entered with touchtones, for example, from callers with rotary telephones. You can use it also to collect alphanumeric input from callers, for example, street addresses or city names (Figure 182).

Example of Use

ABC Automotive wants to gather names and addresses from customers calling the Inquiry Line. Callers are prompted:

> "If you would like to be on our mailing list, please say your name, street address, city, state, and zip code.
>
> Please begin recording at the sound of the tone."

When the caller responds, the information is stored in a variable for later transcription to a mailing database.

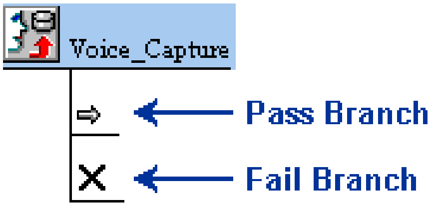


Figure 182. Voice Capture Node

Behavior

- When a caller speaks a valid response, the Pass branch is taken.
- If a caller does not speak a response and has exceeded the number of tries, the Fail branch is taken.

Attributes

Attributes are assigned and edited using the Node Inspector (Figure 183).

The Prompts Tab

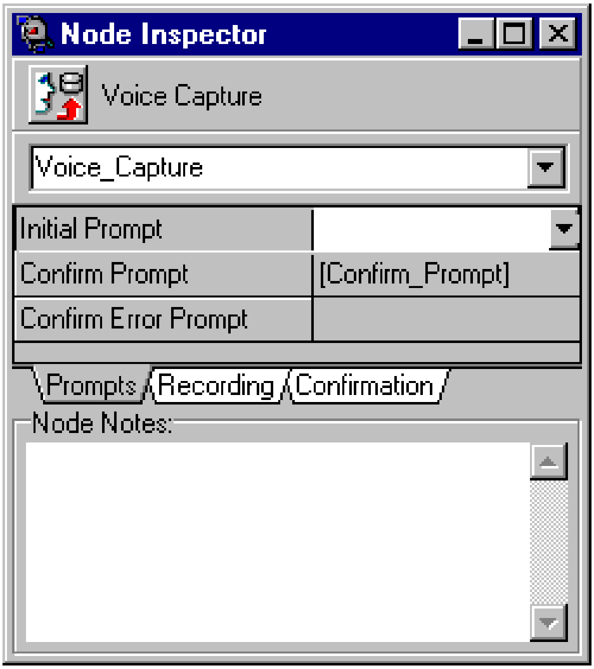


Figure 183. Voice Capture Node — Prompts Tab

- **Initial Prompt** — From the pulldown menu, select the prompt you want to play when the Voice Capture node begins to execute. The Initial Prompt must be defined using the Prompt Editor window. See "The Prompt Editor" in Chapter 19, "Working with Prompts", for additional information.
- **Confirm Prompt** — From the pulldown menu, select the prompt to play after the caller responds. This prompt is used to get feedback as to whether the caller's initial response was correct.
- **Confirm Error Prompt** — (Inactive until a Confirm Prompt has been selected) From the pulldown menu, select the prompt to play when the caller does not enter a valid **Yes** or **No** digit.

The Recording Tab

Figure 184 shows the Recording tab.

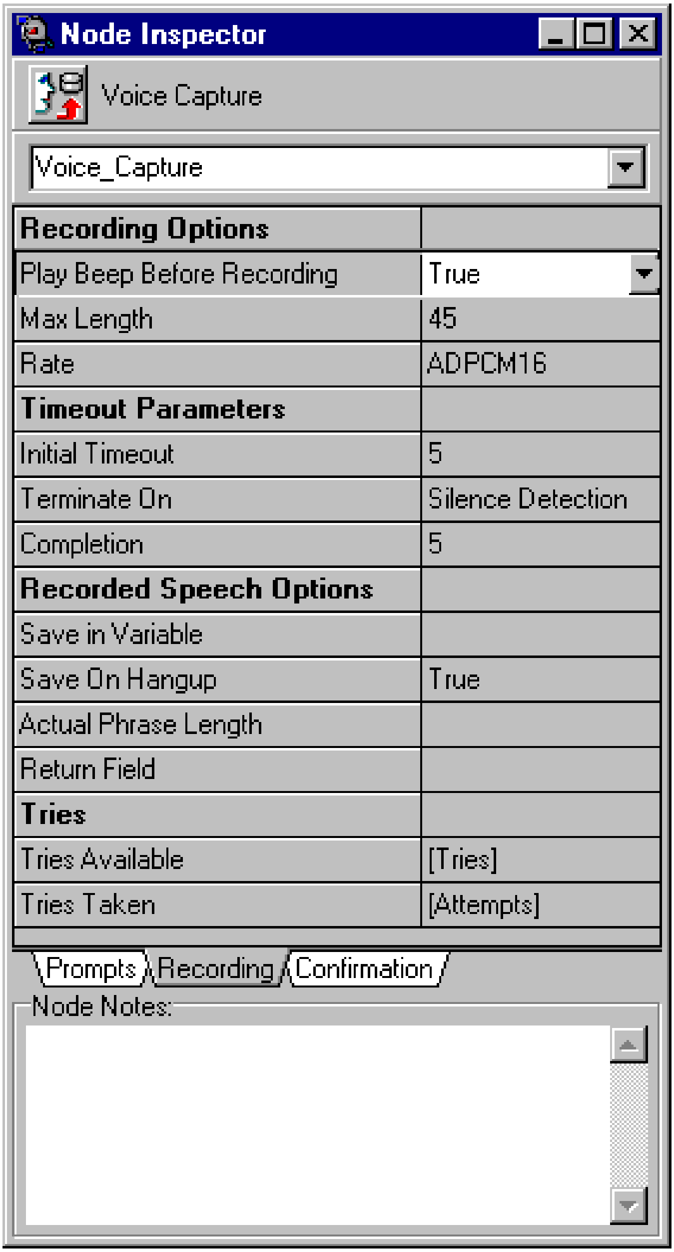


Figure 184. Voice Capture Node — Recording Tab

Recording Options

- **Play Beep Before Recording** — When set to ***True*** (default), this option plays a beeping tone at the end of the Initial Prompt before beginning the recording.
- **Max Length** — Enter or from the pulldown menu select the value or variable that contains the value for the maximum length of time (in seconds) to allow for each caller's response. Range = 1–999, default = 45
- **Rate** — From the pulldown menu, select the voice-encoding algorithm to use when recording the caller's voice. Options include:
  - ADPCM16 (Default)
  - ADPCM32
  - PCM64
  - SBC16
  - SBC24
  - SBC24AA

  NOTE:
  For more information about these voice-encoding algorithms, see the *INTUITY™ CONVERSANT® System Speech Development, Processing, and Recognition* documentation for your system.

Timeout Parameters

- **Initial Timeout** — Enter or from the pulldown menu select the number or the variable containing the number of seconds to wait for the caller to respond before replaying the Initial Prompt. Default = 5
- **Terminate On** — From the pulldown menu, select the means by which the end of the recording is detected and the recording is terminated. Options include:
  - **Maximum Duration** — Terminates the response only when the value in the **Max Length** field is reached.
  - **Silence Detection** — (Default) Terminates the response when a user-determined amount of silence is detected. The number of seconds of silence is set in the **Completion** field (see below).
  - **Touchtone Detection** — Terminates the response when a touchtone is detected.
- **Completion** — (Used only with the Silence Detection option above) Enter the number or from the pulldown menu select the variable containing the number of seconds of silence to use in detecting the end of the recording. Default = 5

Recorded Speech Options

- **Save in Variable** — Enter the variable name or from the pulldown menu select the variable to be used to store the caller's response (must be a number variable with a Spoken As setting of **NX** and a Speak Using setting of **recorded** in the Variables Manager. See Chapter 14, "Working with Variables", for additional information.
- **Save On Hangup** — When set to **True** (default), this option saves the caller's response even if the caller hangs up.
- **Actual Phrase Length** — Type or select from the pulldown menu the name of the variable to assign the number representing the length of the recorded phrase in seconds.
- **Return Field** — Type or select from the pulldown menu the name of the variable to assign the return code value from the voice capture operation. Table 98 lists and briefly describes the return codes for the Voice Capture node.

Table 98. Voice Capture Node Return Codes

Return Code	Description
1	Coding completed normally
2	Touchtone termination
3	Completion timeout
-1	Insufficient space
-2	Code failure
-3	Initial timeout
-4	Caller hangup

Tries

- **Tries Available** — Do one of the following:
  - — Enter the number of tries to allow a caller to respond to the Initial Prompt.
  - — From the pulldown menu, select a variable to determine the number of tries to allow a caller to respond to the Initial Prompt.
    Default = [Tries]
- **Tries Taken** — From the pulldown menu, select a variable to hold the number of tries it took a caller to enter a correct response. Default = [Attempts]

The Confirmation Tab

Figure 185 shows the Confirmation tab.

NOTE:
The options on this tab are only available if a Confirm Prompt is selected on the Prompts tab.

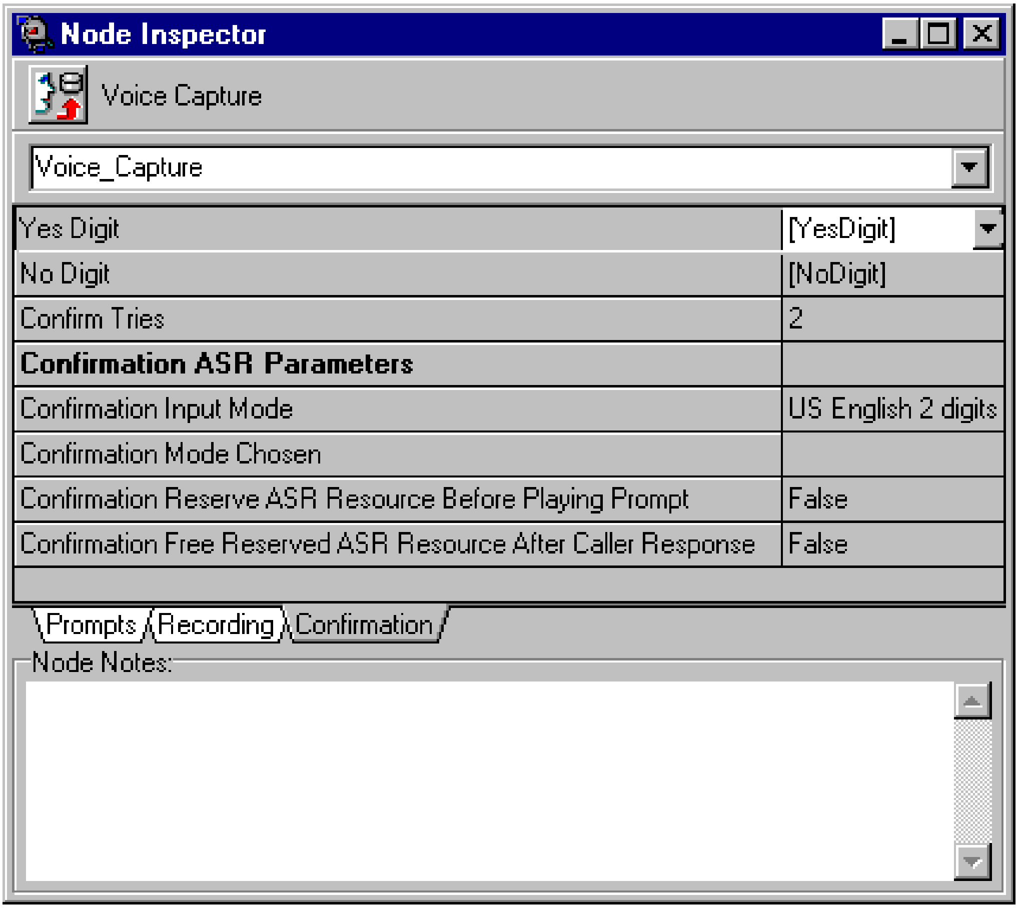


Figure 185. Voice Capture Node — Confirmation Tab

- **Yes Digit** — Enter the digit, using a variable name or a literal value, to be used to confirm the caller's response. Default = [YesDigit]
- **No Digit** — Enter the digit, using a variable name or a literal value, to be used to negate a caller's response. Default = [NoDigit]
- **Confirm Tries** — Enter the number or from the pulldown menu select the variable representing the number of tries the caller has to verify a correct recording. Default = 2

Confirmation ASR Parameters

- **Confirmation Input Mode** — From the pulldown menu, select whether you want to use Touchtones (default), WholeWord speech recognition, or FlexWord speech recognition as confirmation input from the caller.
- **Confirmation Mode Chosen** — (Not available if Touchtone is selected as the Input Mode) From the pulldown menu, select the variable that holds the value indicating how the caller responded to the Confirmation Prompt.
- **Confirmation Reserve ASR Resource Before Playing Prompt** — (Not available if Touchtone is selected as the Input Mode) To make sure that the ASR resource is available before obtaining spoken confirmation, select **True**. (Default = False)
- **Confirmation Free Reserved ASR Resource After Caller Response** — (Not available if Touchtone is selected as the Input Mode) To free the ASR resource as soon as spoken confirmation is retrieved from the caller, select **True**. (Default = False)

Database Nodes

This section describes the database nodes.

The Query Table Node

The Query Table node retrieves records from a database table based upon the specified selection criteria (Figure 186).

Example of Use

A customer calls the ABC Automotive Inquiry Line about the status of an order. The order number is obtained from the customer and is used as the selection criteria to retrieve the order status from a local database table.

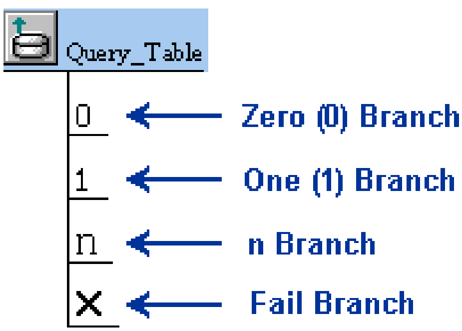


Figure 186. Query Table Node

Behavior

- Records are selected from the specified database table based upon the defined selection criteria (see “Attributes” below).
- When selection criteria are not specified, *all* the records in the database table are retrieved.
- The field values from the first record found that matches the selection criteria is placed in the **Assignments from records** variable (see “Attributes” below).
- If no records matching the selection criteria are found, the Zero (0) branch is taken.
- If only one record matching the selection criteria is found, the One (1) branch is taken.
- If more than one record matching the selection criteria are found, the n branch is taken.
- If the specified database table could not be accessed for some reason, the Fail branch is taken.
- If multiple records matching the selection criteria are found, the values from the first record found are returned. The remaining records can be accessed by using the NextRecord or PrevRecord function of the Set and Test node, or by using the Automenu node to speak all the selected records back to the caller.

Attributes

Attributes are assigned and edited using the Node Inspector (Figure 187).

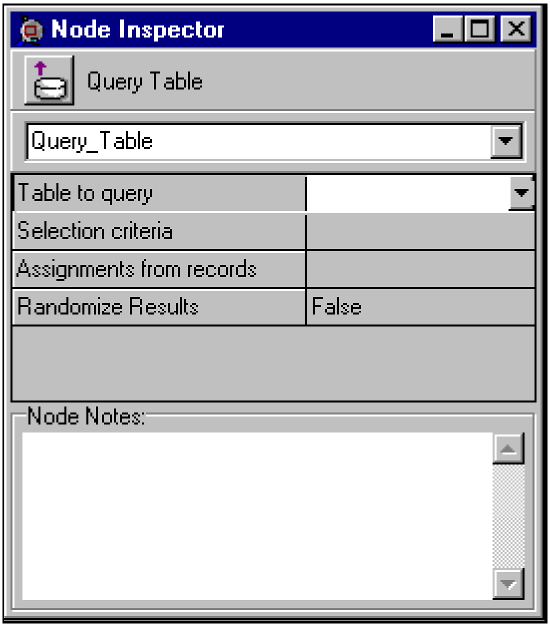


Figure 187. Node Inspector — Query Table

- **Table to query** — Enter or from the pulldown menu select the table from which the records are to be selected.

NOTE:
The following two fields both allow you to open the Query Table Editor, and you can accomplish what you want to do for both fields at the same time. You do not have to close and re-open the Query Table Editor for each field.

- **Selection criteria** — Displays the fields in the database table and the variables or literal values to which the fields are to be compared. When the button is clicked, Voice@Work displays the Query Table Editor window (Figure 188). Use this editor to assign data from record fields to values or variables.
- **Assignments from records** — Displays the fields in the database table and the variables that will be assigned the values found in each field. When the button is clicked, Voice@Work displays the Query Table Editor window (Figure 188). Use this editor to assign data from record fields to values or variables.

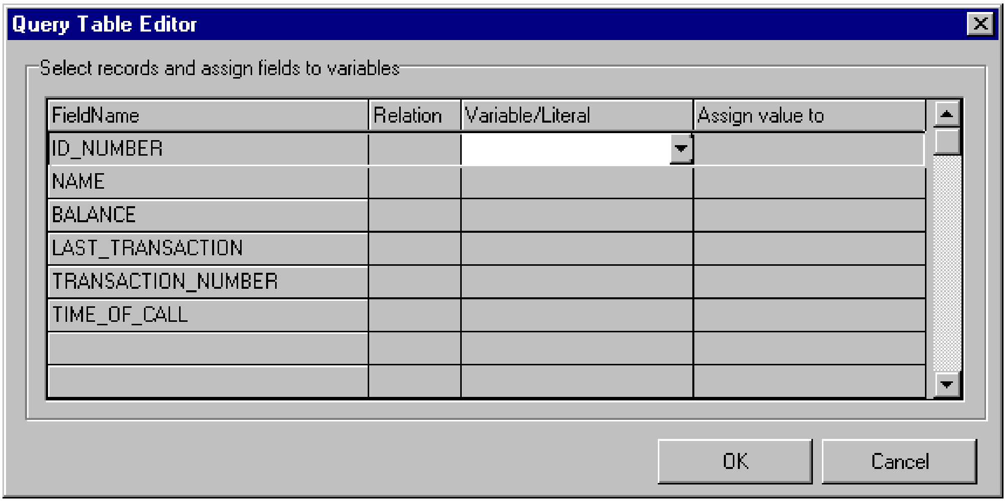


Figure 188. Query Table Editor

Table 99 describes the column fields in the Query Table Editor.

Table 99. Query Table Editor Fields

Field/Column	Description
Field Name	Displays the names of all defined fields for the selected table.
Relation	Allows you to select and displays the relation operator for the selected variable field. Possible value: **=**

Continued on next page

Table 99. Query Table Editor Fields — *Continued*

Field/Column	Description
Variable/Literal	Allows you to select from a pulldown menu the variable, the value of which to compare with the database field, to set as the selection criteria. **NOTE:** The variable type for this variable or literal value must match the type for the **Assign value to** field.
Assign value to	Allows you to select from a pulldown menu the variable to assign the value from the first record that matches the selection criteria. For additional records, you can use the NextRecord and PrevRecord functions in the Set and Test node. **NOTE:** The variable type for this variable (or literal value) must match the type for the **Variable/Literal** field.

The Update Table Node

The Update Table node is used to change data within a specified database table (Figure 189).

Example of Use

A customer calls ABC Automotive wanting to change the quantity of a particular item ordered earlier. The order information is obtained from the caller and used as selection criteria to locate the order information records in the database. If a match is found, the quantity field can be updated in the database table.

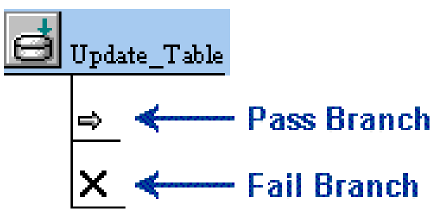


Figure 189. Update Table Node

Behavior

- When a record is found that matches the selection criteria, the record is changed in the database table.
- If no selection criteria are specified, all records are selected from the specified database table.
- If **Update multiple records** is set to **True** and at least one record matching the selection criteria is found, all records found are updated.
- If **Update multiple records** is set to **False** and more than one record matching the selection criteria is found, no records are updated.
- If one or more records in the database table are updated, the Pass branch is taken.
- If no records in the database table are updated, the Fail branch is taken.
- If for some reason the database table cannot be accessed, the Fail branch is taken.

Attributes

Attributes are assigned and edited using the Node Inspector (Figure 190).

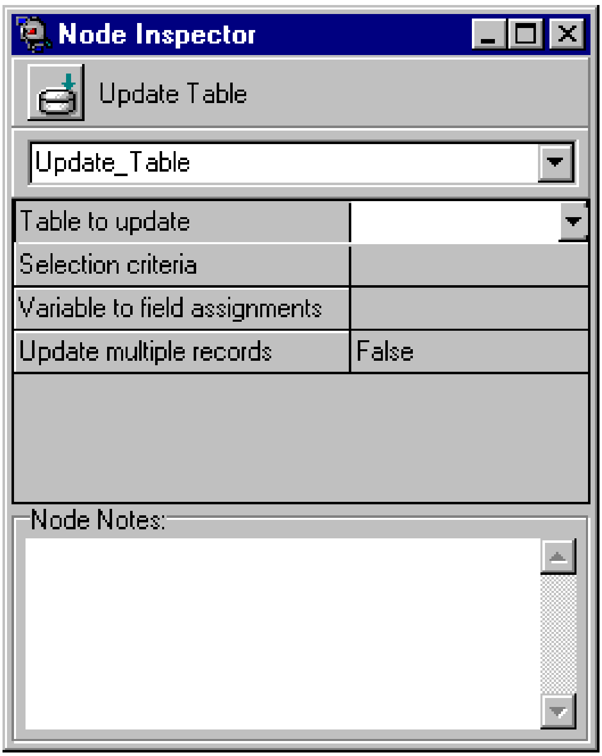


Figure 190. Node Inspector — Update Table

- **Table to update** — Enter or from the pulldown menu select the table in which the records are to be changed.

NOTE:
The following two fields both allow you to open the Update Table Editor, and you can accomplish what you want to do for both fields at the same time. You do not have to close and re-open the Update Table Editor for each field.

- **Selection criteria** — Displays the fields in the database table and the variables or literal values to which the fields are to be compared. When the button is clicked, Voice@Work displays the Update Table Editor window (Figure 191). Use this editor to assign data from record fields to values or variables.
- **Assignments from records** — Displays the fields in the database table and the variables that will be assigned the values found in each field. When the button is clicked, use this editor to assign data from record fields to values or variables.
- **Update multiple records** — When set to **True**, this option allows you to update more than one record at a time.

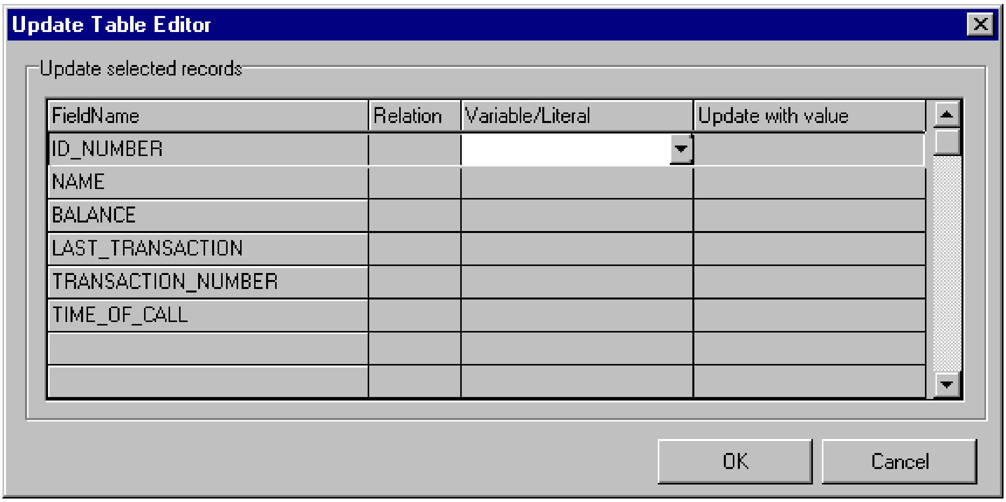


Figure 191. Update Table Editor

Table 100 describes the column fields in the Update Table Editor.

Table 100. Update Table Editor Fields

Field/Column	Description
Field Name	Displays the names of all defined fields for the selected variable
Relation	Allows you to select and displays the relation operator for the selected variable field. Possible values: **=**
Variable/Literal	From the pulldown menu, select the variable to set as the selection criteria for updating the variable. **NOTE:** The variable type for this variable or literal value must match the type for the **Update with value** field.
Update with value	From the pulldown menu select the variable containing data to update the field with. For additional records, you can use the NextRecord and PrevRecord functions in the Set and Test node. **NOTE:** The variable type for this variable or literal value must match the type for the **Variable/Literal** field.

The Add Record Node

The Add Record node adds a record to a database table during application execution (Figure 192).

Example of Use

A customer places an order with ABC Automotive. Once the order information is obtained from the caller, the corresponding order information can be added to a local database table by using the Add Record node.

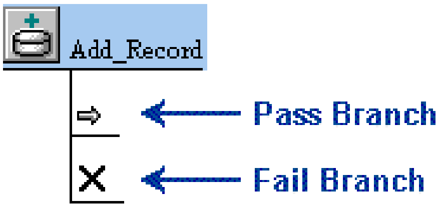


Figure 192. Add Record Node

Behavior

- If a record is added to the database table, the Pass branch is taken.
- If a record cannot be added (for example, because it would create a duplicate record for a field defined as a key field), the Fail branch is taken.

Attributes

Attributes are assigned and edited using the Node Inspector (Figure 193).

Figure 193. Node Inspector — Add Record Node

- **Table to add record** — From the pulldown menu, select the database table to which the record is to be added.
- **Assignments to new record** — This attribute displays values to be assigned to the new record. When the button is clicked, Voice@Work displays the Add Record Editor window (Figure 194). Use this editor to populate the fields of the record with values or variables.

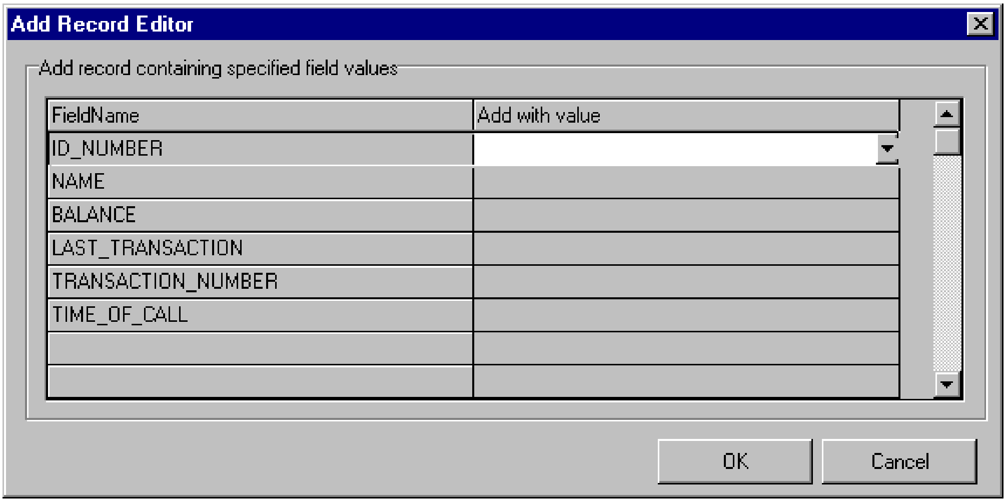


Figure 194. Add Record Editor

Table 101 describes the column fields in the Add Record Editor.

Table 101. Add Record Editor Fields

Field/Column	Description/Comments
FieldName	This column displays the names of all defined fields for the selected variable.
Add with value	From the pulldown menu, select the variable containing data to add to the field.

The Delete Record Node

The Delete Record node deletes a record from a database table during application execution (Figure 195).

Example of Use

A customer wants to cancel an existing order. Once the order information is obtained from the caller, the information is used as selection criteria. If a match is found, the order can be deleted from the local database table.

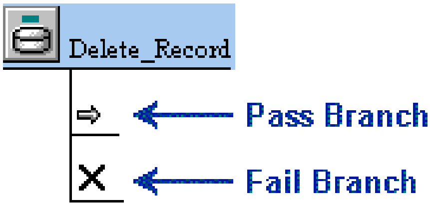


Figure 195. Delete Record Node

Behavior

- Records are selected from the database table based upon the selection criteria. If no selection criteria are specified, all records in the database table are selected.
- If one or more records are deleted from the database table, the Pass branch is taken.
- If multiple records are found and the **Delete multiple records** field is set to False, no records are deleted and the Fail branch is taken.
- If a record cannot be deleted (for example, because the database table resource is not available), the Fail branch is taken.

Attributes

Attributes are assigned and edited using the Node Inspector (Figure 196).

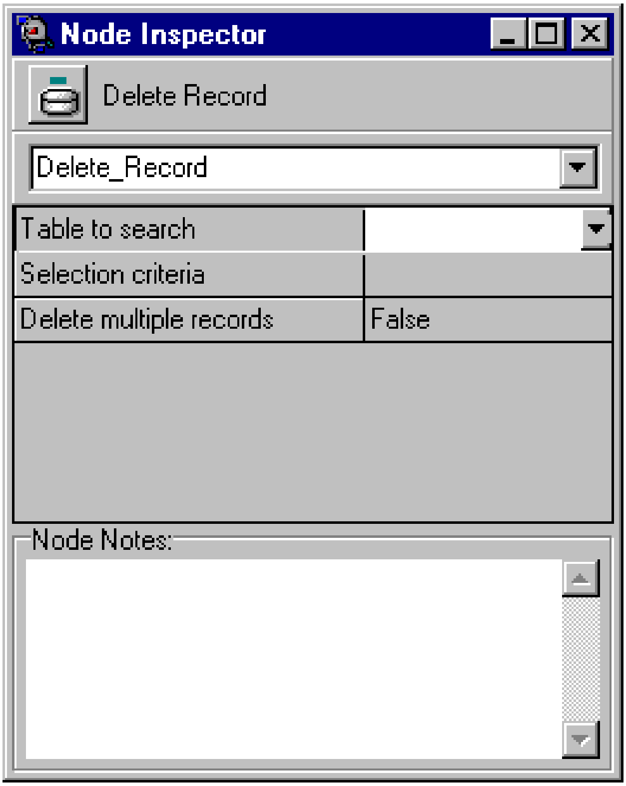


Figure 196. Node Inspector — Delete Record Node

- **Table to search** — From the pulldown menu, select the database table from which the record is to be deleted.
- **Selection criteria** — Displays selection criteria, if any, to be used in deleting the records. When the button is clicked, Voice@Work displays the Delete Record Editor window (Figure 197). Use this editor to define selection criteria for the fields of the record with values or variables.
- **Delete multiple records** — When set to **True**, this option deletes all records that match the selection criteria.

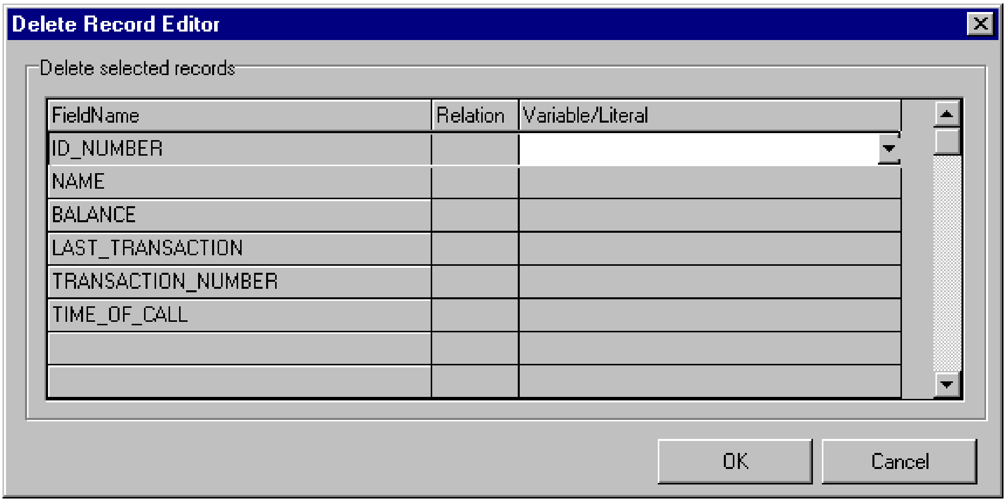


Figure 197. Delete Record Editor

Table 102 describes the column fields in the Delete Record Editor.

Table 102. Delete Record Editor Fields

Field/Column	Description/Comments
FieldName	This column displays the names of all defined fields for the selected variable.
Relation	This column allows you to select and displays the relation operator for the selected variable field. Possible value: **=**
Variable/Literal	From the pulldown menu, select the variable containing data to delete from the database.

Host Nodes

This section describes the host nodes.

The Host Connection Node

The Host Connection node determines the status of the connection with a host machine and allocates a logical unit (LU) for this call (Figure 198).

NOTE:
This node is typically one of the first ones in an application using remote hosts. The Host Connection node should be placed before the Answer Call node because host connectivity should be checked before accepting incoming calls. Otherwise, the caller can be subjected to a period of silence of up to 5 seconds while the connection is being checked.

Example of Use

Your application needs to download account information from a remote host for a customer on the line. Before you can get account information from the host, however, you need to verify that the host is connected and that an LU has been allocated for the call. If the host is *not* connected, it may take several seconds to establish a connection. Meanwhile, the customer is on the line waiting to find out the status of the account. You can use the Host Connection node to verify that the host is connected before the call is answered.

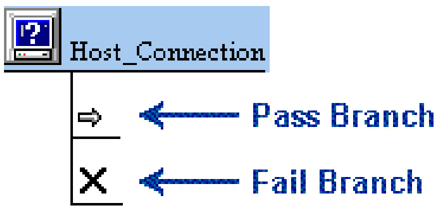


Figure 198. Host Connection Node

Behavior

- If the connection is established within the allotted time, the Pass branch is taken.
- If the connection cannot be established within the allotted time, the Fail branch is executed.
- When a connection with the host has been established, an LU is assigned to the application. The LU remains assigned to the application until it is released or the call is terminated. If an LU is not available, the Fail branch is executed.

Attributes

Attributes are assigned and edited using the Node Inspector (Figure 199).

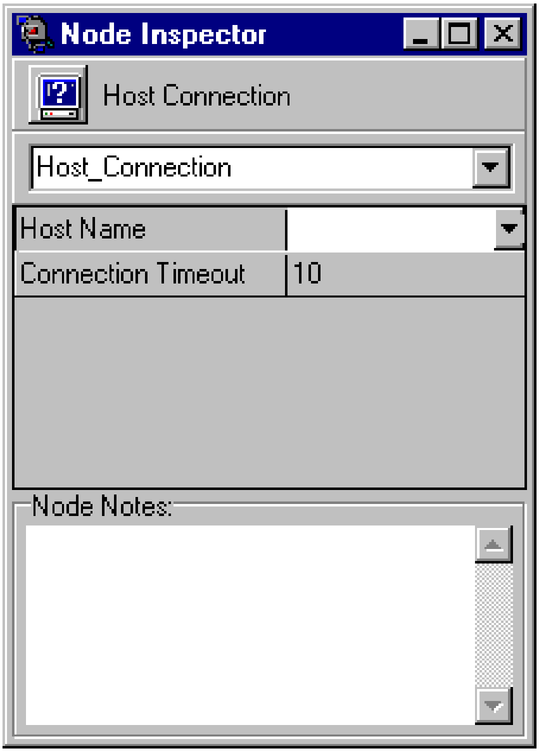


Figure 199. Node Inspector — Host Connection Node

- **Host Name** — Enter the name of the host, or select the name of the host from the pulldown menu.

  NOTE:
  The host must be defined in the Host Definition Editor before you can use this node. See "The Host Definition Editor" in Chapter 17, "Working with Hosts", for additional information.

- **Connection Timeout** — Enter the number of seconds to allow for the connection to be established.

  NOTE:
  The value of this field should be greater than or equal to the value defined for the Logical Unit Timeout on the Host Definition tab of the Host Definition Editor. See "The Host Definition Tab" in "The Host Definition Editor" in Chapter 17, "Working with Hosts", for additional information.

The Host Send Node

The Host Send node sends data to a host machine (Figure 200).

Example of Use

Your application needs to upload customer account information to a remote host. Before you can send the account information to the host, a login sequence must be established for security purposes. The Host Send node can be used to send the necessary login and password information, as well as the account information.

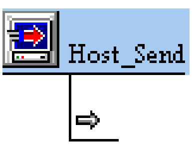


Figure 200. Host Send Node

Behavior

- When taken, the Host Send node sends data to the designated host machine. The transmission of data is considered a "blind send" because the application does not know whether the host received the data. The host does *not* confirm the transmission.
- Once the data is transmitted, the default exit branch is taken.

Attributes

Attributes are assigned and edited using the Node Inspector (Figure 201).

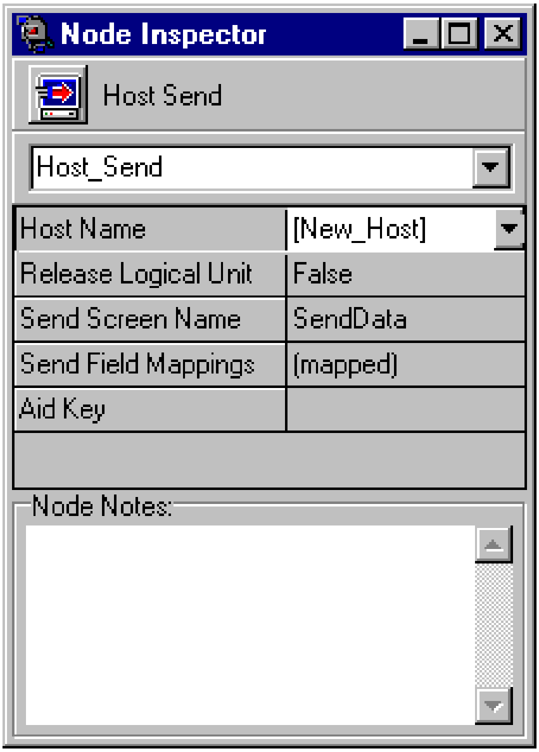


Figure 201. Node Inspector — Host Send Node

- **Host Name** — Enter the name of the host, or select the name of the host from the pulldown menu.

  NOTE:
  The host must be defined in the Host Definition Editor before you can use this node.

- **Release Logical Unit** — When set to **True**, this attribute releases the LU after the data is transmitted.
- **Send Screen Name** — From the pulldown menu, select the name of the screen to send to the host. The screen names available are based on the host definition.

- **Send Field Mappings** — Click the button to open the Host Screen Field Mappings window. From this window, select which fields of the selected screen to send to the host.

  NOTE:
  This option is active *only* if you have set the **Direction** field on the Field Definition tab of the Host Definition Editor to **To Host** or **Both**. See "The Host Definition Editor" in Chapter 17, "Working with Hosts", for additional information.

- **Aid Key** — (Optional) From the pulldown menu, select an Aid Key to send to the host with the mapped fields.

The Host Send/Receive Node

The Host Send/Receive node both sends data to and receives data from a host machine in the network (Figure 202).

Example of Use

A customer wants account information that resides on a remote host. Your application sends the caller's account number and Personal Identification Number (PIN) *to* the host and gets the account balance *from* the host in return.

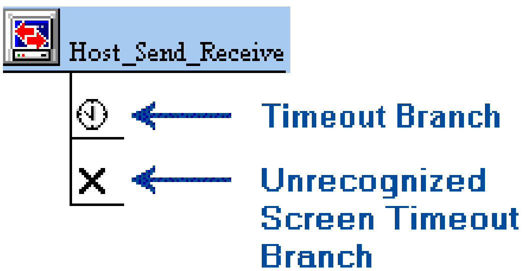


Figure 202. Host Send and Receive Node

Behavior

- If no response from the host is received within the specified time, the Timeout branch is taken.
- The application executes the Unrecognized Screen Timeout branch when the Receive Screen data sent from the host is not the correct screen (that is, when the data sent from the host does not match the format defined on the Host Send/Receive screen), and the correct format was not sent within the time allotted in the host definition.

Attributes

Attributes are assigned and edited using the Node Inspector (Figure 203).

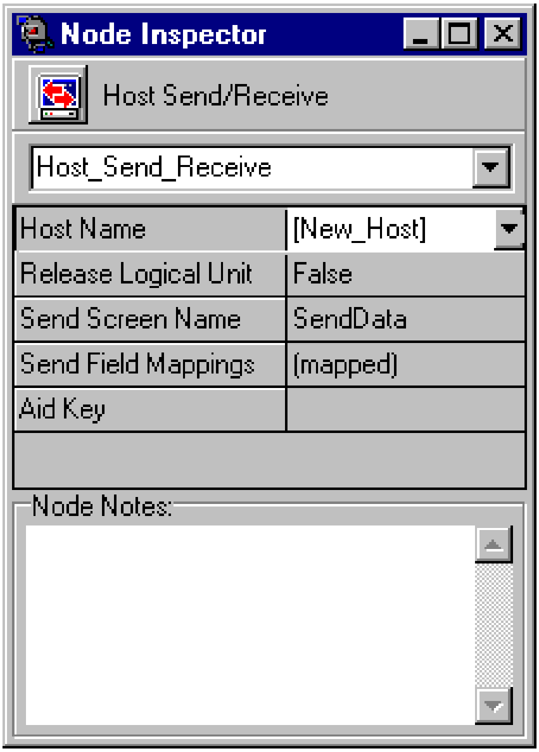


Figure 203. Node Inspector Window — Host Send/Receive Node

- **Host Name** — Enter the name of the host, or select the name of the host from the pulldown menu.

  NOTE:
  The host must be defined in the Host Definition Editor before you can use this node. See "The Host Definition Editor" in Chapter 17, "Working with Hosts", for additional information.

- **Release Logical Unit** — When set to **True**, this attribute releases the LU after the data is transmitted.
- **Send Screen Name** — From the pulldown menu, select the name of the screen to send to the host. The screen names available are based on the host definition.

- **Send Field Mappings** — Click the  button to open the Host Screen Field Mappings window. From this window, select which fields of the selected screen to send to the host.

  NOTE:
  This option is only available if at least one Direction column field value in the Field Definition tab of the Host Definition Editor is set to **To Host** or **Both**, and a new branch has been inserted on the appropriate node.

- **Aid Key** — From the pulldown menu, select an Aid Key to send to the host.

Miscellaneous Nodes

This section describes the miscellaneous nodes.

The Inline Code Node

The Inline Code node allows you to write TAS code directly in your Voice@Work application (Figure 204). When the code is generated and the application is transferred, the TAS code is inserted at the point where this node is placed in the application.

For more information about writing and using TAS code, see the *INTUITY™ CONVERSANT® System Application Development with Advanced Methods* documentation for your system.

Example of Use

The developer writing the application to handle calls for the ABC Automotive Inquiry Line wants to include comments in the final source code to be installed on the target voice response system. Since the Code Generation tool does not automatically include comment blocks, the developer uses the Inline Code node to insert comments at the desired places in the application.

The same developer also uses the Inline Code node to call another application and return to the current application when the other is done executing. This is executed much like a subroutine.

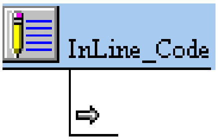


Figure 204. Inline Code Node

Behavior

- This node takes TAS code as you have written it and inserts it into the generated application code.
- Since the TAS code can do whatever the developer wants it to do, the behavior of this node depends on what has been written into it.
- This node allows you to paste from the clipboard into the Inline Code Editor.

CAUTION:
Code is inserted “as is” into your application with no syntax checks performed. Any errors are not detected until you attempt to install the application on the target system.

Attributes

Attributes are assigned and edited using the Node Inspector (Figure 205).

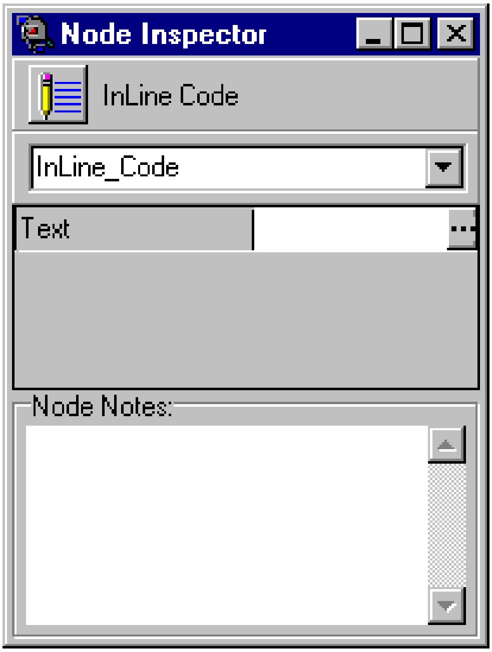


Figure 205. Node Inspector — Inline Code Node

- **Text** —The field displays the first part of any code you have written. To write the TAS code, click the button. Voice@Work displays the Inline Code window (Figure 206). Enter your TAS code in the **Text of Code** field. When you are finished, click **OK**.

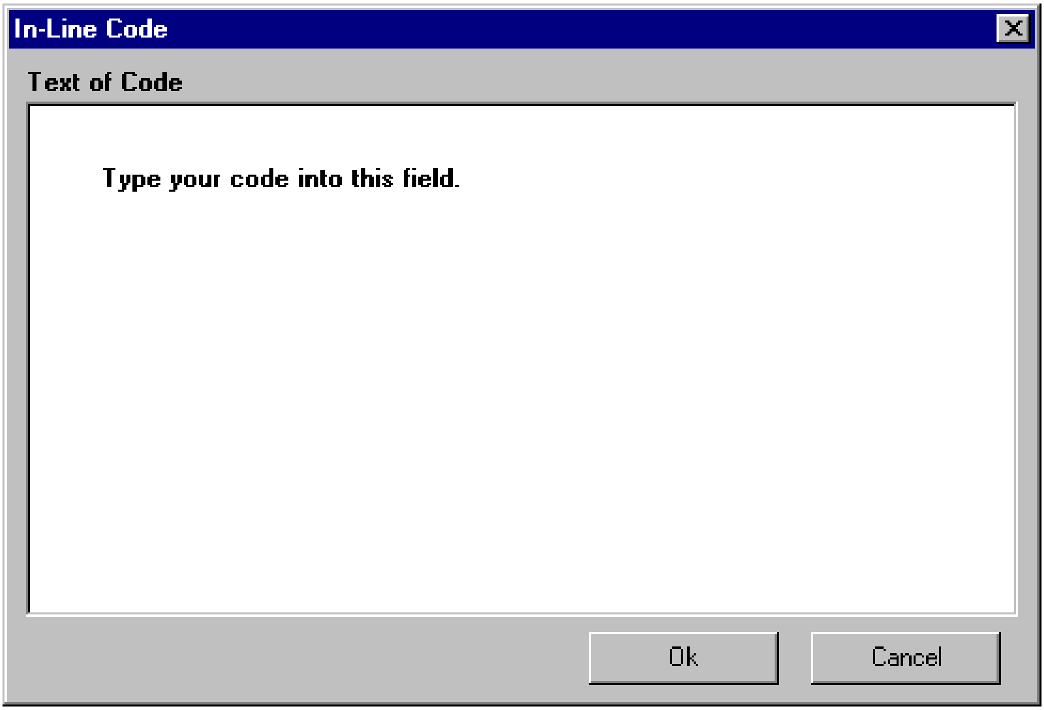


Figure 206. Inline Code Window

The Trace Variables Node

The Trace Variables node allows you to print out values of selected variables on the target voice response system (Figure 207). This is useful primarily as a debugging tool after installing your application on the target system.

NOTE:
To be useful, this node requires that you run the **trace** command on the target system. See the *INTUITY™ CONVERSANT® System Description* documentation for your system.

Example of Use

You have installed your application on the target system, and you want to make sure the application is updating a particular variable properly. You would place a Trace Variables node in the call flow both before and after the variable is supposed to be updated. The output from the **trace** command on the INTUITY CONVERSANT system displays the value of the variable both before and after it is updated.

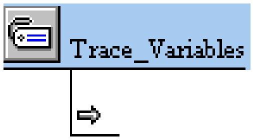


Figure 207. Trace Variable Node

Behavior

- The selected variable is "tagged" to be traced when installed on the target voice response system.
- When the **trace** function is invoked on the INTUITY CONVERSANT system, the value for the selected variable is printed to the output.

Attributes

Attributes are assigned and edited using the Node Inspector (Figure 208).

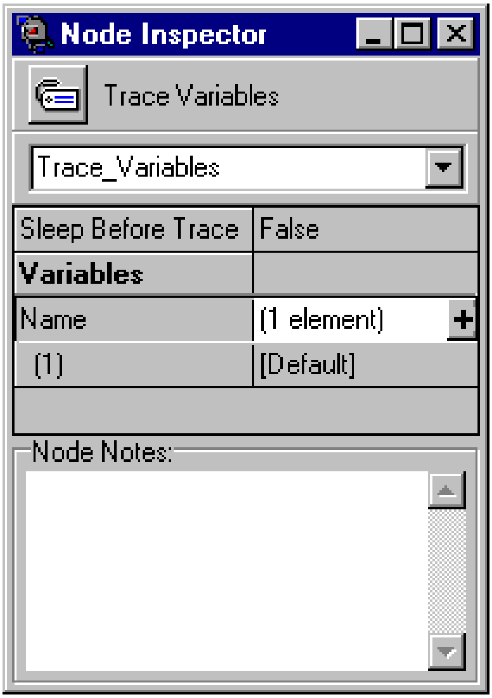


Figure 208. Node Inspector — Trace Variables Node

- **Sleep Before Trace** — When set to **True**, this attribute inserts a one-second pause in the code before the **trace** function is executed on the target system. This can be useful when you need to flush the buffer before executing the **trace** command.

Variables

- **Elements**
  - Click the icon to add variables to trace at this point in the call flow, and then select the variable you want from the pulldown menu.
  - The **# elements** field displays how many variables have been selected.
  - The **Name** column field displays the number assigned to each variable selected.
  - From the corresponding pulldown menu, select the variable whose values you want to trace. All variables defined for the current application are displayed on this pulldown menu.

The Save and Exec Node

NOTE:
In earlier releases of Voice@Work, this was called the Execute Script node.

The Save and Exec node calls another application on the target system (Figure 209). Once control leaves the current application, it does not return. This node is typically used in conjunction with the Load Variables node.

NOTE:
The Save and Exec *node* is similar but not identical to the Execute *external function*. The differences lie in the way the two pass arguments and variables to the new application. When calling a Script Builder application, you *must* use the Execute external function because Script Builder does not have Load Variables node functionality or any equivalent action. For Voice@Work applications, you can use either the Save and Exec node or the Execute external function.

Example of Use

ABC Automotive has two developers working on the script for its Inquiry Line, which is a very large application. To make it easier for the two developers to do their work, they split the script into two parts, with each developer working on part of it as a separate application. They design it in such a way that control passes from one part to the other at a certain point in the script, and certain variable values are passed at the same time. They use the Save and Exec node to pass control from one part to the other.

Figure 209. Save and Exec Node

Behavior

- This node calls another application (script) and executes it, passing control to the new application.
- Control does not return to the current application. Control stays with the new application.
- This node can pass 2 Kbytes of variable data as part of the node's operation.
- If for some reason the target system cannot execute the new application — if, for example, the application does not exist on the target system — the Fail branch is taken.

Attributes

Attributes are assigned and edited using the Node Inspector (Figure 210).

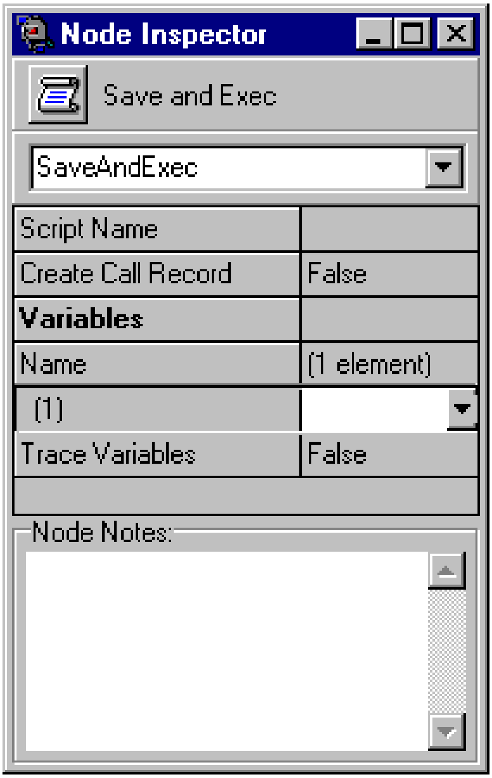


Figure 210. Node Inspector — Save and Exec Node

- **Script Name** — Type in a literal value, type in the name of a variable containing the name of the application, or from the pulldown menu select the variable containing the name of the application or script to execute.
- **Create Call Record** — When set to **True**, this attribute causes a call data record to be written to the appropriate database.

Variables

— Click the button to add a new variable element to be passed to the application you are calling.

— The # **elements** field displays how many variables have been selected.

— The **Name** column field displays the number assigned to each variable selected.

— From the corresponding pulldown menu, select the variable whose values you want to pass to the new application. All variables defined for the current application display on this pulldown menu.

- **Trace Variables** — When set to **True**, this attribute causes the application to trace the values of all variables defined here on the target voice response system. This allows you to use the **trace** command for these variable values on the target system to display the values that are passed to the new application.

The Load Variables Node

The Load Variables node loads variable data passed to it from another application into the current application (Figure 211). This node is typically used in conjunction with the Save and Exec node and is usually the first node in the current application.

Example of Use

ABC Automotive has two developers working on the script for its Inquiry Line, which is a very large application. To make it easier for the two developers to do their work, they split the script into two parts, with each developer working on part of it as a separate application. They design it in such a way that control passes from one part to the other at a certain point in the script, and certain variable values are passed at the same time. They use the Load Variables node to receive the variable data, such as account numbers, being passed from the first application into the second.

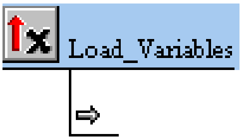


Figure 211. Load Variables Node

Behavior

- This node loads variable data from another application (Save and Exec node) and assigns the data to variables defined for the current application.
- The Load Variables node can receive 2 Kbytes of variable data as part of the node's operation.

Attributes

Attributes are assigned and edited using the Node Inspector (Figure 212).

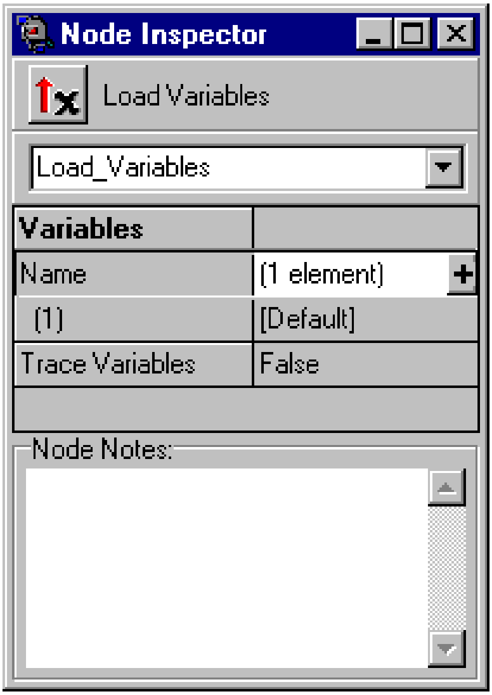


Figure 212. Node Inspector — Load Variables Node

Variables

— Click the ⊞ icon to add a variable to which to assign variable data.

— The # **elements** field displays how many variables have been selected.

— The **Name** column field displays the number assigned to each variable selected.

— Type in the name or from the corresponding pulldown menu select the variable to which you want to assign the variable data. All variables defined for the current application are displayed on this pulldown menu.

NOTE:
If you type in the name, it must be the name of a variable already defined for the application.

- **Trace Variables** — When set to **True**, this attribute causes the application to trace the values of all variables defined here on the target voice response system. This allows you to use the **trace** command for these variable values on the target system.

Using the SCI Tools

A

Overview

After you have finished creating, generating, and transferring your Voice@Work application to the target system, you must install the application on the target system before you can use it.

The INTUITY CONVERSANT system includes a set of Service Creation Installation (SCI) tools to help you accomplish this task.

Purpose

The purpose of this appendix is to provide you with information and procedures for using the SCI tools to install Voice@Work applications on your target system. Topics include:

- About the SCI Tools
- Preparing to Install an Application
- Installing the Application
- Assigning the Installed Application Using the Application Install Tab
- Removing an Application
- Importing or Removing a Professionally Recorded Speech File
- Transferring an Application from Floppy Disks

About the SCI Tools

SCI refers to the collection of utilities you use to install your application and the resources it needs on the target system after the files have been generated and transferred.

NOTE:
You must log in as root and perform the following procedures on the target voice response system.

There are four utilities included in the SCI tools:

- **sci** – Installs the application on the target system.
- **scrm** – Removes an application, speech files, or database tables from the target system.
- **scsp** – Imports or removes professionally recorded speech files to or from the target system.
- **scat** – Transfers an application from floppy disks to the target system.

Preparing to Install an Application

Before you can install an application on the target system, first do the following:

1. Create your application using Voice@Work. See Chapter 3, "Creating an Application Using Voice@Work", for the procedure.
2. Generate the source code using the Code Generation too. See Chapter 23, "The Code Generation Tool", for the procedure.
3. Transfer the application to the target system using the Application Transfer tool. See Chapter 24, "The Application Transfer Tool", for the procedure.

   NOTE:
   When you transfer the application, Voice@Work places it in the following directory on the target system:

   /lt/trans/sc/*<application_name>*/

   where <*application_name*> is the name of the application.

Installing the Application

Voice@Work offers the following options for installing applications on the target system:

- Installing the application from within Voice@Work, using the **Application Install** tab in the Code Generation / Application Transfer window
- Installing the application using the **sci** command at the target system or using telnet

You can also use the **Application Install** tab to assign the installed application to one or more channels on the target system.

Using the Application Install Tab

To install an application on the target system using the **Application Install** tab in the Code Generation / Application Transfer window, do the following:

1. Click the **Application Install** tab in the Code Generation / Application Transfer window.

   Voice@Work displays the Application Install pane (Figure 213).

   Note that the **Remote System:** field displays the name of target system on which the application is to be installed and the name and installation pathname for the application to be installed. For example, in the screen capture below, the application is to be installed on the target system named cbsdov6 in the directory path /lt/trans/sc/design02.

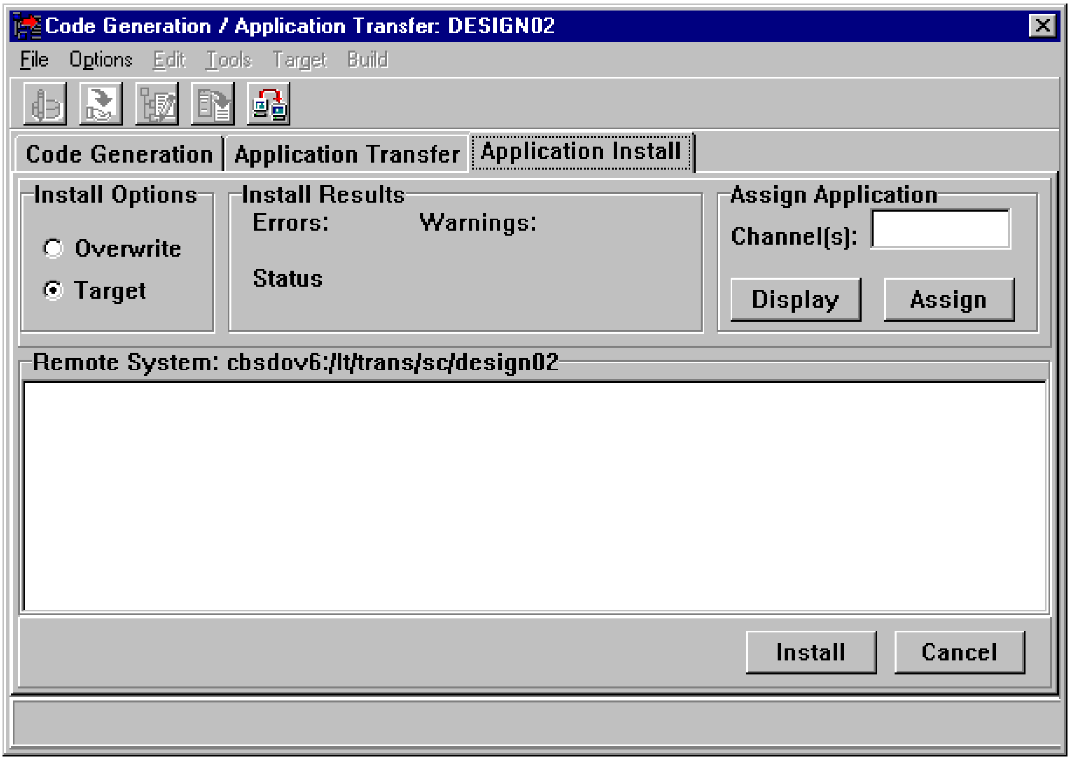


Figure 213. Application Install Tab Pane

2. From the **Install Options** box, select the appropriate installation option:
   - If you want to overwrite existing resources with matching names, such as database tables, host definitions, speech pools, and speech recognition grammars, click **Overwrite**.

     This is the same as using the **-O** option in the **sci** command (see "Using the sci Command")
   - If you do not want to overwrite existing resources with matching names but would rather use existing resources, click **Target**.

     This is the same as using the **-T** option in the **sci** command (see "Using the sci Command")
3. Click **Install**.

   During the installation, the **Application Install** pane displays the following information (Figure 214):

- The **Install Results** box displays the current status of the procedure, as well as the number of errors and warnings encountered during installation. An animated computer symbol also appears, to emphasize the fact that the installation is in progress.
- The **Remote System:** field displays general information about the installation, including information about errors and warnings. In case of error or warnings, you can scroll through the displayed text to get more details.

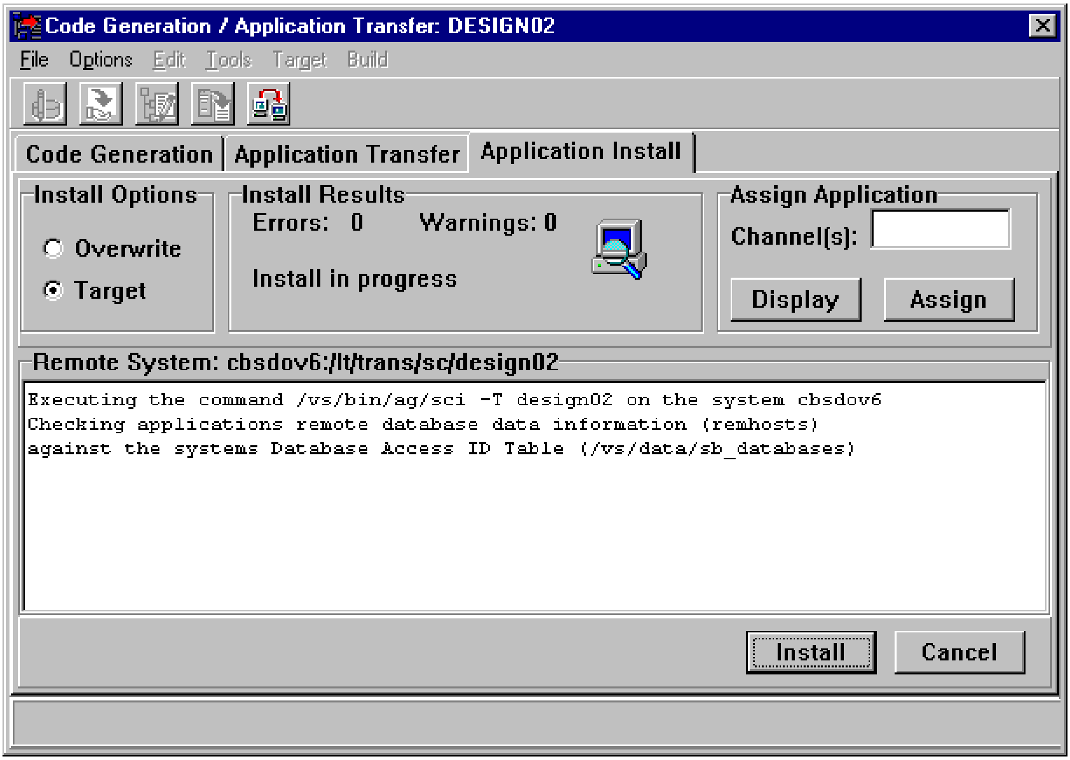


Figure 214. Appearance of the Application Install Tab during Installation

If the installation was successful, the **Application Install** pane appears similar to Figure 215.

NOTE:
The success or failure of the installation is based upon whether the TAS compiler can successfully compile the application on the target system. The Errors and Warnings count messages displayed in the Install Results box are based on messages from the entire sci process, including database table, host definition, and speech interface components. Thus, an application can install successfully yet still generate errors and warnings here.

For example, during application installation, a database table error might occur, which would cause the Errors count to be 1. The application as a whole, however, could still install successfully, in which case the status message would read Install successful. This means that the application is installed and usable, but actions using the affected database table would not work correctly.

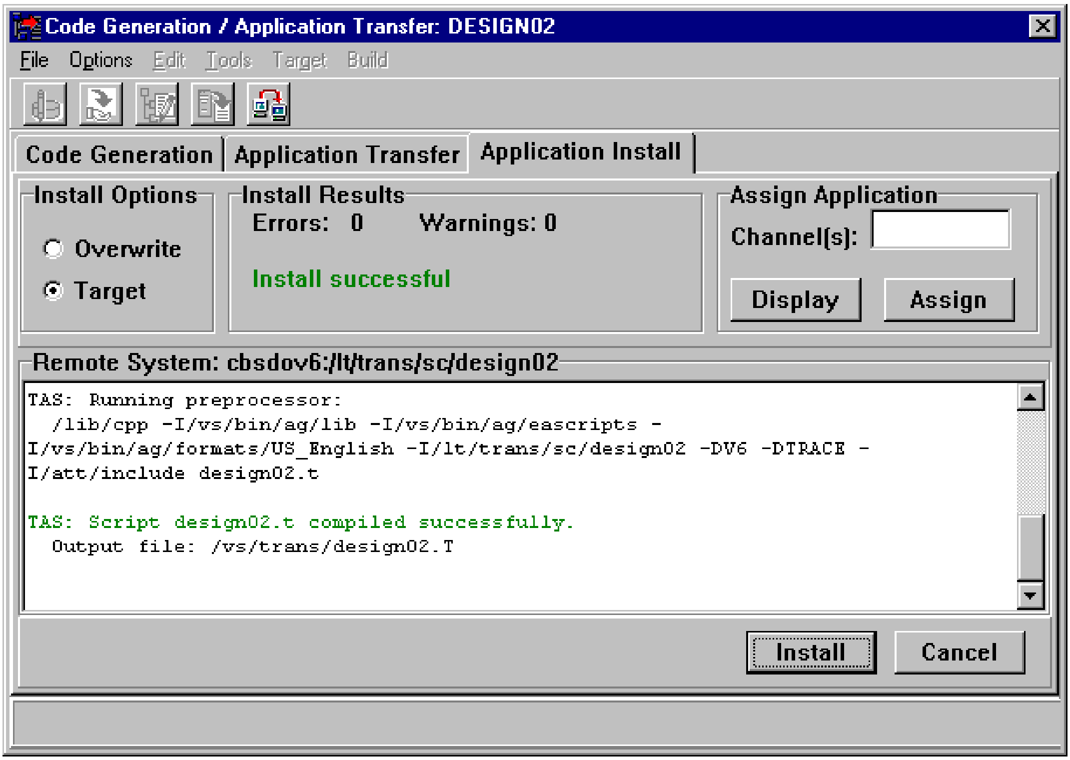


Figure 215. Appearance of Application Install Tab after Successful Installation

If the installation was *not* successful, the **Application Install** pane appears similar to Figure 216.

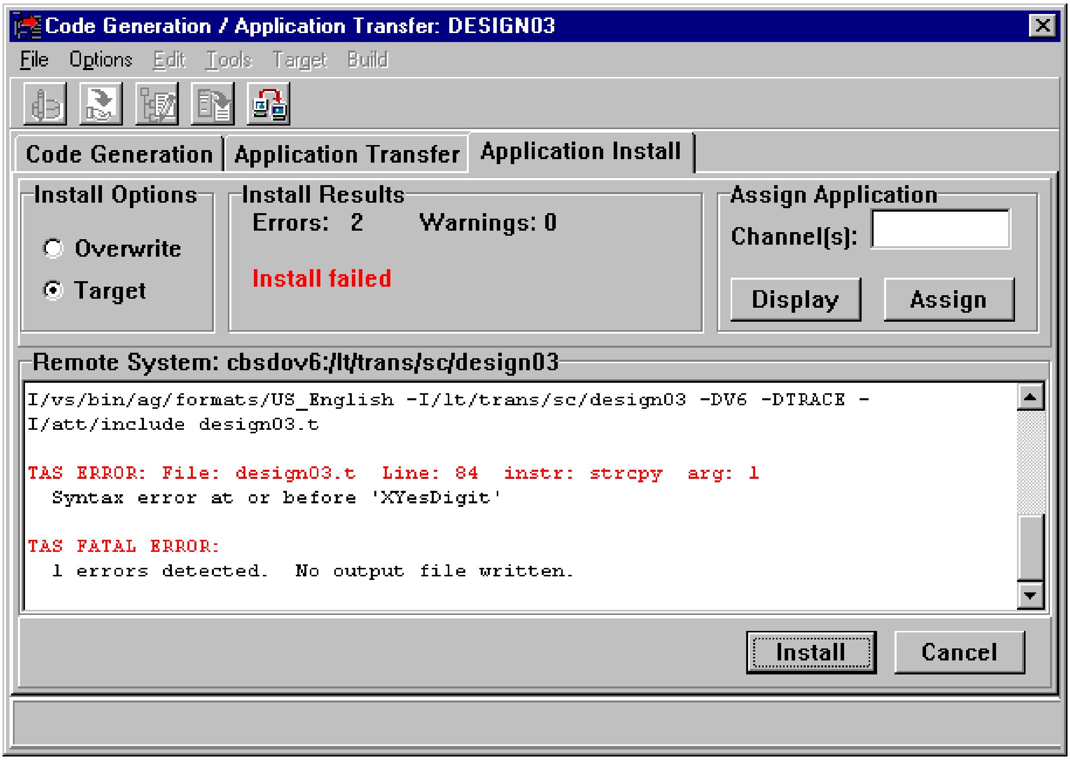


Figure 216. Appearance of Application Install Tab after Installation Failure

Using the sci Command

To install an application on the target system using the **sci** command, do the following:

1. Log in to the target system as root.
2. At the system prompt, enter some form of the following command:

   sci [-O | -T | -I] [-t] ***<application_name>***

Command Syntax

In the **sci** command above:

- **[-O | -T | -I]** are options. You may use only one of these three options at a time.

  -O overwrites resources with matching names (on the target system) without warning the user. Examples of resources affected by this option include database tables, host definitions, speech pools, and speech recognition grammars.

  -T uses existing resources, where name conflicts exist on the target system, without warning the user.

  -I prompts the user when there are file name conflicts, before either overwriting resources or using existing resources. This is the default if no option is entered as part of the command.

- **[-t]** is an option. If used, the option turns off tracing for Trace Variables nodes.
- ***<application_name>*** is the name of the application.

Examples

To install an application named **acct_bal** using the resources that exist on the target system, and turn off tracing for Trace Variables nodes, enter:

sci -T -t acct_bal

To install an application named **acct_bal, o**verwrite the resources on the target system, and leave tracing on for Trace Variables nodes, enter:

sci -O acct_bal

The install.log file

During the **sci** installation process, a file named **install.log** is produced. The purpose of the **install.log** file is to provide developers and support personnel with information about the installation and compilation of a Voice@Work application. Although the **install.log** file displays error messages from ORACLE, TAS, and other INTUITY CONVERSANT software, it does not provide any explanation about the cause or solution to the error. For information about causes of and solutions for errors, see the error message documention for your system.

Restarting the Database DIP

If you are installing database tables as part of the application installation and using the interactive mode (**-I** option), the **sci** tool asks if you want to restart the database data interface process (DIP).

If all of the tables are completely new tables, or if you have not changed the table schema for any of the tables, select **No**.

If you have changed the table schema on any existing table since the application using it has been run, and tables have been accessed in that application, select **Yes**.

CAUTION:

*If you select **Yes** to restart the database DIP, then any calls that attempt to access a database for that application while the database DIP is being restarted will fail. For this reason, you should plan to restart the database DIP at a low call volume time, if possible.*

Assigning the Installed Application Using the Application Install Tab

Once you have installed the application on the target system, you can also assign the application to one or more channels on the target system, using the **Application Install** tab of the Code Generation / Application Transfer window.

To assign the application using the **Application Install** tab, do the following:

1. (Optional, but recommended) Check to see if the channel or channels to which you want to assign the application are available for assignment on the target system. To check channels, do the following:
   a. Enter the number or numbers of the channels you want to check in the **Channel(s):** field.

      Use a hyphen to separate the first and last numbers of a series (for example, **4-6**). Use commas to separate multiple entries (for example, **1,4-6**).

      To check all channels, leave this field blank.
   b. Click **Display**.

      The **Remote System:** field displays information about the selected channel or channels.
2. In the **Channel(s):** field, enter the number or numbers of any channels to which you want to assign the application.
3. Click **Assign**.

   The system assigns the application to the designated channels and the **Remote System:** field displays a message confirming the assignment.

Removing an Application

The following section details removing an application using the SCI tools.

Procedure

To remove resources from the target system using the SCI tools, do the following:

1. Log in to the target system as root.
2. At the system prompt, enter some form of the following command:

   scrm -[d | i | s | t | a] ***<application_name>***

Command Syntax

- **scrm** (Service Creation ReMove) is the command to remove resources from the target system.
- **-[d | i | s | t | a]** are options. You may use any combination of these options.

  -d – Removes local database tables

  -i – Removes installed files

  -s – Removes speech files

  -t – Removes transaction files

  -a – Removes all files associated with the application
- **<application_name>** represents the name of the application.

Examples

To remove all database and speech files from an application named **acct_bal**, enter:

scrm -ds acct_bal

To remove all files associated with an application named **acct_bal**, enter:

scrm -a acct_bal

Importing or Removing a Professionally Recorded Speech File

scsp is the command used to import the recorded speech file to or remove it from the target system.

Lucent Technologies recommends that you use professionally recorded speech in your Voice@Work applications. Such professionally recorded speech can be obtained from Lucent Technologies or from independent sources.

More information about obtaining professionally recorded speech for your Voice@Work applications can be found in the *INTUITY™ CONVERSANT® System Speech Development, Processing, and Recognition* book for your system.

NOTE:
Before you can import a professionally recorded speech file to the target system, you must first:

- Create the application using Voice@Work (see Chapter 3, "Creating an Application Using Voice@Work"). Voice@Work creates a list of all the required speech phrases during the Code Generation process.
- Obtain professionally recorded speech files on floppy disk.
- Generate the source code using the Code Generation tool (see Chapter 23, "The Code Generation Tool").
- Transfer the application to the target system using the Application Transfer tool (see Chapter 24, "The Application Transfer Tool").

Procedure

To import (or remove) a recorded speech file to (from) the target system using the SCI tools, do the following:

1. Log in to the target system as root.
2. If you are importing a recorded speech file, insert the floppy disk containing the file into the floppy disk drive.
3. At the system prompt, enter some form of the following command:

   scsp [-o] [-r] ***<application_name> <phrase_list_filename>***

   NOTE:
   The speech file is copied to the following directory:

   /lt/trans/sc/*<application_name>*/pro_speech/*<phrase_list_ filename>*

Command Syntax

- **scsp** is the basic command used to import or remove the recorded speech file to or from the target system.
- **[-o]** is an option. If used, the option overwrites files with matching file name without warning the user.

  When transferring an application with a name matching an application that already exists on the target system, Lucent Technologies recommends that you first remove the old application files using the **scrm** command on the target system.

  For example, suppose you had an application named **abc_auto** on the target system, and you later developed another, totally different application to replace it. In the earlier **abc_auto** application, you used certain FlexWord word lists and database tables. In the newer version, you chose not to use the same FlexWord word lists and database tables.

  When you transfer the new version of **abc_auto** to the target system, only the files with matching file names are replaced. The old FlexWord lists and database tables, however, are still there. This can cause conflicts when using the newer version.

  To avoid this problem, first remove the old application (or the appropriate application files) from the target voice-response system, using the **scrm** command. Then you can transfer and install the new application without having to worry about file conflicts.
- **[-r]** is an option. If used, the option removes the recorded speech file from the application.
- ***<application_name>*** is the name of the application.
- ***<phrase_list_filename>*** is the file name of the recorded speech's phrase list file. This is required because an application may have multiple phrase list files.

  NOTE:
  You must define a phrase list file for your recorded speech and generate it using the Code Generation too. See Chapter 23, "The Code Generation Tool". This assigns it a **.pl** extension. The floppy disk with the professionally recorded speech must also contain a file called **Backup.pl** that contains the phrase list for the speech. This phrase list should be the same as the one you generated in your application.

  Also, the actual recorded speech must be in files formatted with the phrase number and a **.phr** extension (for example, **1000.phr**). A file called **Sequence** must contain the number of floppies in the set, with the last floppy containing the word "LAST". After all the phrases are copied from the floppy disk, if there is a **.pl** file in **/speech/talk** that matches what was copied in, the speech will be loaded into the talkfile.

Examples

To import a recorded speech file named **greeting** into an application named **acct_bal**, without being prompted if the target system already has speech file with that name, enter:

scsp -o acct_bal greeting

To remove a recorded speech file named **greeting** from an application named **acct_bal**, enter:

scsp -r acct_bal greeting

Transferring an Application from Floppy Disks

Voice@Work allows you to transfer your finished applications to the target voice response system using floppy disks. See Chapter 24, ‘‘The Application Transfer Tool’’, for the procedures.

Voice@Work Icons

B

Overview

As Voice@Work is a graphically oriented tool, many of its commands, tools, and features are available using icons. In Voice@Work, these icons are arranged on three palettes, each with its own function and set of icons.

- The Toolbar contains icons for menus and commands found on the menu bar.
- The Node Palette contains the icons for all node objects.
- The External Functions Palette contains the icons for any external functions defined for the application.

In addition, several of the development tools and editors have their own sets of icons.

Purpose

The purpose of this appendix is to act as a visual reference guide for the user to identify icons used in Voice@Work, to give keyboard and menu equivalents for the icons, and to give a brief description of each icon's functions. Icons are grouped according to their appearance in Voice@Work and arranged in the following sections:

- Toolbar Icons
- Node Palette Icons
- External Function Palette Icons
- Miscellaneous Other Icons

Toolbar Icons

The toolbar icons are all related to the menus on the menu bar of the main screen. The icons are grouped according to menu.

File Menu Icons

Table 103 describes the icons for the File menu. See Chapter 5, “The File Menu” for additional information about these icons.

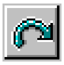
Table 103. File Menu Icons

Icon	Related Menu Option	Action
	New	Creates a new application.
	Open	Opens an existing application.
	Close	Closes the current application.
	Save	Saves the current application to disk.
	Delete	Deletes the current application.
	Print	Prints the application; see “The Print Reports Window” in Chapter 5, “The File Menu”.
	Exit	Closes any open application and exits Lucent Voice@Work.

Edit Menu Icons

Table 104 describes the icons for the Edit menu. See Chapter 6, “The Edit Menu” for additional information about these icons.

Table 104. Edit Menu Icons

Icon	Related Menu Option	Action
	Undo	Reverses (undoes) the last action or series of actions since the last Save action that can be undone.
	Redo	Reverses (redoes) any actions performed by the “Undo” option, one at a time.
	Cut	Removes selected node(s) from the call flow and places them on the clipboard.
	Copy	Places a copy of selected node(s) on the clipboard.
	Paste	Pastes the nodes from the clipboard onto any open branch.
	Find...	Searches across all call flows in the application to find: ■ Text strings used in component names ■ Locations where specified items are used ■ Unused items ■ Text strings

View Menu Icons

Table 105 describes the icons for the View menu. See Chapter 7, ''The View Menu'' for additional information about these icons.

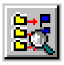
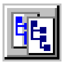
Table 105. View Menu Icons

Icon	Related Menu Option	Action
	Globals	Opens the Globals Manager and all associated resource manager; see Chapter 13, ''The Globals Manager''.
	Variables	Opens the Variables Manager and the Globals Manager, if it is docked there; see Chapter 14, ''Working with Variables''.
	Phrases	Opens the Phrases Manager and the Globals Manager, if it is docked there; see Chapter 18, ''Working with Phrases''
	Prompts	Opens the Prompts Manager and the Globals Manager, if it is docked there; see Chapter 19, ''Working with Prompts''.
	Tables	Opens the Tables Manager and the Globals Manager, if it is docked there; see Chapter 20, ''Working with Tables''.
	External Functions	Opens the External Functions Manager and the Globals Manager, if it is docked there; see Chapter 16, ''Working with External Functions''
	FlexWords	Opens the FlexWords Manager and the Globals Manager, if it is docked there; see Chapter 15, ''Working with FlexWord Speech Recognition''.
	Hosts	Opens the Hosts Manager and the Globals Manager, if it is docked there; see Chapter 17, ''Working with Hosts''.

Tools Menu Icons

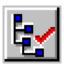
Table 106 describes the icons for the Tools menu. See Chapter 9, "The Tools Menu" for additional information about these icons.

Table 106. Tools Menu Icons

Icon	Related Menu Option	Action
	Application Resources Tool	Opens the Application Resources window, which allows you to: — See what resources are in use with the current application — Import additional resources into your application as needed — See what resources in your application are being shared
	Call Flow Inventory Tool	Opens the Call Flows window, which allows you to: — See what call flows exist for your application — See where each call flow is used within the application, and what other call flows (if any) are used within each call flow — Locate a specific call flow within an application — Provide an alternate means of dragging a call flow to the Design Pad
	Code Generation/ Application Transfer Tool	Opens the Code Generation/ Application Transfer window, which allows you to: — Generate the code necessary to run the application on the target system — Transfer the generated code to the target system

Continued on next page

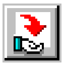
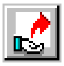
Table 106. Tools Menu Icons — *Continued*

Icon	Related Menu Option	Action
	Events String Editor	Opens the Events Editor, which allows you to: — Determine which variables will be treated as Call Data Events on the target system — Display both a list of all the variables defined for the application and a list of the variables selected to be treated as Call Data Events
	Node Inspector Tool	Opens the Node Inspector window, which allows you to: — Edit node attributes easily and quickly — Display different information and editing options for each node, depending on the node's predetermined attributes
	Preference Editor	Opens the Preference Editor, which allows you to set global preferences for using Voice@Work.
	Simulation Tool	Runs the Simulation tool, which tests your application in a simulated environment, which is often helpful in finding problems within the application
	Verify Design Tool	Runs the Verify Design tool, which automatically searches through each call flow of your application to check your application design for possible errors or omissions, and displays the results in the Verification Results window.

Sharing Menu Icons

Table 107 describes the icons for the Sharing menu. See Chapter 10, "The Sharing Menu" for additional information about these icons.

Table 107. Sharing Menu Icons

Icon	Related Menu Option	Action
	Refresh from Shared Resources	Allows you to refresh your shared-in resources, to ensure that you are working with the most recent version of the shared resource.
	Update Shared Resources	Allows you to update your shared-out resources, to make the most recent version of the shared resource available to any developer who might be using it.

Help Menu Icon

Table 108 describes the icon for the Help menu. See Chapter 11, "The Help Menu" for additional information about this icon.

Table 108. Help Menu Icon

Icon	Related Menu Option	Action
	Help Contents...	Opens the online Help (to the contents topic)

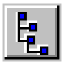
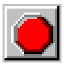
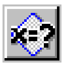
Node Palette Icons

Each Node Palette icon represents a different type of node object you can use in your application. These icons are grouped according to general node functionality. See Chapter 12, “Palettes”.

Control Node Icons

Table 109 describes the Control Node icons.

Table 109. Control Node Icons

Icon	Node Name	Action
	Call Flow node	Creates a new call flow or encapsulates a set of nodes.
	Return node	Stops the execution of the call flow that contains it. Disconnects the call, if it has not already been disconnected.
	Set and Test node	Allows you to manipulate and evaluate a wide range of variable expressions and branch conditions.
	Time Branch node	Directs the call based on time of day, day of the week, or specific date.
	Asynchronous Event node	Defines the subroutine to be executed if an asynchronous event, such as a caller hanging up, is detected.

Telephony Node Icons

Table 110 describes the Telephony Node icons.

Table 110. Telephony Node Icons

Icon	Node Name	Action
	Answer Call node	Answers the incoming call; typically one of the first nodes of any application.
	Transfer Call node	Transfers a call to another number.
	Disconnect Call node	Disconnects the incoming call and executes the next node in the call flow (if one exists).

Miscellaneous Node Icons

Table 111 describes the Miscellaneous Node icons.

Table 111. Miscellaneous Node Icons

Icon	Node Name	Action
	Inline Code node	Allows you to write TAS code directly in your Voice@Work application.
	Trace Variables node	Allows you to trace values of selected variables on the target system; useful primarily as a debugging tool after installing your application on the target system.

Continued on next page

Table 111. Miscellaneous Node Icons — *Continued*

Icon	Node Name	Action
	Execute Script node	Calls another Voice@Work application, leaves the current application, and does not return; typically used in conjunction with the Load Variables node.
	Load Variables node	Loads variable data passed to it from another Voice@Work application into the current application; typically used in conjunction with the Execute Script node and is usually the first node in the current application.

IVR (Interactive Voice-Response) Node Icons

Table 112 describes the IVR Node icons.

Table 112. IVR Node Icons

Icon	Node Name	Action
	Announcement node	Plays a simple announcement to the caller.
	Menu node	Presents a list of options from which the caller can choose.

Continued on next page

Table 112. IVR Node Icons — *Continued*

Icon	Node Name	Action
	Prompt and Collect node	Prompts a caller for input such as account or credit card numbers.
	Automenu node	Creates a dynamic menu of choices derived from a list of previously retrieved database records.
	Voice Capture node	Records a caller's spoken response.

Database Node Icons

Table 113 describes the Database Node icons.

Table 113. Database Node Icons

Icon	Node Name	Action
	Query Table node	Retrieves records from a database table, based upon the specified selection criteria.
	Update Table node	Changes data within a specified database table.
	Add Record node	Adds a record to a database table during application execution.
	Delete Record node	Deletes a record from a database table during application execution.

Host Node Icons

Table 114 describes the Host Node icons.

Table 114. Host Node Icons

Icon	Node Name	Action
	Host Connection node	Determines the status of the connection with a host machine in the network.
	Host Send node	Sends data to a remote host machine in the network.
	Host Send/Receive node	Sends data to *and* receives data from a remote host machine in the network.

External Function Palette Icons

The External Functions Palette contains the icons for any external functions you may use in your application. These external functions can be imported using the Application Resources tool (see Chapter 21, "The Application Resources Tool" for additional information) or created within your Voice@Work application. The External Functions Palette is blank until you import or create an external function for your application.

Similar to the standard palette configuration options used for both the Toolbar and the Node Palette, external functions can be grouped together into user-defined categories and represented by customized bitmap images on the External Functions Palette (for information about configuring palettes, see Chapter 12, "Palettes").

Voice@Work provides an icon for each of the standard external functions packaged with the software. To view these icons and descriptions of each of the standard external functions, see Appendix C, "Standard External Functions".

Voice@Work also provides one standard bitmap image you can use with your own custom external functions (Figure 217).

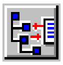


Figure 217. External Functions Bitmap Image

To use other (customized) bitmaps, you can:

- Create your own, using any graphics software that supports the *.bmp file format.
- Use an existing bitmap image.

(For more information about creating and using customized bitmap images, see "Defining the External Function's Appearance" in "The External Function Editor" in Chapter 16, "Working with External Functions").

For more information about these capabilities and about External Functions in general, see Chapter 16, "Working with External Functions".

Miscellaneous Other Icons

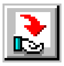
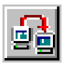
In addition to the main palettes, several of the tools in Voice@Work have their own palettes and corresponding icons. The tools that have their own icon palettes are:

- The Code Generation/Application Transfer Tool
- The Events String Editor
- The Simulation Tool

Code Generation/Application Transfer Tool Icons

Table 115 describes the Code Generation/Application Transfer tool icons. See Chapter 23, "The Code Generation Tool" and Chapter 24, "The Application Transfer Tool" for additional information on these icons.

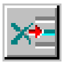
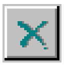
Table 115. Code Generation/Application Transfer Tool Icons

Icon	Icon/Function Name	Action
	Database Table Provisioning	(Available only on the Code Generation tab) Used to set table options for Code Generation and export during Application Transfer.
	Refresh from Shared Resources	(Available only on the Code Generation tab) Allows you to refresh your shared-in resources, to ensure that you are working with the most recent version of the shared resource.
	Display Files on Remote System or Floppy	(Available only on the Application Transfer tab) Allows you to view the files in the destination area before transferring your application.
	External Function Source Code	(Available only on the Code Generation tab) Allows you to create an external function for the target system, by writing or editing TAS code from within Voice@Work
	Telnet	(Available only on the Application Transfer and Application Install tabs) Allows you to establish and use a telnet connection without having to exit Voice@Work

Events String Editor Icons

Table 116 describes the Events String Editor icons. See Chapter 25, "The Events String Editor" for additional information about these icons.

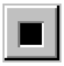
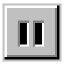
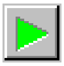
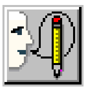
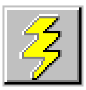
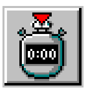
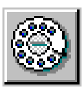
Table 116. Events String Editor Icons

Icon	Icon/Function Name	Action
	Append Variable	Adds a variable to the end of the Selected Variables list.
	Add Variable Before...	Inserts a variable *before* a variable on the Selected Variables list.
	Add Variable After...	Inserts a variable *after* a variable on the Selected Variables list.
	Delete Variable	Deselects a single variable from the Selected Variables list.
	Delete All Variables	Deselects (clears) *all* variables from the Selected Variables list.
	Move Variable Up	Moves a variable *up* on the Selected Variables list.
	Move Variable Down	Moves a variable *down* on the Selected Variables list.

Simulation Tool Icons

Table 117 describes the Simulation tool icons. See Chapter 26, "The Node Inspector" for additional information about these icons.

Table 117. Simulation Tool Icons

Icon	Icon/Function Name	Action
	Stop Simulation	Stops a simulation.
	Pause Simulation	Pauses a simulation.
	Run Simulation	Runs a simulation from the currently active node or from where it was stopped or paused.
	Reset Variables	Resets the variables to their defaults; for example, to run the simulation again.
	Step-Through Simulation	Allows you to step through a simulation one node at a time
	Respond with Speech	Allows you to simulate spoken responses for FlexWord and WholeWord speech recognition features.
	Generate Asynchronous Event	Used to simulate an asynchronous event, such as a customer hanging up in the middle of a node.
	Force Timeout	Enables you to skip the countdown, or delay, that is programmed into many nodes, and go to the next node.
	Dial Pulse Recognition	Enables you to simulate a Dial Pulse Recognition (DPR) response from a caller

Standard External Functions

C

Overview

Voice@Work includes, as part of the standard package, a variety of external functions that you can use in developing your applications. These are known as standard external functions (or sometimes, *system* external functions).

For more information about external functions in general, see Chapter 16, "Working with External Functions".

Purpose

The purpose of this appendix is to provide brief descriptions of the external functions (nodes) supplied as part of the Voice@Work package. These standard external functions are grouped according to general functionality. External function groups include:

- ASAI External Functions
- FAX External Functions
- Transfer External Functions
- Phrase External Functions
- GeoTel External Functions
- UCID External Functions
- Nabnasset External Functions
- Speech Features External Functions
- PRI External Functions

- Form Filler External Functions
- ANI External Functions
- Proxy Text-to-Speech (PTTS) External Functions
- General External Functions

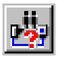
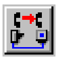
More information about most of these external functions can be found in the *INTUITY™ CONVERSANT® System Application Development with Script Builder* documentation for your system.

ASAI External Functions

Table 118 describes the external functions that can be used in conjunction with the ASAI feature on the target system.

NOTE:
The external functions in this group will only work on the target system if you have purchased the ASAI option.

Table 118. ASAI External Functions

Name	Icon	Description
A_Callinfo		Accesses call information, such as ANI and DNIS, obtained from ASAI for a call on a VIS line
A_Event		Retrieves information related to a call being monitored by an ASAI domain
A_RouteSel		Used to send an ASAI route message to the PBX (used in conjunction with A_Event – see above)
A_Trans		Used by voice scripts running on VIS lines to transfer a call

FAX External Functions

Table 119 describes the external functions that can be used in conjunction with the FAX features on the target system.

Table 119. FAX External Functions

Name	Icon	Description
FAX_Queue		Queues a fax for transmission
FAX_CovrPage		Combines two faxes into a single fax, with the second fax starting on the same page as the first (same as BT_FAX_CovrPage)
FAX_Send		Directs the system to transmit the specified graphic image or text file to a designated fax telephone number. Can be used to schedule a fax transmission (same as BT_FAX_Send)
Exec_UNIX		Directs the system to execute a UnixWare command or shell script (same as BT_Exec_UNIX)
Concat5		Concatenates (joins together) up to five character strings to make a resulting single string (same as BT_Concat5)
FAX_Get		Directs the system to receive a fax on the current call and to place it into a designated directory and file (same as BT_FAX_Get)
FAX_CNG		Turns FAX CNG tone detection off or on (same as BT_FAX_CNG)
BT_FAX_Get		Directs the system to receive a fax on the current call and to place it into a designated directory and file (same as FAX_Get)
BT_FAX_CovrPage		Combines two faxes into a single fax, with the second fax starting on the same page as the first (same as FAX_CovrPage)

Continued on next page

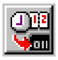
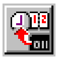
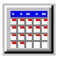
Table 119. FAX External Functions — *Continued*

Name	Icon	Description
BT_FAX_Current		Directs the system to transmit on the current call the specified graphical image or text files to a designated fax telephone number
BT_FAX_Combine		Combines up to three files into a single fax, with each file starting on a separate (new) page
BT_FAX_CNG		Turns FAX CNG tone detection off or on (same as FAX_CNG)
BT_Concat5		Concatenates, or joins together, up to five character strings to make a resulting single string (same as Concat5)
BT_Exec_UNIX		Directs the system to execute a UnixWare command or shell script (same as Exec_UNIX)
BT_FAX_Send		Directs the system to transmit the specified graphic image or text file to a designated fax telephone number. Can be used to schedule a fax transmission (same as FAX_Send)

Date/Time External Functions

Table 120 describes the external functions that can be used in conjunction with the date and time features on the target system.

Table 120. Date/Time External Functions

Name	Icon	Description
datetime_u		Converts a date field and a time field to UNIX date&time, computing the internal UNIX time
u_datetime		Converts UNIX date&time to a date field and a time field by converting the internal UNIX clock value
getday		Converts UNIX time to a day of the week and returns a numeric field which corresponds to the appropriate day (1 = Sunday, 2 = Monday, and so on)
mon_year		Speaks the date as a month and, optionally, the year as a two-digit or four-digit year

Phrase External Functions

Table 121 describes the external functions that can be used in conjunction with phrase files on the target system.

Table 121. Phrase External Functions

Name	Icon	Description
Background		Turns off or on a pre-recorded phrase that plays in the background of the call
Msg_Code		Records a phrase, using by default the phrase number or NX number to identify the phrase

Continued on next page

Table 121. Phrase External Functions — *Continued*

Name	Icon	Description
Msg_Code_Tag		Records a phrase, using by default the phrase tag to identify the phrase
Msg_Delete		Deletes a phrase, using by default the phrase number or NX number to identify the phrase
Msg_Delete_Tag		Deletes a phrase, using by default the phrase tag to identify the phrase
pack_phrNX		Converts a talkfile number and a phrase number into a combined number to be spoken in NX format
unpack_phrNX		Reverses the pack_phrNX function, separating the combined number into a talkfile number and a phrase number

Transfer External Functions

Table 122 describes the external functions that can be used in conjunction with call transfer features on the target system.

Table 122. Transfer External Functions

Name	Icon	Description
Call_Bridge		Places an outbound call; if answered, bridges new and current channels
ixfer		Allows a script to place an outbound call to a user-defined telephone number; maintain the connection while the caller interacts with the person on the other channel; and when the called person hangs up, continue with the next action step
transfera		Flashes (waits for call progress tones), dials the number, and flashes again
transferb		Flashes (waits for call progress tones), dials the number, waits for call progress tones, and flashes again
Complete		Performs a non-blind transfer call complete routine; connects the caller to a third party; should only be used to connect the caller to the attendant (third party) after an answer is detected or during ringing in a non-blind transfer action
Reconnect		Reconnects the caller to the script; should only be used to reconnect to the caller after a No Answer, Busy, Reorder (Fast Busy), or Error condition was encountered in a non-blind transfer action

GeoTel External Functions

Table 123 describes the external functions that can be used in conjunction with GeoTel features on the target system.

Table 123. GeoTel External Functions

Name	Icon	Description
icrTrans		Obtains an extension from a GeoTel Intelligent Call Router (ICR) in order to transfer a call

UCID External Functions

Table 124 describes the external functions that can be used in conjunction with Universal Caller Identification(UCID) features on the target system.

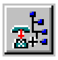
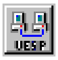
Table 124. UCID External Functions

Name	Icon	Description
set_uui		Sets User-to-User Information (UUI) for an outbound call
get_uui		Extracts the UUI from an incoming call
get_ucid		Extracts the UCID from a call
clear_ucid		Clears the UCID from a call

Nabnasset External Functions

Table 125 describes the external functions that can be used in conjunction with UCID features on the target system.

Table 125. Nabnasset External Functions

Name	Icon	Description
Trap		Handles a caller hangup (requires that the label “Hungup” be added to the script)
Untrap		Performs additional actions after the call is done; requires that the label “Quit” be added to the script
vesp_dip		Accesses the Voice Enhanced Services Platform (VESP) dip

Speech Features External Functions

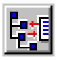
Table 126 describes the external functions that can be used in conjunction with speech recognition features on the target system.

Table 126. Speech Features External Functions

Name	Icon	Description
tts_file		Speaks the text from a specified ASCII text file using Text-to-Speech (TTS)
DPR_Disable		Enables or disables the DPR recognizer

Continued on next page

Table 126. Speech Features External Functions — *Continued*

Name	Icon	Description
SP_Allocate		Allocates speech recognition resources
SR_Allocate		Allocates speech recognition resources
SR_Prompt		Turn on or off “barge-in” or “speech recognition during prompt”

PRI External Functions

Table 127 describes the external functions that can be used in conjunction with Primary Rate Interface (PRI) features on the target system.

Table 127. Speech Features External Functions

Name	Icon	Description
ISDN_billing		Retrieves the billing number (also known as ANI or CPN) for the incoming PRI call
ISDN_service		Sets the service type for outgoing PRI calls

Form Filler External Functions

Table 128 describes the external functions that can be used in conjunction with Form Filler features on the target system.

Table 128. Form Filler External Functions

Name	Icon	Description
FF_Code		Codes a message for the Form Filler application
FF_Store		Stores the phrases for the Form Filler application

ANI External Functions

Table 129 describes the external functions that can be used in conjunction with ANI (Automatic Number Identification) features on the target system.

Table 129. ANI External Functions

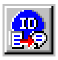
Name	Icon	Description
Attr_ANI		Allows the system to request a billing number for incoming calls on a call-by-call basis

Proxy Text-to-Speech (PTTS) External Functions

Table 130 describes the external functions that can be used in conjunction with Proxy Text-to-Speech (PTTS) features on the target system.

For more information about PTTS, see the *INTUITY™ CONVERSANT® Proxy Text-to-Speech (PTTS)* documentation for your system.

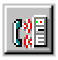
Table 130. Proxy Text-to-Speech (PTTS) External Functions

Name	Icon	Description
proxyparams		Defines the parameters for the PTTS speaking voice, including gender, rate of speech, volume, pitch, and intonation
proxytext		Converts either a file or a buffer of text to speech
proxytts_on		Used to toggle between PTTS and TTS
proxylang		Used to select a particular language within an application – should only be used when using multiple languages and the language needs to change within the application
proxyvoice		Used to select a particular voice tag within PTTS – should only be used when an application requires two different voice tags having the same language and gender

General External Functions

Table 131 describes the general external functions that can be used in Voice@Work applications.

Table 131. General External Functions

Name	Icon	Description
Conv_data		Used with the Converse Vector feature on the DEFINITY to pass data. When a call is sent to the INTUITY CONVERSANT, the switch can send data across; when the call ends the INTUITY CONVERSANT can send data back to the switch.
getarg		Extracts an argument after Execute is completed; used in conjunction with the Execute external function
Execute		Executes another script on the current channel **NOTE:** The Execute external function is similar but not identical to the Save and Exec node. The differences lie in the way the two pass arguments and variables to the new application. When calling a Script Builder application, you *must* use the Execute external function because Script Builder does not have Load Variables node functionality or any equivalent action. For Voice@Work applications, you can use either the Save and Exec node or the Execute external function.
Make_Call		Places or dials a call
sleep		“Sleeps” or pauses, the execution of the application for the specified number of seconds

Manipulating Variables

D

Overview

Variables in Voice@Work refer to application-specific information holders. They may be used to collect and hold data, act as counters, or make use of system information. Variables are grouped together and accessed through the Variables Manager (see Chapter 14, “Working with Variables”).

Once created, variables can be used as you would use them in any other application development environment. When set up properly, you can perform arithmetic operations on them, convert them to other types, and compare them with other variables.

Purpose

The purpose of this appendix is to provide additional information about using and manipulating variables in Voice@Work applications. Topics include:

- Variables and Arithmetic Operations
- Converting Variable Values
- Variable Comparisons

Variables and Arithmetic Operations

The following is a summary of the rules governing arithmetic operations between any two variables of the same or different classes:

- Addition and subtraction can only be performed on variables of the same class (Currency, Number, and Real variable classes only).

  The exceptions to this rule are addition and subtraction involving Date and Time variables. A Number variable can be added to or subtracted from a Date or Time variable (see below for details).

- Multiplication and division must involve either a Number or a Real variable (see below for details).
- Character — No arithmetic operations can be performed on any Character variable.

Rules for Currency Variables

- If *both* variables are Currency variables, they can be added or subtracted.
- Multiplication or division can be performed between a Currency variable and a Number or a Real variable, provided the destination is a Currency variable.

Rules for Date Variables

- Date values are stored internally in the **yyyymmdd** format.
- A Number variable may be added to or subtracted from a Date variable. For example, adding 32 to a Date value of 19980112 (January 12, 1998), results in a value of 19980213 (February 13, 1998).
- When adding or subtracting a Number variable from a Date variable, the Number variable is treated as days. For example, to add 2 months to the date March 12, 1998, you must populate a Number variable with the value 61 (19 days left in March + 30 days for April + 12 days for May = 61 days).
- If you populate the Number variable with a value of 2, the resulting Date value will only be March 14, 1998 (19980314).

Rules for Number Variables

- The maximum and minimum values that a Number variable can have are 2,147,483,647 and -2,147,483,648, respectively. Values outside this range will produce errors and unpredictable results.
- If both variables are Number variables, they can be added, subtracted, multiplied, or divided.

- Multiplication or division can only be performed between a Number and a Currency variable if the destination is a Currency variable.
- A Number and a Real variable may only be multiplied or divided (not added or subtracted). If the destination is a Number variable, the resulting value is always an integer. Decimal places are ignored and discarded. For example, 3 * 12.15 results in a value of 36.45, which is stored as 36. 3 + 12.15 is not a valid operation in Voice@Work.

Rules for Real Variables

- If both variables are Real variables, they can be added, subtracted, multiplied, or divided.
- Multiplication or division can only be performed between a Real and a Currency variable if the destination is a Currency variable.
- A Real and a Number variable can only be multiplied or divided (not added or subtracted). For example, 3 * 12.15 results in a value of 36.45. 3 + 12.15 is not a valid operation in Voice@Work.
- When setting the field length for a Real variable used as a destination for arithmetic operations, remember the following:
  - One place must be reserved for the decimal point and one for the minus symbol. Make sure you allow enough places to include possible decimal points and minus symbols in any computation results you may get.
  - If you do not allow for enough digits in the destination variable field length, the MathDIP returns only the digits that will fit in the allotted space.

Rules for Time Variables

- Time values are stored internally in a 24-hour **hhmmss** format.
- A Number variable can be added to or subtracted from a Time variable. For example, adding 600 to a Time variable of 205500 (8:55 P.M.) results in a value of 210500 (9:05 P.M.).
- When adding or subtracting a Number value from a Time value, the Number value is treated as seconds. For example, to add 10 minutes to the time 12:31, you must populate a Number variable with the value 600 (10 minutes times 60 seconds = 600 seconds). If you populated the Number variable with a value of 10, the resulting Time display is unchanged, because the minutes portion of the time value remains unchanged.

NOTE:
The seconds portion of the Time variable is not currently supported in the Spoken As formats in Voice@Work.

Converting Variable Values

When data of one field type is assigned to another field type, Voice@Work converts the value to the type of the field receiving the value. How Voice@Work handles this depends on the type of data in the original field.

NOTE:
If a conversion cannot be done, Voice@Work populates the target variable with the previous value if there is one. Otherwise, Voice@Work uses the default value for the target variable type.

For more details about converting a different variable type, see the appropriate section for that variable below)

Converting Character Variable Values

Voice@Work converts character variable values to other types according to the rules in Table 132.

Table 132. Converting Character Variables

Type of Field Converting to	Rules and Comments	Examples
Character	If the length of the field receiving the value is less than the original, Voice@Work truncates the string to the length of the new field.	If the string “This string is 34 characters long.” is converted to a field limited to 24 characters, the result is: “This string is 34 charac”
Currency	Voice@Work strips out all except number, dollar sign ($), and leading minus symbol (-) characters and converts the resulting string to a number.	The string “12-31-97” is converted to: 123197 The string “This is a string.” cannot be converted, so the Currency default is used: 0.00
Date	Voice@Work converts the original value to the date format **yyyymmdd**.	The string “12-31-97” is converted to: 19971231 The string “800-555-1536” cannot be converted, so the Date default is used: 19980101

Continued on next page

Table 132. Converting Character Variables — *Continued*

Type of Field Converting to	Rules and Comments	Examples
Number	If the string being converted contains only numeric characters, Voice@Work converts the string to an integer value. Voice@Work ignores any leading spaces. If there is a leading minus sign (-), Voice@Work assigns the integer a negative value. All non-numeric characters except the minus symbol (-) are stripped out.	The string “-1235ba-6” is converted to: -12356 The string “This is a string.” cannot be converted, so the Number default is used: 0
Real	Voice@Work strips out all except number and leading negative sign (-) characters and converts the resulting string to a number.	The string “12-31-97” is converted to: 123197 The string “This is a string.” cannot be converted, so the Real default is used: 0.0
Time	Voice@Work converts the original value to the time format **hhmmss**.	The string “12-31-57” is converted to: 123157 The string “12-31-97” cannot be converted, so the Time default is used: 000000

Converting Currency Variable Values

Voice@Work converts currency variables to other types according to the rules in Table 133.

Table 133. Converting Currency Variables

Type of Field Converting to	Rules and Comments	Examples
Character	Since the currency variable is really a special type of character variable, currency values convert intact.	The value $123.45 is converted to: “$123.45”
Currency	If the length of the field receiving the value is less than the original, Voice@Work truncates the value to the length of the new field. Dollar signs are stripped out before conversion.	The value $123.45 is converted to: 123.45 If the variable $12345.67 is converted to a field limited to 7 characters, the result is: 12345.6
Date	Voice@Work cannot convert currency values to date values because of the leading dollar sign. The previous value or default date value is used.	The variable $123.45 cannot be converted, so the Date default is used: 19980101
Number	Voice@Work cannot convert currency values to number values because of the leading dollar sign. The previous value or default number value is used.	The variable $123.45 cannot be converted, so the Number default is used: 0
Real	Voice@Work strips out all except number characters, decimal points, and leading minus symbols and converts the resulting string to a number.	The variable $123.45 is converted to: 123.45
Time	Voice@Work cannot convert currency values to time values because of the leading dollar sign. The previous value or default time value is used.	The variable $123.45 cannot be converted, so the Time default is used: 000000

Converting Date Variable Values

Voice@Work converts date variables to other types according to the rules in Table 134.

NOTE:
Voice@Work stores date variables in the **yyyymmdd** format and converts them from there.

Table 134. Converting Date Variables

Type of Field Converting to	Rules and Comments	Examples
Character	Voice@Work converts the date variable value to a character string using the **yyyymmdd** format.	The date 12-31-97 is converted to: "19971231"
Currency	Voice@Work converts the date variable value to a number.	The date 12-31-97 is converted to: 19971231
Date	Voice@Work converts date variables intact.	The date 12-31-97 is converted to: 19971231
Number	Since date variables as stored are composed of eight numerical characters, Voice@Work converts date variables to numbers easily.	The date 12-31-97 is converted to: 19971231
Real	Voice@Work converts the date variable to a number.	The date 12-31-97 is converted to: 19971231
Time	Because of the format in which date variable values are stored (**yyyymmdd**), Voice@Work is unable to convert dates to time variables.	The date 12-31-97 cannot be converted, so the Time default is used: 000000

Converting Number Variable Values

Voice@Work converts number variables to other types according to the rules in Table 135.

Table 135. Converting Number Variables

Type of Field Converting to	Rules and Comments	Example
Character	Voice@Work converts the number variable value to a character string.	The number 123197 is converted to: “123197” The number 13579246 is converted to: “13579246”
Currency	Voice@Work converts the number to a number.	The number 123197 is converted to: 123197
Date	Voice@Work tries to interpret the number variable value to see if it can convert it to a date format. If it can, it converts it to the **yyyymmdd** format.	The number 123197 is converted to: 19971231 The number 13579246 cannot be converted, so the Date default is used: 19980101
Number	Number variables are converted “as is”.	The number 123197 is converted to: 123197
Real	Voice@Work converts the number variable to a number.	The number 123197 is converted to: 123197
Time	Voice@Work tries to interpret the number variable value to see if it can convert it to a time format. If it can, it converts it to the **hhmmss** format.	The number 123157 is converted to: 123157 The number 13579246 cannot be converted, so the Time default is used: 000000

Converting Real Variable Values

Voice@Work converts real variables to other types according to the rules in Table 136.

Table 136. Converting Real Variables

Type of Field Converting to	Rules and Comments	Examples
Character	Since the real variable is really a special type of character variable, real values convert intact.	The value 123.4567 is converted to: “123.4567”
Currency	Voice@Work strips out all except number characters, leading minus symbols, and decimal points and converts the resulting value to a number.	The variable 123.4567 is converted to: 1234567
Date	Voice@Work strips out all except number characters and converts the result to a date value using the **yyyymmdd** format.	The variable 12.3197 is converted to: 19971231 The variable 123.4567 cannot be converted, so the Date default is used: 19980101
Number	Voice@Work strips out all but number characters and leading minus symbols.	The variable 123.4567 is converted to: 1234567
Real	If the length of the field receiving the value is less than the original, Voice@Work truncates the value to the length of the new field.	If the variable 12345.6789 is converted to a field limited to 7 characters, the result is: 12345.6 (the decimal point counts as one character)
Time	Voice@Work strips out all except number characters and converts the result to a time value using the **hhmmss** format.	The variable 12.3157 is converted to: 123157 The variable 123.4567 cannot be converted, so the Time default is used: 000000

Converting Time Variable Values

Voice@Work converts time variables to other types according to the rules in Table 137.

NOTE:
Time variables are stored using the **hhmmss** format. All conversions are made using those values.

Table 137. Converting Time Variables

Type of Field Converting to	Rules and Comments	Examples
Character	Since the time variable is really a special type of character variable, time values convert intact.	The value 12:34 (PM) is converted to: “123400”
Currency	Voice@Work converts the time variable value to a number.	The variable 12:34 (PM) is converted to: 123400
Date	Voice@Work converts the result to a date value using the **yyyymmdd** format.	The variable 12:19:57 (PM) is converted to: 19571219 The variable 12:34 (PM) cannot be converted, so the Date default is used: 19980101
Number	Voice@Work converts the time variable value to an integer.	The variable 12:34 (PM) is converted to: 123400
Real	Voice@Work converts the time variable value to a number.	The variable 12:34 (PM) is converted to: 123400
Time	Voice@Work converts the time variable value intact.	The variable 12:34 (PM) is converted to: 123400

Variable Comparisons

When comparing variables, Voice@Work uses the following general rules:

- Only variables of the same class can be compared in operations involving variables. For example, a character variable can only be compared with another character variable, a number variable can only be compared with another number variable, and so on.
- You can use either a variable or a literal value when performing variable comparison operations. However, if you use a literal value, it must take a form appropriate for the class of variable you want to compare it with. For example, you cannot compare a literal value of “abc” with a number variable, because “abc” is a character string.
- When comparing character variables, case is taken into account according to standard rules for character variable comparison.

Comparison Operands

When comparisons are performed between two values, the operation returns a value of “true” or “false”, depending on the type of comparison being performed. Table 138 describes the actions of operands used in variable comparisons.

Table 138. Variable Comparison Operands

Operand	Comparison Action
=	Equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
!=	Not equal to

Comparisons with Character Values

You can perform comparisons between two character variables or between literal values and character variables. You can also make comparisons using a null character value.

Voice@Work compares character values using a *lexicographical* comparison. This means that each character of the two fields starting from the left is compared one at a time, until one field has no more characters or a mismatch occurs.

If no mismatch occurs and the fields are of the same length, they are considered equal. If a field has fewer characters and all previous characters have matched, the longer field is considered greater. If a mismatch occurs, the field with the character having the greater value at the mismatch is considered the greater variable.

Characters are compared according to the standard character ordering scheme for programming languages. Thus, the following comparisons would all return a value of “true”:

““ < any other string

“12” < “2”

“0123” != “123”

“ABC” < “AD”

“000” < “999”

“999” < “AAA”

“AAA” < “AAB”

“AAA” < “AAAA”

“AAA” < “A.A.”

“AAA” < “aaa”

NOTE:
Two character variables can contain numbers that are numerically equal but not lexicographically equal, as in the third example above. If you want a numeric comparison, you must first convert both variables to number variables, and then do the comparison.

Comparisons with Other Values

Voice@Work performs straightforward arithmetic comparisons on all other variable classes.

Converting Graphical Designer Applications

E

Overview

Voice@Work can open and automatically convert applications created using Lucent Technologies Graphical Designer™ version 1.6.1 or above. To convert an application created using an earlier version of Graphical Designer, you must first upgrade it to version 1.6.1 or above.

Purpose

The purpose of this appendix is to provide information about preparing, converting, and opening applications created in Graphical Designer version 1.6.1 or later using Voice@Work. Topics include:

- Preparing for the Conversion
- The Conversion Process
- After Converting the Application

Preparing for the Conversion

Before you attempt to convert a version 1.6.1 application, you should do the following:

- During the conversion process, Voice@Work looks for certain directories and files from the version 1.6.1 application. Therefore, you must have Graphical Designer version 1.6.1 installed on the same machine you are installing Voice@Work. Any applications developed in version 1.6.1 that you want to convert must also reside on the same machine.

- Validate the version 1.6.1 application by running the **Audit** and **Verify** commands. If **Audit** detects any issues, see the Release Notes for Graphical Designer Version 1.6.1 to determine how to resolve them before converting in Voice@Work.
- Make sure that any custom or modified external functions used in your application have the correct definitions present in the Graphical Designer version 1.6.1 External Function definitions. If not, you must remove the external functions from your application or add the definition to the 1.6.1 version.
- Voice@Work does not allow any name collisions, even between different types, such as call flows and external functions. Therefore, make sure that no two components in the 1.6.1 application have the same name. If there are two or more components with the same name, you must rename all but one before attempting to convert and open the application in Voice@Work. If you do not do this, Voice@Work will notify you of the conflicts and abort the conversion process.
- Make sure the version 1.6.1 application has no database field names longer than 25 characters.
- **Back up your 1.6.1 application by saving a copy with a different name or saving a copy of it to a different location.**
- Read the Release Notes for Voice@Work for any issues pertaining to application conversions that may have been discovered after this documentation was produced.

The Conversion Process

During the conversion process itself, keep in mind the following:

- Double-quote marks in phrases (“) are converted to single-quote marks (‘).
- If you have any language (grammar) files other than US English in the application to be converted, Voice@Work converts them to the equivalent US English language files during the conversion process.
- The external functions **ctv_config**, **say_esp**, **VM_Getmsg**, **VM_Mail**, **VM_Sendmsg**, **VM_Subinfo**, **sayesp**, and **FAX_Response** were considered standard external functions on CONVERSANT Version 4.x systems, but are not considered standard external functions on INTUITY CONVERSANT Version 6 and above systems. If you used any of these functions in your application, they are copied to a local directory for custom external functions and thereafter treated as custom external functions.
- Voice@Work reports any errors it encounters. If it can recover from a particular error, Voice@Work continues with the conversion. If not, Voice@Work notifies you that it cannot continue and aborts the conversion process without modifying the original application.

- In rare cases, a corrupted 1.6.1 application may cause access violations during or immediately after the conversion process. If this happens, or if you get a “corrupted application” message or any other serious error messages, call your Technical Service Center for assistance.
- Large applications may take several minutes to convert and open. Do not try to do anything with the system while the application is being converted and opened.

After Converting the Application

After the version 1.6.1 application has been successfully converted and opens in Voice@Work, you should do the following:

- Validate the conversion by running the Simulation and Verify Design tools See Chapter 28, ‘‘The Simulation Tool’’ and Chapter 29, ‘‘The Verify Design Tool’’, for the procedures.
- Save it after you have verified that it works properly.

If you are not editing the application but simply converting it, you must still do the following before you can use it:

- Regenerate the code using the Code Generation tool. See Chapter 23, ‘‘The Code Generation Tool’’, for the procedures.
- Transfer the application using the Application Transfer tool. See Chapter 24, ‘‘The Application Transfer Tool’’, for the procedures.
- Reinstall the application on the target system using the SCI tool. See Appendix A, ‘‘Using the SCI Tools’’, for the procedures.

System Variable Descriptions

F

Overview

System variables are variables that are predefined and included as part of the Voice@Work package. Each system variable provided as part of the Voice@Work package has its own attributes, some of which can be changed.

Purpose

The purpose of this appendix is to provide complete descriptions of each system variable provided as part of the Voice@Work package, including default attributes. System variables include:

- ANI (Automatic Number Identification)
- Attempts
- Channel
- CI_RECOG
- CollectedDigits
- Date
- Default
- DNIS (Dialed Number Identification Service)
- Errno
- FindBest
- NoDigit

- PivotYear
- RecordCounter
- RecordsFound
- Time
- Tries
- YesDigit

ANI (Automatic Number Identification)

As a variable, ANI stores the phone number of the calling party. This variable is only available and functional if you have purchased the Adjunct/Switch Application Interface (ASAI) optional feature package.

When spoken by the INTUITY CONVERSANT system, the collected value is spoken as **digits**.

Default attributes are listed in Table 139:

Table 139. ANI System Variable Default Attributes

Attribute	Default Value
Type	character
Spoken As	digits*
Speak Using	current default*
Phrase Table	not applicable
Input As	char
Length	10
Default	none*

* Indicates a value that you can edit or reset.

Attempts

As a variable, Attempts stores the number of times the caller tried to respond to a prompt.

When spoken by the INTUITY CONVERSANT system, the collected value is spoken as a number.

Default attributes are listed in Table 140:

Table 140. Attempts System Variable Default Attributes

Attribute	Default Value
Type	number
Spoken As	number*
Speak Using	current default*
Phrase Table	not applicable
Input As	#
Length	10
Default	1*

* Indicates a value that you can edit or reset.

Channel

As a variable, Channel stores the channel number on which the current call is being handled.

When spoken by the INTUITY CONVERSANT system, the collected value is spoken as a number.

Default attributes are listed in Table 141:

Table 141. Channel System Variable Default Attributes

Attribute	Default Value
Type	number
Spoken As	number*
Speak Using	current default*
Phrase Table	not applicable
Input As	#
Length	10
Default	1*

* Indicates a value that you can edit or reset.

CI_RECOG

As a variable, CI_RECOG is a number representing the recognizer used to identify a caller response. Possible values include the following:

0 – Touch Tone

1 – FlexWord

2 – WholeWord

3 – Dial Pulse

NOTE:
This variable is set by the system and does not need to be changed by the user.

Default attributes are listed in Table 142.

Table 142. CI_RECOG System Variable Default Attributes

Attribute	Default Value
Type	number
Spoken As	number
Speak Using	current default
Phrase Table	not applicable
Input As	#
Length	10
Default	nil

CollectedDigits

As a variable, CollectedDigits stores the value of the digit(s) pressed by the caller in responding to a prompt.

When spoken by the Intuity CONVERSANT system, the collected values are spoken as digits.

Default attributes are listed in Table 143:

Table 143. CollectedDigits System Variable Default Attributes

Attribute	Default Value
Type	character
Spoken As	digits*
Speak Using	current default*
Phrase Table	not applicable
Input As	char
Length	68
Default	none*

* Indicates a value that you can edit or reset.

Date

As a variable, Date stores the date on which the call was received, in *yyyymmdd* format.

When spoken by the INTUITY CONVERSANT system, the collected value is spoken as *M d, y*. For example, **19980102** would be spoken as "January second (pause) nineteen ninety eight" unless you change the **Spoken As** field value.

Default attributes are listed in Table 144:

Table 144. Date System Variable Default Attributes

Attribute	Default Value
Type	date
Spoken As	M d, y*
Speak Using	current default*
Phrase Table	not applicable
Input As	mmddyy
Length	8
Default	date the application was created*

* Indicates a value that you can edit or reset.

Default

As a variable, Default stores caller input as a character string.

When spoken by the INTUITY CONVERSANT system, the collected values are spoken as character-based strings.

Default attributes are listed in Table 145:

Table 145. Default System Variable Default Attributes

Attribute	Default Value
Type	character
Spoken As	char*
Speak Using	current default*
Phrase Table	not applicable
Input As	char
Length	24
Default	none*

* Indicates a value that you can edit or reset.

DNIS (Dialed Number Identification Service)

As a variable, DNIS stores the number the calling party dialed. This variable is only available and functional if you have purchased the Adjunct/Switch Application Interface (ASAI) optional feature package.

When spoken by the INTUITY CONVERSANT system, the collected value is spoken as digits.

Default attributes are listed in Table 146:

Table 146. DNIS System Variable Default Attributes

Attribute	Default Value
Type	character
Spoken As	digits*
Speak Using	current default*
Phrase Table	not applicable
Input As	char
Length	10
Default	none*

* Indicates a value that you can edit or reset.

Errno

As a variable, Errno stores a number. Errno can be assigned to return error codes or any other number value.

When spoken by the INTUITY CONVERSANT system, the collected value is spoken as a number.

Default attributes are listed in Table 147.

Table 147. Errno System Variable Default Attributes

Attribute	Default Value
Type	number
Spoken As	number*
Speak Using	current default*
Phrase Table	not applicable
Input As	#
Length	10
Default	0*

* Indicates a value that you can edit or reset.

FindBest

As a variable, FindBest is used to help improve recognition accuracy in applications that use DPR or speech recognition.

For more information about this system variable, see one of the following:

- *Intuity™ CONVERSANT® System Version 7.0 Application Development with Script Builder*, 585-313-206.
- Appendix F, "Recognition Post-Processing," in *Intuity™ CONVERSANT® System Version 7.0 Speech Development, Processing, and Recognitiion*, 585-313-201.

Default attributes are listed in Table 148.

Table 148. FindBest System Variable Default Attributes

Attribute	Default Value
Type	character
Spoken As	char*
Speak Using	current default*
Phrase Table	not applicable
Input As	char
Length	68
Default	none*

* Indicates a value that you can edit or reset.

NoDigit

As a variable, NoDigit is the value of the number for a caller to press signifying a "no" response.

When spoken by the INTUITY CONVERSANT system, the value is spoken as a digit.

Default attributes are listed in Table 149.

Table 149. NoDigit System Variable Default Attributes

Attribute	Default Value
Type	character
Spoken As	digits*
Speak Using	current default*
Phrase Table	not applicable
Input As	char
Length	1
Default	2*

* Indicates a value that you can edit or reset.

PivotYear

PivotYear is used to designate a year which the INTUITY CONVERSANT can use to determine how to assign character variable date strings to a particular century. Values that are less than the value of PivotYear are interpreted to be in the 21st century (20xx). Values that are greater than the value of PivotYear are interpreted to be in the 20th century (19xx).

This variable is used *only* when converting a character variable in the **mmddyy** format to a date variable.

If the PivotYear value is set to "0" (the default), then the century is determined by the current date (century) setting for the INTUITY CONVERSANT. In other words, before January 1, 2000, dates are assigned to the 20th century (for example, 1956). After January 1, 2000, dates are assigned to the 21st century (for example, 2056).

For example, suppose you received the character string, **012156**, from a host screen, and you want to store and use it as a date variable. You have set the value of PivotYear to "60". Using a Set and Test node, you assign the character variable from the host to a date variable. Based upon the PivotYear setting, the date is thus interpreted as **20560121** (January 21, 2056).

Default attributes are listed in Table 150.

Table 150. PivotYear System Variable Default Attributes

Attribute	Default Value
Type	number
Spoken As	number*
Speak Using	current default*
Phrase Table	not applicable
Input As	#
Length	10
Default	0*

* Indicates a value that you can edit or reset.

RecordCounter

As a variable, RecordCounter is used as a counter to track the number of records that match selected criteria. Each number in the final set represents a different record of all those found.

When spoken by the INTUITY CONVERSANT system, the collected value is spoken as a number.

Default attributes are listed in Table 151.

Table 151. RecordCounter System Variable Default Attributes

Attribute	Default Value
Type	number
Spoken As	number*
Speak Using	current default*
Phrase Table	not applicable

Continued on next page

Table 151. RecordCounter System Variable Default Attributes — *Continued*

Attribute	Default Value
Input As	#
Length	10
Default	0*

* Indicates a value that you can edit or reset.

RecordsFound

As a variable, RecordsFound increments by 1 each time a record is found. The final value of the number is the total number of records found.

When spoken by the INTUITY CONVERSANT system, the collected value is spoken as a number.

Default attributes are listed in Table 152.

Table 152. RecordsFound System Variable Default Attributes

Attribute	Default Value
Type	number
Spoken As	number*
Speak Using	current default*
Phrase Table	not applicable
Input As	#
Length	10
Default	0*

* Indicates a value that you can edit or reset.

Time

As a variable, Time stores the time at which the call was received, in hh24mmss format.

When spoken by the INTUITY CONVERSANT system, the collected value is spoken as *hhmm(AP)M*. For example, **155227** would be spoken as "three fifty two P M" unless you change the **Spoken As** field value.

Default attributes are listed in Table 153.

Table 153. Time System Variable Default Attributes

Attribute	Default Value
Type	time
Spoken As	hhmm(AP)M*
Speak Using	current default*
Phrase Table	not applicable
Input As	hhmm(AP)M
Length	6
Default	Time the application was created*

* Indicates a value that you can edit or reset.

Tries

As a variable, Tries represents the number of times to allow a caller to attempt a response to a prompt.

When spoken by the INTUITY CONVERSANT system, the collected value is spoken as a number.

Default attributes are listed in Table 154.

Table 154. Tries System Variable Default Attributes

Attribute	Default Value
Type	number
Spoken As	number*
Speak Using	current default*
Phrase Table	not applicable
Input As	#
Length	10
Default	3*

* Indicates a value that you can edit or reset.

YesDigit

As a variable, YesDigit is the value of the number for a caller to press signifying a "yes" response.

When spoken by the INTUITY CONVERSANT system, the value is spoken as a digit.

Default attributes are listed in Table 155:

Table 155. YesDigit System Variable Default Attributes

Attribute	Default Value
Type	character
Spoken As	digits*
Speak Using	current default*
Phrase Table	not applicable
Input As	char
Length	1
Default	1*

* Indicates a value that you can edit or reset.

Building a Sample Application

G

Overview

This appendix provides a step-by-step example of how to build a simple application using Voice@Work.

Purpose

The purpose of this appendix is to provide instructions for building a simple application step-by-step in Voice@Work. Topics include:

- Before You Get Started
- Creating the Basic Call Flow
- Editing the Nodes
- Debugging the Application
- Generating the Code
- Transferring the Application
- Installing the Application on the Target System

Before You Get Started

Before building your first application, you must do the following:

- Plan and design the application.
- Set up and start Voice@Work.
- Create a new application.

Planning and Designing Your Application

In designing your own applications, a very important first step is carefully planning and designing what you want your application to do (see Chapter 3, "Creating an Application Using Voice@Work").

For this sample application, the planning and designing have been done for you. The following simple application which you will build does the following:

1. Answers the phone
2. Plays a greeting announcement to the caller
3. Transfers the call to another number and announces the transfer to the caller
4. Disconnects and terminates the call — in other words, hangs up

Setting Up and Starting Voice@Work

The instructions for building the application in this appendix assume that you have already done the following:

- Installed Voice@Work successfully (see Chapter 1, "Introduction to Voice@Work")
- Started Voice@Work (see "Opening Voice@Work the First Time" in Chapter 2, "Getting Started with Voice@Work")
- Set global preferences (see "Setting Global Preferences" in Chapter 2, "Getting Started with Voice@Work" or Chapter 27, "The Preference Editor")
- Configured your workspace (see Chapter 2, "Getting Started with Voice@Work")

Creating a New Application

Before you can begin building your first application, you must create it. To create a new application, do the following:

1. Click the  icon on the Toolbar.

   Voice@Work displays the New Voice@Work Application window (Figure 218). Notice that Voice@Work suggests a default name, in the example below, **design01**.

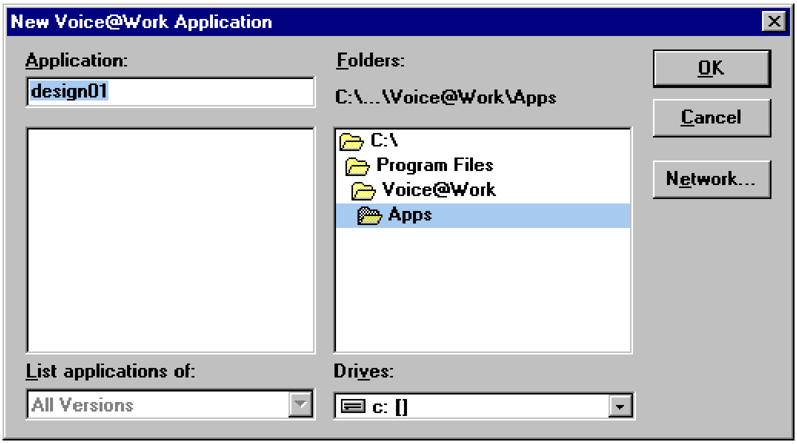


Figure 218. New Voice@Work Application Window

2. Type **my_first** in the **Application:** field.
3. Click **OK**.

   Voice@Work creates a new application called **my_first** and clears the design pad.

You are now ready to begin developing your application.

Creating the Basic Call Flow

The most basic task in developing a Voice@Work application is building one or more call flows by dragging the appropriate nodes onto the design pad.

To build the call flow for this application, do the following:

1. Add the Answer Call node to the call flow by doing the following:

   c. Locate the Answer Call node icon  on the Node Palette and drag it onto the design pad.

   Voice@Work displays the node naming dialog box (Figure 219). If the **Automatically name nodes** checkbox in the Preference Editor is selected, Voice@Work suggests a name for the node.

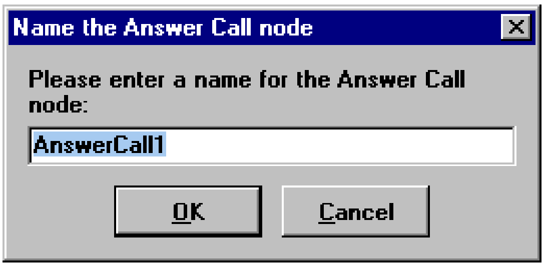


Figure 219. Node Naming Dialog Box

   d. Type **Answer_Phone** in the name field.

   NOTE:
   When the node naming dialog box opens, the suggested name is highlighted. To replace it, simply type the new name. If the suggested name is *not* highlighted, you must first double-click it to highlight it, and then replace it by typing in the new name.

   e. Click **OK**.

   Voice@Work adds the new node to the design pad display and opens the Node Inspector, if it is not already open (Figure 220).

   NOTE:
   Many nodes require further definition after they have been placed in the call flow. You can choose to complete these definitions either when you place the node in the call flow or later on, when you are done placing all the nodes in the call flow. This is a matter of personal style. For the purpose of this example, you will define the nodes after placing all nodes in the call flow.

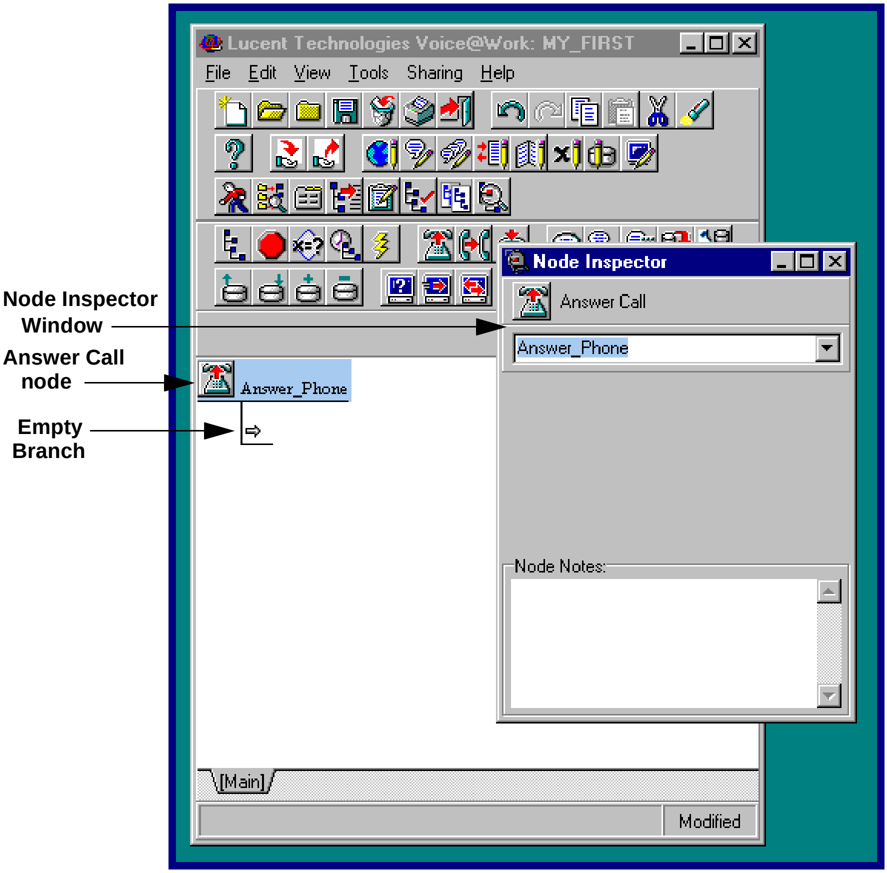


Figure 220. Call Flow with Answer Call Node Added

2. Add an Announcement node by doing the following:

   a. Locate the Announcement node icon  on the Node Palette and drag it onto the design pad.

   NOTE:
   You may need to move the Node Inspector window to get to the Announcement or other nodes.

b. Place your cursor directly over the empty branch of the Answer Call node (Figure 220) and release the mouse button.

NOTE:
If you do not place the cursor directly over the empty branch to release the mouse button, the new node is not attached to the call flow. It is added to the design pad display, but it is disconnected from the call flow.

Voice@Work displays the node naming dialog box.

c. Type **Say_Hello** in the name field.

d. Click **OK**.

Voice@Work adds the new node to the call flow and changes the display in the Node Inspector window (Figure 221).

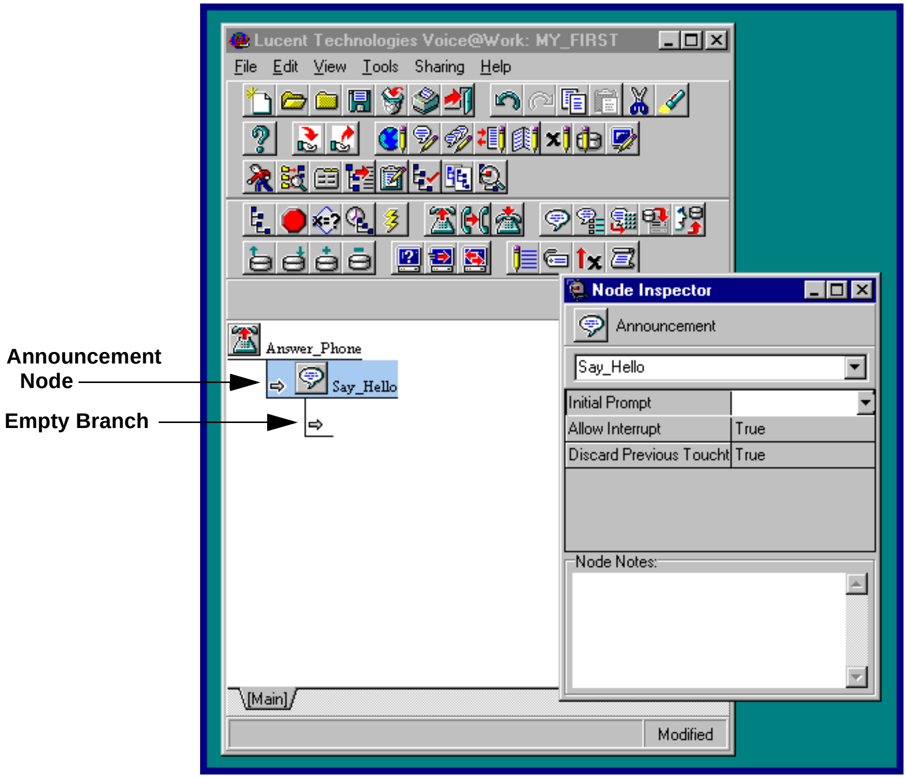


Figure 221. Call Flow with Announcement Node Added

Note carefully the appearance of the call flow with this new node added. If you did not drop the Announcement node onto the empty branch of the Answer Call node, the Announcement node is not connected. Figure 222 shows what the call flow looks like if the nodes are not connected.

NOTE:
If you accidentally drop the node in the wrong place, you can drag-and-drop it again to place it correctly.

Also, if the nodes in a call flow are not connected, the application works only as far as the disconnected node. Any nodes after the disconnected node do not execute. This is true of the application both in Simulation mode in Voice@Work and in operation on the target voice response system.

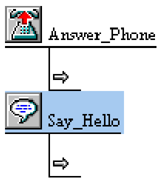


Figure 222. Example of Disconnected Nodes

3. Add a Transfer Call node by doing the following:
   a. Locate the Transfer Call node icon  on the Node Palette and drag it onto the design pad.
   b. Place your cursor directly over the empty branch of the Announcement node (Figure 221) and release the mouse button.

      NOTE:
      If you do not place the cursor directly over the empty branch to release the mouse button, the new node is not attached to the call flow. It is added to the design pad display, but it is disconnected from the call flow.

      Voice@Work displays the node naming dialog box.
   c. Type **Transfer_Call** in the name field.
   d. Click **OK**.

      Voice@Work adds the new node to the call flow and changes the display in the Node Inspector window (Figure 223).

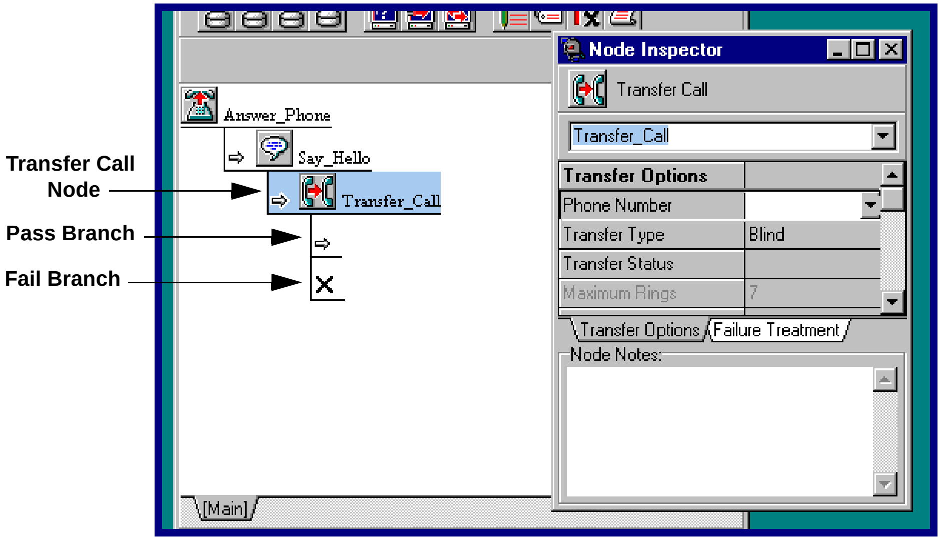


Figure 223. Call Flow with Transfer Call Node Added

Notice that this node has two empty branches. The first branch is the Pass branch, which is the branch that is taken if the node is executed successfully. In this case, it means that the call is transferred successfully.

The second branch, with the X symbol, is the Fail branch. This is the branch that is taken if, for any reason, the call cannot be successfully transferred.

4. Add a Disconnect Call node by doing the following:

   a. Locate the Disconnect Call node icon  on the Node Palette and drag it onto the design pad.

   b. Place your cursor directly over the empty Pass branch of the Transfer Call node (Figure 223) and release the mouse button.

      Voice@Work displays the node naming dialog box.

   c. Type **Hangup** in the name field.

   d. Click **<u>O</u>K**.

      Voice@Work adds the new node to the call flow and changes the display in the Node Inspector window (Figure 224).

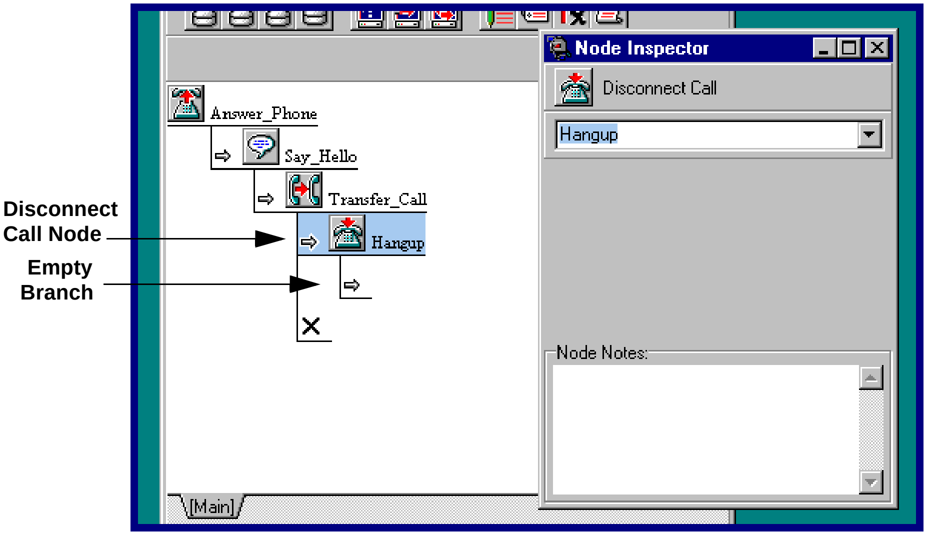
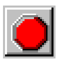


Figure 224. Call Flow with Disconnect Node Added

5. Add a Return node by doing the following:

NOTE:
The purpose of the Return node is not as obvious as some of the other nodes. In the [Main] call flow, the Return node terminates the application script and allows the target system to reset itself for another caller. For more information about the Return node, see the Return node description in Chapter 31, "Node Descriptions".

a. Locate the Return node icon  on the Node Palette and drag it onto the design pad.

b. Place your cursor directly over the empty branch of the Disconnect Call node (Figure 224) and release the mouse button.

   Voice@Work displays the node naming dialog box.

c. Type **Quit** in the name field.

d. Click **OK**.

   Voice@Work adds the new node to the call flow and changes the display in the Node Inspector window (Figure 225).

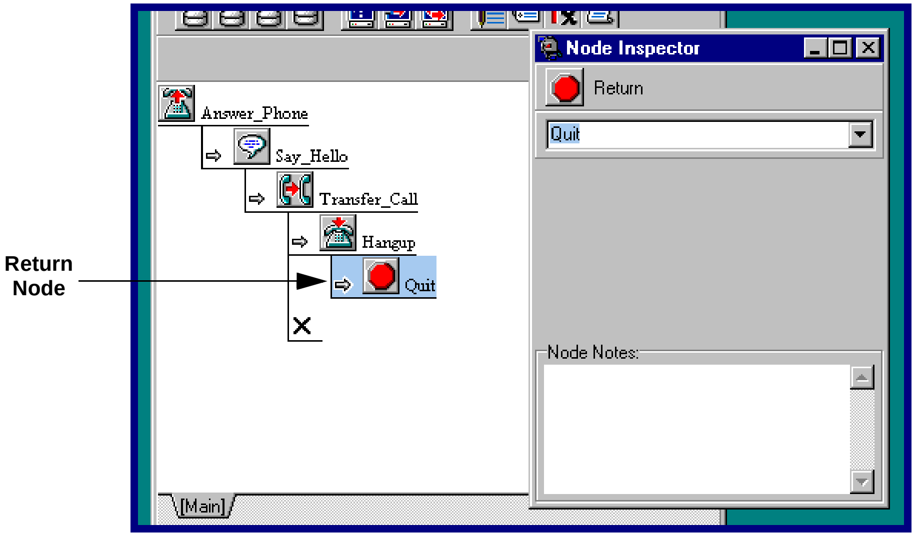


Figure 225. Call Flow with Return Node Added

Notice that the Return node has no branches underneath it.

6. Add another Announcement node by doing the following:

NOTE:
This node is executed only if the call cannot be transferred. The purpose of the announcement is to let the caller know before hanging up that the call could not be transferred.

a. Locate the Announcement node icon  on the Node Palette and drag it onto the design pad.

b. Place your cursor directly over the empty Fail branch of the Transfer Call node (Figure 225), designated by the large black X, and release the mouse button.

Voice@Work displays the node naming dialog box.

c. Type **Cannot_Transfer_Announcement** in the name field.

d. Click **OK**.

Voice@Work adds the new node to the call flow and changes the display in the Node Inspector window (Figure 226).

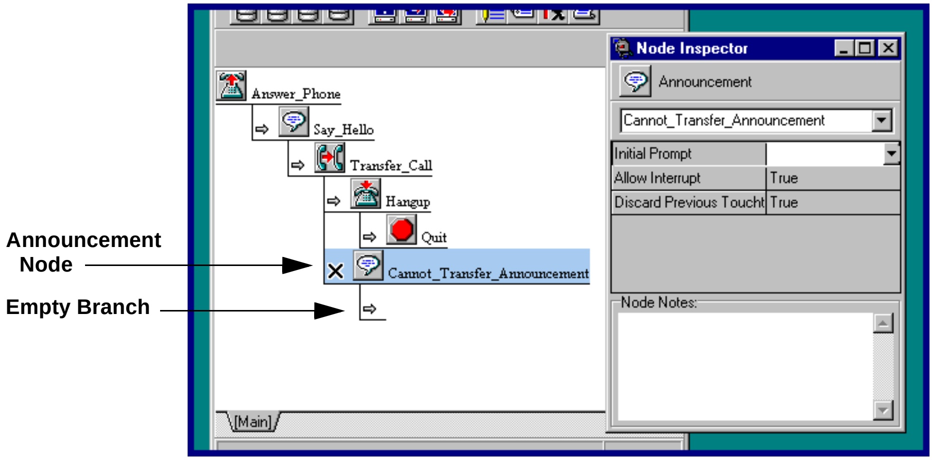


Figure 226. Call Flow with Final Announcement Node Added

Editing the Nodes

Many nodes require further definition before they are fully functional in a call flow. This editing can take the form of defining prompts, selecting variables, or defining argument values. These edits are accomplished using the Node Inspector. For more details about using the Node Inspector or editing individual nodes, see Chapter 30, “Working with Nodes” or Chapter 31, “Node Descriptions”.

In this sample application, the following node edits are required:

- Define the initial announcement prompt to be played to the caller when the call is answered.
- Set the telephone number transfer options for the Transfer Call node.
- Define the announcement to be played to the caller if the call cannot be transferred.
- Direct the empty branch on the **Cannot_Transfer_Announcement** Announcement node to jump to the Disconnect Call node.

Defining the First Announcement Node Prompt

As it stands now, the application knows to play an announcement to the caller after answering the phone. The application does not know, however, what to say in that announcement, because the content of that announcement has not been defined. You must define the Initial Prompt that will contain the announcement to play to the caller.

To define the Initial Prompt, do the following:

1. Click the Announcement node which you named **Say_Hello** (Figure 221).

   Voice@Work highlights the Announcement node. The Node Inspector moves to the background. Depending on how you have configured your workspace, it may now be either partially or totally hidden by the main window.

2. Click in the Node Inspector window to bring it to the front.

   NOTE:
   If the Node Inspector is totally hidden by the main window, you must bring it to the front by clicking its icon on the Node Palette, by selecting **Node Inspector** from the Tools menu, or by right-clicking the node you want to edit and selecting **Edit Node** from the popup menu.

   To avoid this, position the Node Inspector so that its window overlaps outside the main window's. This enables you simply to click in the window to bring it to the front.

3. Double-click in the **Initial Prompt** field.

   Voice@Work displays the Name Prompt dialog box (Figure 227).

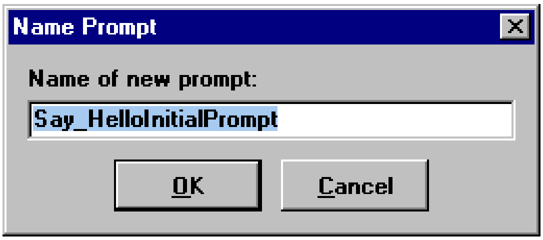


Figure 227. Name Prompt Dialog Box

4. Type **Greeting** to replace the suggested name in the prompt name field.
5. Click **OK**.

   Voice@Work displays the Prompt Editor window (Figure 228).

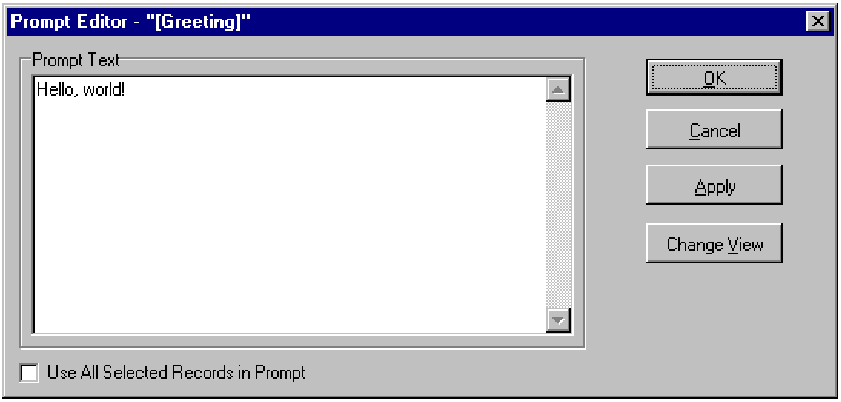


Figure 228. Prompt Editor Window

6. In the **Prompt Text** field, type the following:

   Hello, world!

   This will become a phrase and will be a part of the custom phrase table.

   NOTE:
   Unless you have Text-to-Speech installed on your target voice response system, the text you type here does not get played back or synthesized into audible speech in Voice@Work. This text is primarily for phrase identification purposes.

   To use recorded speech files to play back your phrase text during Simulation and on the target system, you must provide separate speech sound files. For information about recording or importing recorded speech files, see ''The Phrase Editor'' in Chapter 18, ''Working with Phrases''.

7. Click **OK**.

   Voice@Work displays the New Phrase Assignments window (Figure 229).

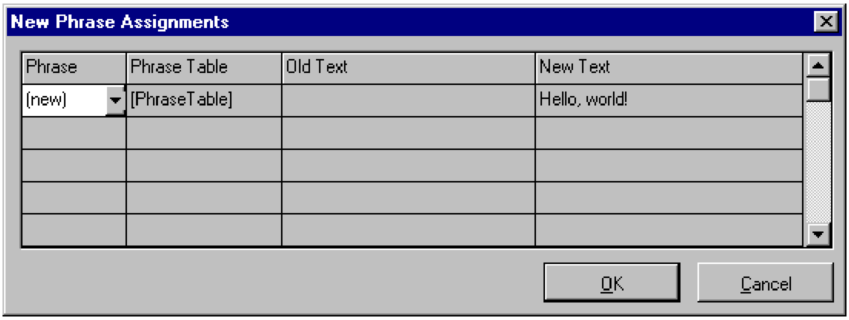


Figure 229. New Phrase Assignments Window

You can learn more about the New Phrase Assignments window in Chapter 18, "Working with Phrases". For now, accept the defaults.

8. Click **OK**.

   Voice@Work performs the following actions:

   - Creates a new prompt named [Greeting]
   - Places the name of the prompt [Greeting] in the Node Inspector **Initial Prompt** field
   - Creates a new phrase and automatically assigns it a phrase number
   - Opens the Globals Manager and displays the new prompt and new phrase in their respective global resource manager windows

You will see later how this prompt and phrase text are used in verifying and simulating your application.

Setting the Transfer Call Node Options

The application now knows that, after the announcement, it should transfer the call to another number, but it does not yet know how to handle the call transfer. You must tell it how to handle the transfer by setting the Transfer Call node options.

To set the Transfer Call node options, do the following:

1. Click the Transfer Call node, which you named **Transfer_Call**.

   Voice@Work highlights the Transfer Call node. The Node Inspector moves to the background. Depending on how you have configured your workspace, it may now be either partially or totally hidden by the main window.

2. Click in the Node Inspector window to bring it to the front.
3. Click in the **Phone Number** field of the Node Inspector (Figure 230) and enter the number **1234** or another, valid telephone number, if you plan to test this application on a working voice response system.

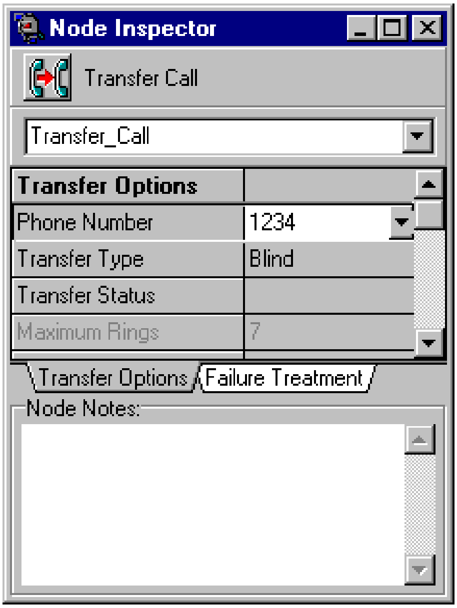


Figure 230. Node Inspector for Transfer Call Node

Notice that the **Transfer Type** field says **Blind**. This means that the call is transferred without the system checking for call completion.

Notice also that this node has two sets of options represented by two tabs near the bottom of the Node Inspector window: **Transfer Options** and **Failure Treatment**. For now, accept the defaults.

For more information about call transfer types and the other options on the Transfer Call node, see "The Transfer Call Node" in Chapter 31, "Node Descriptions".

Defining the Transfer Call Node Failure Announcement

The application now knows to answer the call, play a simple announcement, and transfer the call to another number before hanging up. But what if for some reason the call cannot be transferred, for example, due to an equipment failure?

In such a case, the application directs the call to the Fail branch of the Transfer Call node, where you placed an Announcement node called **Cannot_Transfer_Announcement**. You must tell the application what announcement to play when this Fail branch is taken. You want the system to play an announcement to the caller warning that the call transfer could not be completed before hanging up.

The procedure for defining this announcement are very similar to the announcement you created earlier. To define the failed transfer announcement, do the following:

1. Click the Announcement node which you named **Cannot_Transfer_Announcement** (Figure 226).

   Voice@Work highlights the **Cannot_Transfer_Announcement** Announcement node.

2. Click in the Node Inspector window to bring it to the front.
3. Double-click in the **Initial Prompt** field.

   Voice@Work displays the Name Prompt dialog box (Figure 231).

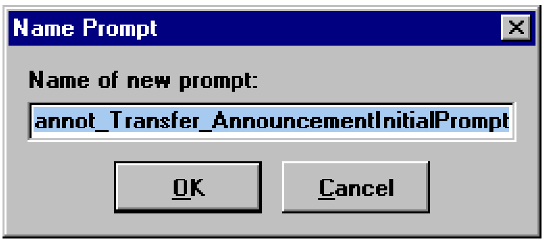


Figure 231. Name Prompt Dialog Box

4. Type **Cannot_Transfer** to replace the suggested name in the prompt name field.
5. Click **OK**.

   Voice@Work displays the Prompt Editor window (Figure 232).

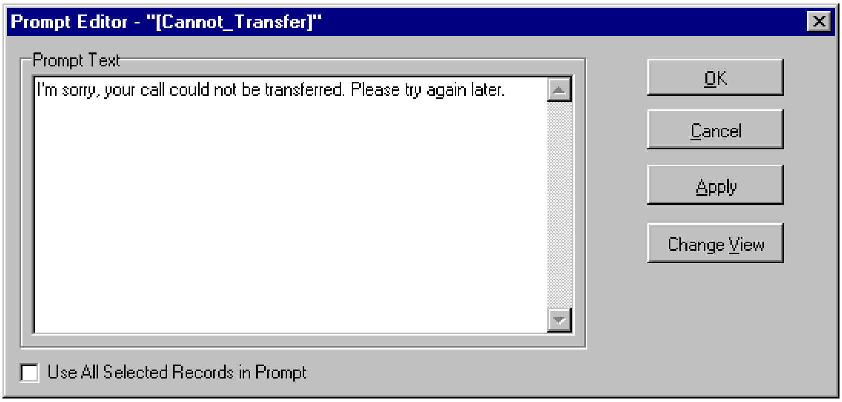


Figure 232. Prompt Editor Window

6. In the **Prompt Text** field, type the following:

   I'm sorry, your call could not be transferred. Please try again later.

   This will become a phrase and will be a part of the custom phrase table.

7. Click **OK**.

   Voice@Work displays the New Phrase Assignments window.

8. Click **OK** to accept the defaults, as you did before.

   Voice@Work performs the following actions:

   - Creates a new prompt named [Cannot_Transfer]
   - Places the name of the prompt [Cannot_Transfer] in the Node Inspector **Initial Prompt** field
   - Creates a new phrase and automatically assigns it a phrase number
   - Opens the Globals Manager and displays the new prompt and new phrase in their respective global resource manager windows

Directing the Call to Terminate after a Failed Call Transfer Attempt

The application now knows to play the announcement to play to the caller when a call cannot be transferred, but it still does not know how to finish the call in such a case. You must direct the application to disconnect the call and end the script, much as you did for the successful call transfer.

You could add another Disconnect Call node and another Return node to accomplish this, but you do not need to. Since you already have those nodes in place, you can simply instruct the application to jump to the Disconnect Call node and continue from there, whenever the Fail branch of the Transfer Call node is executed.

To direct the system to hang up the call if the call transfer cannot be completed, do the following:

1. Right-click the empty branch of the Announcement node which you named **Cannot_Transfer_Announcement** (Figure 226).

   Voice@Work displays a popup menu of options (Figure 233).

   NOTE:
   One of Voice@Work's powerful features is the extensive use of right-click "popup" menus. Most windows, nodes, and tools make use of them.

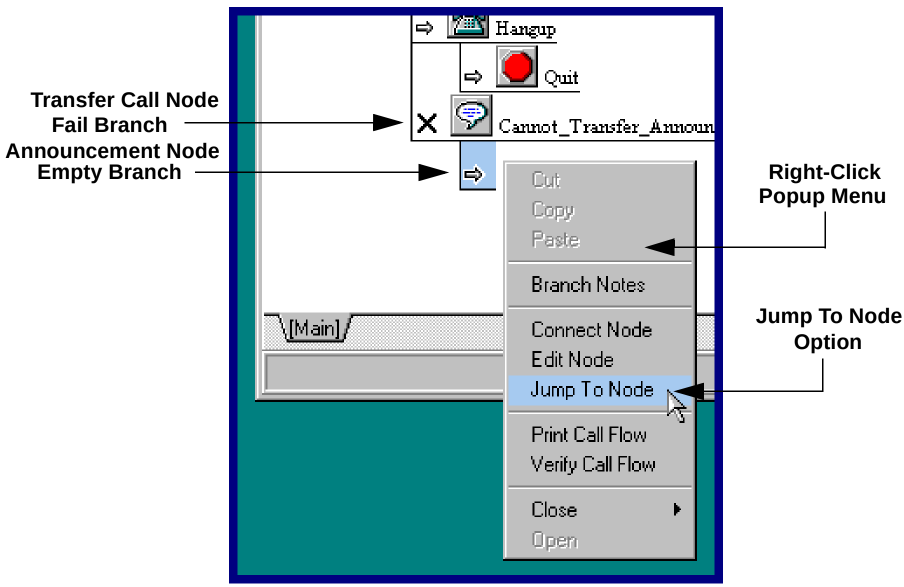


Figure 233. Right-Click Popup Menu for Transfer Call Node

2. Click the **Jump To Node** option.

   Voice@Work displays the Nodes dialog box (Figure 234). This dialog box displays all the nodes used in the current call flow.

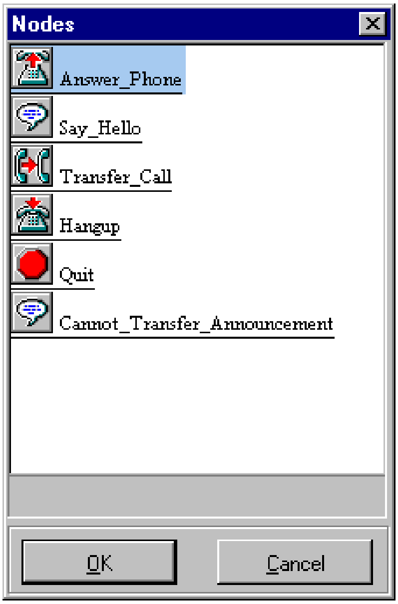


Figure 234. Nodes Dialog Box

3. Click the Disconnect Call node, which you named **Hangup**.
4. Click **OK**.

   Voice@Work changes the call flow's appearance to show that the empty branch now points to the Disconnect Call node (Figure 235). The symbol indicates that the node branch jumps to another node.

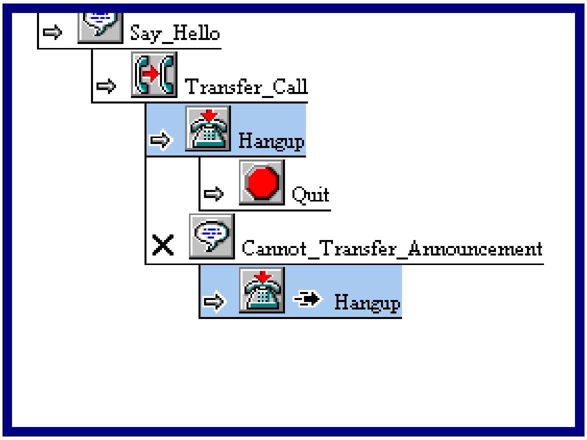


Figure 235. Call Flow with Jump to Disconnect Call Node

You have just completed building your first application call flow. Through building this simple call flow, you should now have an idea of how to build your own application call flows.

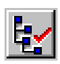
Once your application has been built, you must test and debug your application and generate the Transaction Assembler Script (TAS) code. The TAS code allows your application to be installed on the target voice response system. You must also transfer the generated TAS code to the target voice response system and install it.

Debugging the Application

Before generating the source code and transferring the application to the target system, you should always test and debug the application using the Verify Design and Simulation tools included with Voice@Work.

Verifying the Design

The Verify Design tool checks the application for possible errors or omissions in the application's design.

To verify the design using the Verify Design tool, click its icon  on the Toolbar.

The Verify Design tool automatically checks the application call flow and displays the results in the Verification Results window (Figure 236). The red check marks indicate the possible errors or omissions the Verify Design tool found.

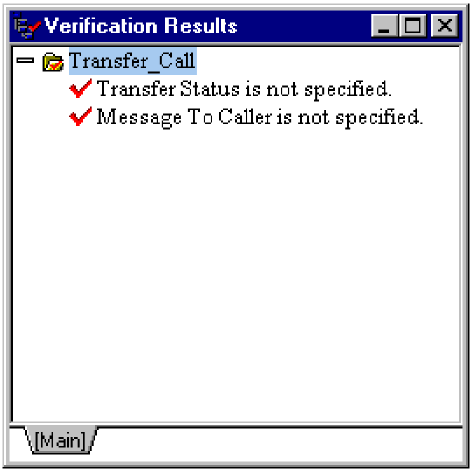


Figure 236. Verification Results Window

Since the Verify Design tool identifies only *possible* errors or omissions, you must decide whether the flagged items really are errors or omissions. In this case, the Transfer Status was deliberately left unspecified, so you need not be concerned about it.

You do, however, want to specify a message to notify that the call is to be transferred to another number. Otherwise, the caller might be confused upon hearing clicks and silence and might think there was something wrong with the system.

To correct this omission and define a message to play before transferring the call, do the following:

1. Double-click the second red check mark, designated **Message to Caller is not specified.**

   Voice@Work locates the omission in the Transfer Call node and highlights the node. At the same time, it locates in the Node Inspector the field you are to fill in.

2. To close the Verification Results window, click the close window  button.

3. Double-click the **Message to Caller** field in the Node Inspector window (Figure 237).

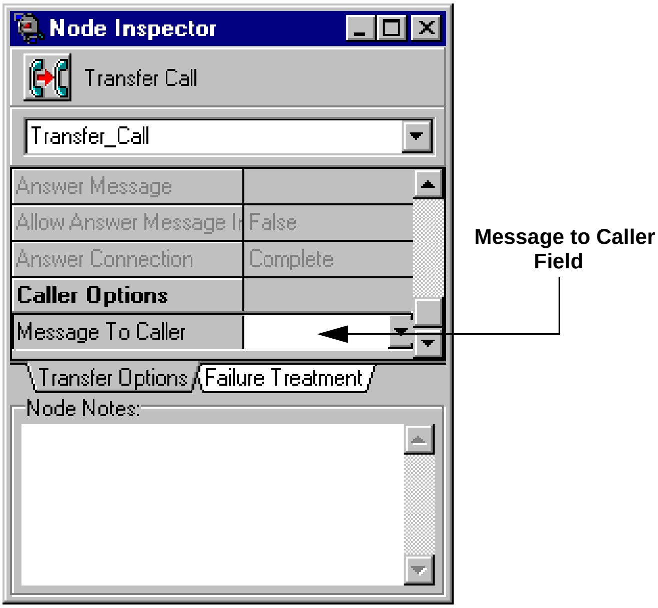


Figure 237. Node Inspector Window for Transfer Call Node

Voice@Work displays the Name Prompt dialog box.

4. Type **Transfer_Message** in the prompt name field.

   Voice@Work displays the Prompt Editor.

5. Type the following text in the **Prompt Text** field:

   Please hold while your call is transferred.

6. Click **OK**.

   Voice@Work displays the New Phrase Assignments window.

7. Click **OK**.

   Voice@Work performs the following actions:

   - Creates a new prompt named [Transfer_Message]
   - Places the name of the prompt [Transfer_Message] in the Node Inspector **Message to Caller** field
   - Creates a new phrase and automatically assigns it a phrase number
   - Opens the Globals Manager and displays the new prompt and new phrase in their respective global resource manager windows

For more information about the Verify Design tool, see Chapter 29, "The Verify Design Tool".

Simulating the Application's Performance

The Simulation tool allows you to simulate how the application will behave after you generate the code and transfer and install it on the target system. This tool can be very useful in finding problems with the application's design.

To simulate the application using the Simulation tool, do the following:

1. Click the Answer Call node, which you named **Answer_Phone**, to highlight it.

   NOTE:
   A simulation always starts running from whatever node is highlighted at the time the Simulation tool is selected. Highlighting this node allows the simulation to start from the beginning of the node.

2. Click the Simulation tool icon  on the Toolbar.

   The Simulation tool starts running automatically. You can monitor the progress of the simulation in the Application Simulation window (Figure 238).

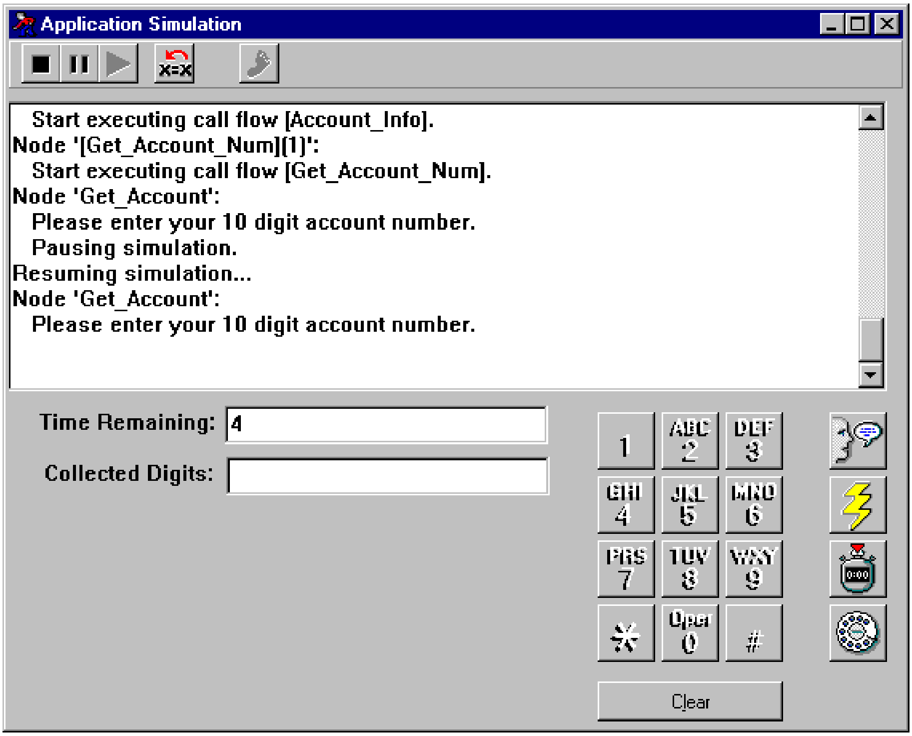


Figure 238. Application Simulation Window

When the simulation reaches the Transfer Call node, the Simulation tool asks how to simulate the result of the attempted call transfer. If you select the **Answer** option, the Simulation tool simulates a successful transfer. If you select the **Error...** option, the Simulation tool simulates an unsuccessful transfer and takes the Fail branch of the Transfer Call node.

3. Click **Answer**.

   The Simulation tool completes the simulation of a successful call transfer.

4. To thoroughly test the application, you should always simulate as many possibilities as you can. In this case, there is only one other possibility: that the call transfer attempt was not successful. To test this, run the simulation again, this time selecting the **Error...** option when the simulation reaches the Transfer Call node.

5. To close the Application Simulation window, click the close window button.

For more information about using the Simulation tool, see Chapter 28, "The Simulation Tool".

Generating the Code

Now that you have designed and tested the application, you can generate the TAS code to be transferred to the target system.

To generate the TAS code, do the following:

1. Click the **Code Generation/Application Transfer** icon.

   If you have not saved the application, Voice@Work displays a dialog box asking whether you want to save it. Otherwise, go to Step 3.

2. Click **Yes**.

   Voice@Work displays the **Code Generation / Application Transfer** window (Figure 239). This window displays the kinds of files you can generate from the current application. Notice that, because you have no database tables, host definitions, or phrase sound files defined for this application, these options are grayed out and inactive.

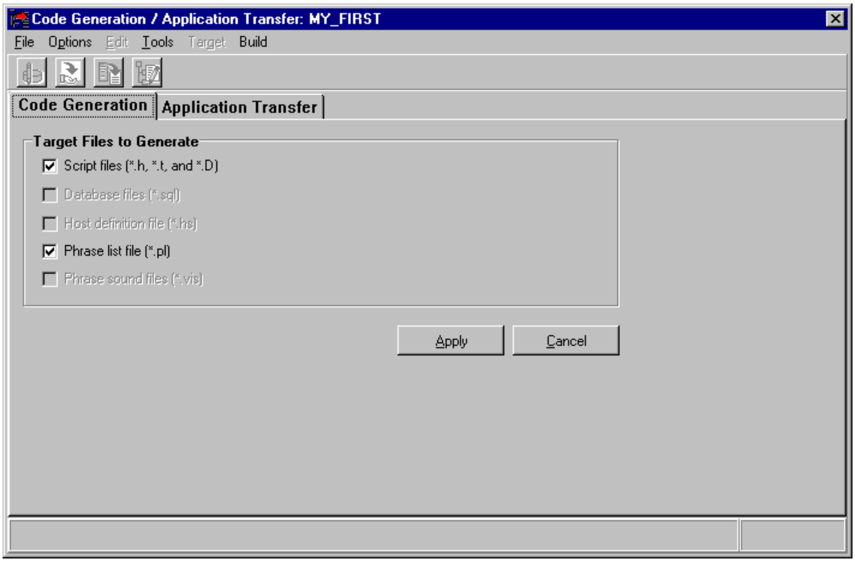


Figure 239. Code Generation / Application Transfer Window

3. Click **Apply**.

   Voice@Work automatically generates the TAS code needed to make this application run on the target system. When it is finished generating the code, Voice@Work notifies you of any errors encountered during the code generation process (Figure 240).

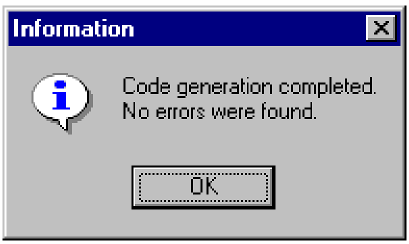


Figure 240. Code Generation Complete Information Box

4. Click **OK**.

For more information about the Code Generation tool, see Chapter 23, ‘‘The Code Generation Tool’’.

Transferring the Application

Once the TAS code is generated, you must transfer the application to the target system, which is where the application will actually run. Voice@Work provides two options for transferring your applications to the target system:

- On floppy disks
- Over a network

Procedures for transferring application files using both of these methods are in Chapter 24, "The Application Transfer Tool".

To see an animated demonstration of this process, see the **Demonstrations** on your Voice@Work CD-ROM disk.

Installing the Application on the Target System

Once the TAS code has been transferred to the target system, it must still be installed and compiled there before you can run the application. This must be done from the target system using the SCI tools. For more information about using the SCI tools to install applications, see Appendix A, "Using the SCI Tools".

For information about testing and running your application on the target system, see the following Lucent Technologies publications for your system:

- *INTUITY™ CONVERSANT® System Administration*, Voice System Administration chapter, Voice Equipment section, Voice Services subsection
- *INTUITY™ CONVERSANT® System Application Design Guidelines*, Testing and Using a Voice Response Application Design chapter

Transferring Files Using a Modem

H

Overview

There may be times when you need to transfer application files to a target system, but you do not have a LAN connection and it is impractical to accomplish the transfer using floppy disks. In such cases, you can establish a network connection using a modem and then transfer the files using the same application transfer procedures you would use for a LAN transfer.

Purpose

This appendix provides information about configuring PPP on PCs using Windows NT 4.0, 95, or 98 operating systems, and INTUITY CONVERSANT Version 6 or 7 systems. It also provides information on troubleshooting and tips for configuring PPP.

Topics include the following:

- Configuring Your System for PPP Connections
- Windows NT 4.0 PPP Configuration
- Windows 95/98 PPP Configuration
- UnixWare 1.1 PPP Configuration
- UnixWare 2.1 PPP Configuration
- Troubleshooting PPP Connections
- Additional Tips for Configuring PPP Connections

Configuring Your System for PPP Connections

To configure your system for PPP connections, you must configure both the PC and the target voice response system. Then you can establish network connections between machines and transfer application files using **Network** transfer options (see “Setting Application Transfer Options” in Chapter 24, “The Application Transfer Tool”).

To configure the PPP connection, select one option from each column of Table 156 below.

Table 156. Configuration Options
Select one from each of the following columns.

To configure a PC running:	To configure a target system running:
Windows NT 4.0, see “Windows NT 4.0 PPP Configuration”	INTUITY CONVERSANT Version 6.0 Update 1, see “UnixWare 1.1 PPP Configuration”
Windows 95 or 98, see “Windows 95/98 PPP Configuration”	INTUITY CONVERSANT Version 7.0, see “UnixWare 2.1 PPP Configuration”

NOTE:
The modem network connections described here may be used to conduct any type of TCP/IP activities you want; you are not limited just to transferring Voice@Work application files. You can, for example, use telnet or FTP between the two computers using this connection.

Sample Configuration Scenario

The procedures in the following sections are based on the following example scenario:

NOTE:
This scenario is an example only. When establishing this connection on your own system, change the information to match your configuration needs. Before beginning the PPP configuration procedures, you need to know the IP addresses for both the PC and the target system.

- Windows-based PC being configured:

Name:	WinPC
IP address:	192.0.0.1

- INTUITY CONVERSANT machine to be connected with:

Name:	ICex
IP address:	192.0.0.2
Phone number:	555-1212
PPP login/password:	mylogin / mypass
Serial port used:	tty01 (COM2)

For more information about configuring PPP connections on PCs and INTUITY CONVERSANT systems, see the appropriate documentation for the system in question.

Windows NT 4.0 PPP Configuration

To configure a PC running Windows NT 4.0 for PPP connections, do the following:

Configure the TCP/IP Settings

To configure the TCP/IP settings, do the following:

1. Open the Control Panel.
2. Double-click the **Network** icon.
3. Select the **Protocols** tab.
4. Verify that the TCP/IP protocol is installed. If it is not, click the **Add...** button and then select **TCP/IP Protocol**.

   NOTE:
   You must have administrative privileges on your Windows NT machine before you can install this protocol.

5. Click **OK**.

Create the Dial-Up Networking Connection

To create the Dial-Up Networking connection, do the following:

1. Double-click the **My Computer** icon on the desktop.

   The system displays the My Computer window.

2. Double-click the **Dial-Up Networking** icon.

   The system displays the **Dial-Up Networking** dialog box.

3. Click the **New...** button.

   The system displays the **New Phonebook Entry Wizard** dialog box.

4. Enter the name of the INTUITY CONVERSANT machine with which you want to connect.

   In the case of the example, the name is **ICex**.

5. Click **Next >**

   The system displays the **Server** dialog box.

6. Check the option that says: "The non-Windows NT server I am calling expects me to type login information after connecting, or to know TCP/IP addresses before dialing."

7. Click **Next >**

   The system displays the **Phone Number** dialog box.

8. Enter the phone number of the INTUITY CONVERSANT machine with which you want to connect.

   In the case of the example, the phone number is **555-1212**.

9. Click **Next >**

   The system displays the **Serial Line Protocol** dialog box.

10. Select the **Point-to-Point Protocol** (**PPP**) option.

11. Click **Next >**

    The system displays the **Login Script** dialog box.

12. Select the **Use a terminal window** option.

13. Click **Next >**

    The system displays the **IP Address** dialog box.

14. Enter the IP address for your PC.

    In the case of the example, the IP address is **192.0.0.1**.

15. Click **Next >**

    The system displays the **Name Server Addresses** dialog box.

16. Click **Next >**

    The system displays the **New Phonebook Entry Wizard** dialog box.

17. Click **Finish**.

Modify the Dial-Up Networking Connection

Before using the Dial-Up Networking connection, you must modify its settings.

To modify the Dial-Up Networking connection, do the following:

1. Double-click the **My Computer** icon on the desktop.

   The system displays the **My Computer** window.

2. Double-click the **Dial-Up Networking** icon.

   The system displays the **Dial-Up Networking** dialog box.

3. From the **Phonebook entry to dial** pulldown menu, select the connection you just created in "Create the Dial-Up Networking Connection".

   In the case of the example, select **ICex.**

4. Click the **More** button.

5. From the popup menu, select **Edit entry and modem properties...**

   The system displays the **Edit Phonebook Entry** dialog box.

6. Select the **Server** tab.

7. Verify that the **Dial-up server type:** field is set to **PPP: Windows NT, Windows 95Plus, Internet**

8. If the **IPX/SPX compatible** check box is selected, clear it.

9. If the **NetBEUI** check box is selected, clear it.

10. Click the **TCP/IP Settings...** button.

    The system displays the **PPP TCP/IP Settings** dialog box.

11. Verify that the **Specify an IP address** option is selected and the correct IP address is entered.

    In the case of the example, the IP address displayed should be **192.0.0.1**.

12. Verify that the **Server assigned name server addresses** option is selected.

    NOTE:
    In some cases, you need to assign the **Primary DNS** entry. For details, see "Troubleshooting PPP Connections".

13. Verify that the **Use IP header compression** check box is selected.

14. If your PC has an ethernet connection with a local network interface card (NIC) card, and you want to access systems on that network while you are connected using this PPP connection, make sure the **Use default gateway on remote network** check box is cleared.

15. Click **OK** (in the **PPP TCP/IP Settings** dialog box).

16. Click **OK** (in the **Edit Phonebook Entry** dialog box).

Open the PPP Connection

To open and use the PPP connection, do the following:

1. Double-click the **My Computer** icon on the desktop.

   The system displays the **My Computer** window.

2. Double-click the **Dial-Up Networking** icon.

   The system displays the **Dial-Up Networking** dialog box.

3. From the **Phonebook entry to dial** pulldown menu, select the connection you just created.

   In the case of the example, select **ICex**.

4. Click the **Dial** button.

   The system dials the target system. When the target system responds, the system displays the **Post_Dial Terminal Screen** and prompts you to log in.

5. Enter the login for the target system.

   In the case of the example, the login is **mylogin**.

   The system prompts you for the login password.

6. Enter the login password for the target system.

   In the case of the example, the login password is **mypass**.

   The system displays a series of unreadable characters in the **Post-Dial Terminal Screen** window.

7. Press the F7 key, or click **Continue**.

   The system finishes establishing the network connection and closes the **Post-Dial Terminal Screen** window.

Windows 95/98 PPP Configuration

To configure a PC running Windows 95 or Windows 98 for PPP connections, do the following:

Configure the TCP/IP Settings

To configure the TCP/IP settings, do the following:

1. Open the Control Panel.
2. Double-click the **Network** icon.
3. Verify that the TCP/IP protocol in installed. If it is not, click the **Add...** button and then select **TCP/IP Protocol**.
4. Click **OK**.

Create the Dial-Up Networking Connection

To create the Dial-Up Networking connection, do the following:

1. Double-click the **My Computer** icon on the desktop.

   The system displays the **My Computer** window.

2. Double-click the **Dial-Up Networking** icon.

   The system displays the **Dial-Up Networking** dialog box.

3. Click the **Make New Connection** icon.

   The system displays the **Make New Connection** dialog box.

4. Enter the name of the INTUITY CONVERSANT machine with which you want to connect.

   In the case of the example, the name is **ICex**.

5. Click **Next >**

6. Enter the telephone number of the INTUITY CONVERSANT machine with which you want to connect.

   In the case of the example, the telephone number would be **555-1212**.

7. Verify that the area code and country code are correct for the INTUITY CONVERSANT machine with which you want to connect.

8. Click **Next >**

9. Click **Finish**

Modify the Dial-Up Networking Connection

Before using the Dial-Up Networking connection, you must modify its settings.

To modify the Dial-Up Networking connection, do the following:

1. Double-click the **My Computer** icon on the desktop.

   The system displays the **My Computer** window.

2. Double-click the **Dial-Up Networking** icon.

   The system displays the **Dial-Up Networking** dialog box.

3. Right-click the icon for the dial-up networking connection you just created.

   In the case of the example, right-click the **ICex** icon.

4. From the popup menu, select **Properties**.

   The system displays the properties dialog box for the selected connection.

5. Select the **General** tab.

6. In the **Connect using:** area, click the **Configure...** button.

   The system displays the **Modem Properties** dialog box for the selected modem.

7. Select the **Options** tab.
8. In the **Connection control** area, select the **Bring up terminal window after dialing** option.
9. Select the **Connection** tab.
10. Verify that the connection preferences are **8** data bits, parity of **none**, and **1** stop bit.
11. Click **OK**.

    The system returns to the properties dialog box for the selected connection.

12. Do one of the following:
    - On Windows 95 machines, select the **Server Types** button.
    - On Windows 98 machines, select the **Server Types** tab.
13. Verify that the **Type of Dial-Up Server:** field is set to **PPP: Internet, Windows NT Server, Windows 98**.
14. If the **NetBEUI** check box is selected, clear it.
15. If the **IPX/SPX compatible** check box is selected, clear it.
16. Verify that the **TCP/IP** check box is selected.
17. Click the **TCP/IP Settings...** button.

    The system displays the **TCP/IP Settings** dialog box.

18. Verify that the **Specify an IP address** option is selected and the correct IP address is entered.

    In the case of the example, the IP address displayed should be **192.0.0.1**.

19. Verify that the **Server assigned name server addresses** option is selected.

    NOTE:
    In some cases, you will need to assign the **Primary DNS** entry. For details, see "Troubleshooting PPP Connections".

20. Verify that the **Use IP header compression** check box is selected.
21. If your PC has an ethernet connection with a local network interface card (NIC), and you want to access systems on that network while you are connected using this PPP connection, make sure the **Use default gateway on remote network** check box is cleared.
22. Click **OK** (in the **TCP/IP Settings** dialog box).
23. Click **OK** (in the properties dialog box).

Open the PPP Connection

To open and use the PPP connection, do the following:

1. Double-click the **My Computer** icon on the desktop.

   The system displays the **My Computer** window.

2. Double-click the **Dial-Up Networking** icon.

   The system displays the **Dial-Up Networking** dialog box.

3. Double-click the icon for the dial-up networking connection you just created.

   In the case of the example, right-click the **ICex** icon.

   The system displays the **Connect To** dialog box.

4. Click the **Connect** button.

   The system displays a **Connecting to...** message box and dials to call the target system. When the target system has been contacted, the system displays the **Post-Dial Terminal Screen** and prompts you to log in.

5. Enter the login for the target system.

   In the case of the example, the login is **mylogin**.

   The system prompts you for the login password.

6. Enter the login password for the target system.

   In the case of the example, the login password is **mypass**.

   The system displays a series of unreadable characters in the **Post-Dial Terminal Screen** window.

7. Press the F7 key, or click **Continue**.

   The system finishes establishing the network connection and closes the **Post-Dial Terminal Screen** window.

UnixWare 1.1 PPP Configuration

To configure the UnixWare 1.1 INTUITY CONVERSANT Version 6.0 Update 1 system for PPP connections, do the following:

1. Set Up the Serial Device

To set up the serial device, do the following:

1. Log in as root on the INTUITY CONVERSANT system.
2. At the system prompt, enter **vi /etc/uucp/Devices**
3. Use the vi editor to add the following line to the end of this file:
   Direct tty01h,M - 9600NP direct_modem

   NOTE:
   If your modem is connected to the COM1 port, use **tty00h** instead of **tty01h**.

4. Make sure there are no other entries in this file relating to tty01. If there are, delete them.
5. Save the file and close the **vi** editor.

2. Change the tty Settings

You must change the tty settings in your terminal window so that when you send the **cu** command directly to the port later, **cu** uses the current tty settings to make the connection. The **cu** command can change your modem settings automatically, based on these settings.

NOTE:
You might want to add this entry to your **.profile** file.

To change the tty settings, at the system prompt enter **stty -parenb cs8 9600**

3. Change the Settings on the Modem

To change the settings on the modem, do the following:

1. At the system prompt, enter **cu -l tty01h**

   The system displays a "Connected" message.

2. Enter the commands from Table 157 in the order listed:

   NOTE:
   You may notice that after entering these commands, the system may or may not respond with "OK". This behavior is normal, and the settings should take effect anyway. Note also that these settings may vary, depending on the modem you are using. Consult your modem documentation for details.

Table 157. Modem Setting Commands

Command	Description
ATSO=1	Turns the "auto answer" feature on
ATS15=130	Sets the parity properly
ATS20=7	Sets the line speed to 9600
ATQ1	Sets the modem to not send Result Codes. If you need to send Result Codes, enter the command ATQ0.
AT&W	Writes the changes to the modem
~.	Returns you to the shell

4. Create a ttymon Process

The following procedure creates a ttymon process to send a login message to incoming callers.

1. At the system prompt, enter **/usr/X/adm/addttymon**
2. After this utility is done, enter **vi /etc/saf/ttymon1/_pmtab**
3. Find the line that defines **/dev/tty01** and replace the word "auto" with "9600NP".

   The new line should look similar to the following:

   ```
   00h1:u::reserved:reserved:login:/dev/tty01h:bhr:0:/usr/
   bin/shserv:60:9600NP:ldte rm:SVR4.2 login\: :::::#
   ```
4. Make sure there is only one line defining **/dev/tty01** in this file. If there are others, remove them.
5. Save the file and exit the **vi** editor.
6. At the next two system prompts, enter
   sacadm -k -p ttymon1
   sacadm -s -p ttymon1

5. Configure the PPP Port

To configure the PPP port, do the following:

1. At the system prompt, enter **/usr/sbin/pppconf**

   The system displays the following menu:

   ```
   PPP Configuration Menu

   1) Configure PPP hosts
   2) Configure incoming PPP parameters
   3) Configure PPP authentication parameters
   q) Quit

   Enter choice:
   ```

2. Enter **1**

   The system displays the following menu:

   ```
   Configure PPP Hosts

   1) Add/modify PPP hosts
   2) Remove PPP hosts
   q) Quit

   Enter choice:
   ```

3. Enter **1**

   The system displays the following prompt:

   ```
   Enter IP address for local PPP end:
   ```

4. Enter the IP address for the INTUITY CONVERSANT machine.

   In the case of the example, it is the IP address of ICex, which is **192.0.0.2**.

   The system displays the following prompt:

   ```
   Enter IP address for remote PPP end:
   ```

5. Enter the IP address for the Windows-based PC with which you want to establish the connection.

   In the case of the example, it is the IP address of WinPC, which is **192.0.0.1**.

   NOTE:
   For either machine, you can use the IP address that is currently being used by the normal TCP/IP settings on that machine. A unique IP address is not necessary for PPP connections.

   The system displays the following prompt:

   ```
   Enter netmask ("-" for default) [255.255.255.0]:
   ```

6. Enter the netmask for your network.

   In the case of the example, it is **255.255.255.0**.

   The system displays the following prompt:

   ```
   Enter UUCP name ("-" for null):
   ```

7. Enter **-** (a hyphen)

   NOTE:
   If you enter anything other than a hyphen here, the system displays additional prompts not discussed here. For more information, see the *TPC/IP Administration* guide for your system.

   The system displays the following menu:

   ```
   Configure PPP Hosts

   1) Add/modify PPP hosts
   2) Remove PPP hosts
   q) Quit

   Enter choice:
   ```

8. Enter **q**

   The system displays the following menu:

   ```
   PPP Configuration Menu

   1) Configure PPP hosts
   2) Configure incoming PPP parameters
   3) Configure PPP authentication parameters
   q) Quit

   Enter choice:
   ```

9. Enter **2**

   The system displays the following menu:

   ```
   Configure Incoming PPP Parameters

   1) Add/modify incoming PPP setup
   2) Remove incoming PPP setup
   q) Quit

   Enter choice:
   ```

10. Enter **1**

    The system displays the following prompt:

    ```
    Enter PPP login name:
    ```

11. Enter the login name that the PC will use to establish the PPP connection.

    In the case of the example, it is **mylogin**.

    NOTE:
    When selecting a login ID, you must use all lower case letters and numbers and follow the login naming conventions for your system.

    The system displays the following prompt:

    ```
    Do you want to create PPP login account mylogin? (y/n)
    ```

12. Enter **y**

    The system displays the following prompt:

    ```
    Enter a password for login "mylogin"
    New password:
    ```

13. Enter the password that the PC will use to establish the PPP connection.

    In the case of the example, it is **mypass**.

    The system prompts you to re-enter the password, and then displays the following prompt:

    ```
    Do you want to specify PPP negotiation parameters?
    (y/n)
    ```

14. Enter **n**

    The system displays the **Configure Incoming PPP Parameters** menu.

15. Enter **q**

    The system displays the **PPP Configuration Menu**.

16. Enter **q**

6. Modify the ppphosts File

Before going on to the last step, there are a couple of items that you must check, and if necessary modify, in the **ppphosts** file.

To modify the **ppphosts** file, do the following:

1. At the system prompt, enter **vi /etc/inet/ppphosts**

   The **ppphosts** file opens in the **vi** editor. At the bottom of the file is your configuration information.

2. Verify that the **paptmout=n** option does not exist. If option does appear, remove it from the file. Make sure only one space exists between all fields in this line after removing the option.

3. Find the machine login line.

   This line takes the form:
   `*`*`loginID`*` - -`
   where *loginID* represents the login to used by the calling machine.

   In the case of the example, it is **mylogin**.

4. Add the following to the login line:
   VJ mru=1500

   The final appearance of this line should be similar to the following example:
   `*`*`mylogin`*` - - VJ mru=1500`

   The letters **VJ** turns on the Van Jacobson TCP header compression needed to make the PPP connection work properly. The **mru=1500** entry resolves problems that can arise when using FTP from a Windows 95/98 machine to a UnixWare 1.1 machine.

5. Save the file and close the **vi** editor.

7. Restart TCP/IP

Before your PPP settings can take effect, you must restart the TCP/IP.

To restart the TCP/IP, do the following:

1. At the system prompt, enter **sh /etc/rc2.d/S69inet stop**
2. Wait at least 5 seconds, and then enter **sh /etc/rc2.d/S69inet start**
3. If you receive any socket errors, repeat steps 1 and 2.

NOTE:
As an alternative to restarting the TCP/IP, you can also reboot the entire system by entering the following command at the system prompt:
shutdown -y -g0 -i6

UnixWare 2.1 PPP Configuration

As part of its standard configuration, the INTUITY CONVERSANT Version 7.0 system uses the remote maintenance circuit card (RMCC) for modem communications. Thus, in most cases, you do not need a separate modem for PPP connections.

To configure the UnixWare 2.1 INTUITY CONVERSANT Version 7.0 system for PPP connections, do the following:

1. Log in as root on the INTUITY CONVERSANT system.
2. At the system prompt, enter **pppconf**

   The system displays the following menu:

   ```
   PPP Configuration Menu

   1) Configure/Modify Dedicated PPP Link
   2) Configure/Modify Dynamic Outgoing PPP Link
   3) Configure/Modify Manual Bringup PPP Link
   4) Configure/Modify Dynamic Incoming PPP Link
   5) Delete an entry
   6) Undo Last Session (Selections 1-5)
   7) List Configures Links
   8) Configure PPP Authentication Parameters
   q) Quit

   Enter choice:
   ```

3. Enter **4**

   The system displays the following prompt:

   ```
   Enter PPP login name (’q’ to quit):
   ```

4. Enter the login name that the PC will use to establish the PPP connection.

   In the case of the example, it is **mylogin**.

   NOTE:
   When selecting a login ID, you must use all lower case letters and numbers and follow the login naming conventions for your system.

   The system displays the following prompt:

   ```
   Are you going to use address pooling for the local
   address? (y/n)
   ```

5. Enter **n**

   The system displays the following prompt:

   ```
   Enter IP address or node name for local side of PPP link
   (- for none):
   ```

6. Enter the IP address for the INTUITY CONVERSANT machine.

   For the example, it is the IP address of ICex, which is **192.0.0.2**.

   The system displays the following prompt:

   ```
   Are you going to use address pooling for the remote
   address? (y/n)
   ```

7. Enter **n**

   The system displays the following prompt:

   ```
   Enter IP address or node name for remote side of PPP
   link (- for none):
   ```

8. Enter the IP address for the Windows-based PC with which you want to establish the connection.

   For the example, it is the IP address of WinPC, which is **192.0.0.1**.

   NOTE:
   For either machine, you can use the IP address that is currently being used by the normal TCP/IP settings on that machine. A unique IP address is not necessary for PPP connections.

   The system displays the following prompt:

   ```
   Do you want to specify PPP advanced options? (y/n)
   ```

   NOTE:
   If you are interested in setting advanced options, please see the *UnixWare Network Administration* guide for your system. This example does not deal with these advanced options.

9. Enter **n**

   The system displays the following prompt:

   ```
   Do you wish to enable packet filtering on this link?
   (y/n)
   ```

10. Enter **n**

    The system displays the following prompt:

    ```
    Do you want to create PPP login account? (y/n)
    ```

11. Enter **y**

    The system displays the following prompt:

    ```
    Enter a password for login "mylogin"
    New password:
    ```

12. Enter the password that the PC will use to establish the PPP connection.

    In the case of the example, it is **mypass**.

    The system prompts you to re-enter the password, and then displays the following prompt:

    ```
    Do you want to specify PPP negotiation parameters?
    (y/n)
    ```

13. Enter **n**

    The system displays the **Configure Incoming PPP Parameters** menu.

14. Enter **q**

    The system displays the **PPP Configuration Menu**.

15. Enter **q**

Troubleshooting PPP Connections

This section contains information that may be helpful when troubleshooting PPP connections between a PC and an INTUITY CONVERSANT system.

Use Table 158 to find the problem you are experiencing.

Table 158. Troubleshooting Problems and Solutions

Problem	Recommended Solution
PPP connection apparently established, but it does not work properly	Close the PPP connection on the target INTUITY CONVERSANT system and then reopen it. Then re-establish the PPP connection between the two systems. For the procedure, see "Manually Configuring the PPP Connection".
PPP connection established, but TCP/IP commands do not work	Try assigning the Primary DNS entry in the PPP TCP/IP Settings dialog box.

Continued on next page

Table 158. Troubleshooting Problems and Solutions — *Continued*

Problem	Recommended Solution
Cannot log in as root to the INTUITY CONVERSANT system from a PC (remote login)	By default, the root login is denied on the INTUITY CONVERSANT when using remote access, such as from a PC. It is possible, however, to configure your INTUITY CONVERSANT system to allow remote access. For the procedure, see “Remote Login Access Procedures”.
Cannot do remote FTP as root	By default, INTUITY CONVERSANT systems do not allow you to FTP as root. It is possible, however, to configure your INTUITY CONVERSANT system to allow you to FTP as root. For the procedure, see “Enable Remote FTP as Root”.
cu -l command gives DEVICE FAILED or DEVICE NOT AVAILABLE message	This problem has been observed when running a port monitor on a port, as INTUITY CONVERSANT systems do to issue login prompts. To correct this problem, you must disable the **ttymon** service. After you have disabled the **ttymon** service, the **cu -l** command will work. When you are finished with that command, you must enable the **ttymon** service again before login prompts are issued again to incoming callers. For more information about disabling or enabling the **ttymon** service, see “Disabling or Enabling ttymon Service”.
No login prompt given for **cu** command	This problem is caused by the target INTUITY CONVERSANT system still having the PPP link semi-active on the port. To correct it, you must kill the **-ppp** process for the port you are attempting to dial into. For the procedure, see “Killing the -ppp Process”.

Continued on next page

Table 158. Troubleshooting Problems and Solutions — *Continued*

Problem	Recommended Solution
PC dials the target system and then hangs up	This problem occurs when the target INTUITY CONVERSANT system has not been set up for Peer Authentication Protocol (PAP). To correct it, you must make sure the target system is set up for PAP. For the procedure, see "Setting Up PAP".
Telnet works but FTP hangs	This problem seems most prevalent when trying to FTP from a Windows 95 or Windows 98 machine to a UnixWare 1.1 (INTUITY CONVERSANT) system using PPP. To correct the problem, try adding the **mru=1500** option to the **ppphosts** file. For the procedure, see "Adding the mru=1500 Option to ppphosts".

Additional Tips for Configuring PPP Connections

This section contains tips to help you when configuring PPP connections between a PC and an INTUITY CONVERSANT system.

Below is a list of the topics in this section:

- Manually Configuring the PPP Connection
- Remote Login Access Procedures
- Enable Remote FTP as Root
- Disabling or Enabling ttymon Service
- Killing the -ppp Process
- Setting Up PAP
- Adding the mru=1500 Option to ppphosts
- Restarting TCP/IP
- Killing or Restarting the Port Monitor
- Understanding Configuration Files
- Debugging PPP Problems in UnixWare 2.0+

Manually Configuring the PPP Connection

Once the PPP connection has been established on the INTUITY CONVERSANT system, the default is for the connection to stay up forever, regardless of the idle time. This can occasionally cause problems when trying to establish a new PPP connection or re-establish a PPP connection from a PC.

One solution for such problems is to bring down the PPP connection if it is still up and then bring it back up.

Checking the Status of the PPP Connection from the PC

To find out whether the PPP connection is up, do the following:

1. Log in as root on the target INTUITY CONVERSANT system.
2. At the system prompt, enter **ifconfig -a**

   The system displays information similar to the following:

```
lo0: flags=49<UP,LOOPBACK,RUNNING>
    inet 127.0.0.1 netmask ff000000
sme0: flags=23<UP,BROADCAST,NOTRAILERS>
    inet 135.XXX.XXX.XXX netmask ffffff00 broadcast
    135.XXX.XXX.255
ppp0: flags=11<UP,POINTOPOINT>
    inet 192.0.0.2 --> 192.0.0.1 netmask ffffff00
```

   The **bold** text in the sample above is the information you are looking for and indicates in the example that a PPP connection (**ppp0**) is up.

   NOTE:
   The PPP connection above (**ppp0**) is an example only. The number might be any number, depending on which how your system is set up to number PPP connections.

Manually Closing the PPP Connection

To close the PPP connection manually, do the following:

1. Log in as root on the target INTUITY CONVERSANT system, if you are not already logged in.
2. At the system prompt, enter **ifconfig ppp# down**

   Replace the # symbol with the number displayed by your system using the **ifconfig -a** command.

   Running the **ifconfig -a** command now would display the following change to the **ppp0** line in the example above:

```
ppp0: flags=10<POINTOPOINT>
```

Reopening the PPP Connection Manually

Once you have closed the PPP connection, you must reopen it before you can use it again.

To reopen the PPP connection manually, do the following:

1. Log in as root on the target INTUITY CONVERSANT system, if you are not already logged in.
2. At the system prompt, enter **ifconfig ppp# up**

   Replace the # symbol with the number displayed by your system using the **ifconfig -a** command.

Remote Login Access Procedures

There are times when you need to log in to an INTUITY CONVERSANT system as the root user from a remote location, such as from a PC. By default, however, the root login is denied when using remote access.

To reconfigure your system to allow remote access, do the following from the INTUITY CONVERSANT system:

1. Log in as root.
2. Modify the **/etc/default/login** file by commenting out the following line:
   console=/dev/console
3. Modify the **/etc/shells** file by removing comment tags from the line containing the following:
   #/sbin/sh

Enable Remote FTP as Root

There are times when you need to FTP files to an INTUITY CONVERSANT system as root from a remote location, such as from a PC. By default, however, the root FTP is denied when using remote access.

To reconfigure your system to allow remote FTP as root, do the following from the INTUITY CONVERSANT system:

1. Log in as root.
2. At the system prompt, enter **vi /etc/shells**
3. Find the line containing the following string:
   #/sbin/sh
4. If the line contains a comment tag, remove the tag.
5. Save the file and close the **vi** editor.
6. At the system prompt, enter **vi /etc/ftpusers**

7. This file contains a list of all the users that are *not* allowed to transfer files using FTP.
8. Find the line that contains the word **root** and remove it.
9. Save the file and close the **vi** editor.

Disabling or Enabling ttymon Service

To disable ttymon service, do the following:

1. Log in as root on the target INTUITY CONVERSANT system.
2. At the system prompt, enter **pmadm -d -p ttymon1 -s 01h**

   NOTE:
   The service **01h** may differ, depending on the port you are using. To see the service tag for the particular port on your system, enter the **pmadm -l** command.

To enable ttymon service, do the following:

1. Log in as “root” on the target INTUITY CONVERSANT system.
2. At the system prompt, enter **pmadm -e -p ttymon1 -s 01h**

Killing the -ppp Process

The **-ppp** process can cause the system to not issue a login prompt when the **cu** command is given. This is caused when the target system still has the PPP link for the port in a semi-active state. To correct this situation, you must kill the **-ppp** process.

To kill the **-ppp** process, do the following:

1. Log in as root on the target INTUITY CONVERSANT system.
2. At the system prompt, enter **ps -ef | grep ppp**

   NOTE:
   In this command, **|** represents the pipe symbol, not the lowercase letter L.

3. Look for the PPP "shell", shown as **-ppp**, running on the port you are attempting to dial in to. If the shell is still active on the port, kill the process.

   This frees the port so the tty monitor can once again issue a login prompt.

Setting Up PAP

When the PAP has not been set up on the target Intuity CONVERSANT system, the PC will dial the target system and then hang up before a PPP connection can be established. To correct this problem, you must ensure that the PAP is set up correctly on the target system.

To set up the PAP on the target system, do the following:

1. Log in as root on the target INTUITY CONVERSANT system.
2. At the system prompt, enter **vi /etc/inet/ppphosts**

   This command opens the **ppphosts** file, so you can check to see whether the PAP is properly set up.
3. Find the configuration line which contains the string:
   <ppplogin> - -

   NOTE:
   The ***<ppplogin>*** in the example above is the name of the PPP login you are checking.

4. Verify that the option **paptmout=n** does not appear in this line. If it does, remove it.

   Make sure you leave only one space between all fields in this line.
5. Save the file and close the **vi** editor.

Adding the mru=1500 Option to ppphosts

If telnet works but FTP hangs, try adding the **mru=1500** option to the **ppphosts** file.

To add the **mru=1500** option, do the following:

1. Log in as root on the target INTUITY CONVERSANT system.
2. At the system prompt, enter **vi /etc/inet/ppphosts**

   The ppphosts file opens in the **vi** editor. At the bottom of the file is your configuration information.
3. Find the machine login line.

   This line takes the form:
   <loginID> - -
   where ***<loginID>*** represents the login to used by the calling machine.

4. Add the following to the login line:
   mru=1500

   The final appearance of this line should be similar to the following example:
   `*<loginID> - - mru=1500`

5. Save the file and close the **vi** editor.

Restarting TCP/IP

To restart the TCP/IP, do the following:

1. Log in as root on the target INTUITY CONVERSANT system.
2. At the system prompt, enter **sh /etc/rc2.d/S69inet stop**
3. Wait at least 5 seconds, and then enter **sh /etc/rc2.d/S69inet start**
4. If you receive any socket errors, repeat steps 2 and 3.

As an alternative to restarting the TCP/IP, you can also reboot the entire system by entering the following command at the system prompt:
shutdown -y -g0 -i6

Killing or Restarting the Port Monitor

To kill the port monitor, at the system prompt enter **sacadm -k -p ttymon#** where # represents the number of the port monitor being killed.

To restart the port monitor, at the system prompt enter **sacadm -s -p ttymon#** where # represents the number of the port monitor being restarted.

Understanding Configuration Files

The INTUITY CONVERSANT system uses a number of configuration files for the PPP process. Some of these are described in the following subtopics.

/etc/passwd

This file contains PPP login user name and password information. It is created by the **pppconf** command while setting up the "incoming" parameters.

CAUTION:
*You should not modify this file manually with an editor such as the **vi** editor. Instead, you should use the procedure given here.*

To create a login manually for sample user mylogin, do the following:

1. Log in as root on the target INTUITY CONVERSANT system.
2. At the system prompt, enter
   useradd -d /usr/lib/ppp -s /usr.lib/ppp/ppp mylogin

   This command adds a new user to the file.
3. At the system prompt, enter **passwd mylogin**
4. Enter the desired password at the system prompts.

/etc/saf/ttymon1/_pmtab

This file contains service information that the target voice response system uses to issue a login prompt to callers dialing in.

This entry is created by the utility **/usr/X/adm/addttymon**. This utility uses the **/etc/uucp/Devices** file to determine which ports it needs to set up a login for.

You may modify this entry, but after doing so, you must kill and restart the port monitor.

The following is an example of a line in this file:

```
00h1:u::reserved:reserved:login:/dev/tty01h:bhr:0:/usr/bin/s
hserv:60:9600NP:ldte rm:SVR4.2 login\: :::::#
```

Normally, the only thing you would want to change is the baud rate, which is highlighted in **bold** in the example above. By default, the **addttymon** command creates this entry with "any" as the baud rate. To assure that the speed is always constant, change this rate to "9600NP", as in the example above.

NOTE:
The **NP** in the example above stands for "No Parity", which means the login is sent at **9600**, **8**, **1**, **None**. This baud rate field can be any key field in the **/etc/ttydefs** file.

/etc/confnet.d/inet/interface

This file is used by TCP/IP to know what interfaces, such as network cards and PPP ports, are available. The appropriate lines in this file are built by the **pppconf** command. You can manually edit this file, but you must restart TCP/IP after doing so (see "Restarting TCP/IP" above).

Example:

```
ppp:0:199.60.6.6:/dev/ppp:199.70.7.7 netmask
255.255.255.0:add_ppp:
```

The syntax is straightforward:

NOTE:
In this example, columns are separated by colons (:).

- Column 2 indicates which PPP interface is being defined, starting with 0.
- Column 3 is always the local IP address for the interface.
- Column 5 begins with the IP address of the remote system with which you want to connect.

To display what interfaces are available to TCP/IP, at the system prompt enter **ifconfig -a**

When TCP/IP is started, it reads the interface file to set up its interfaces. This file is what the ifconfig -a command reports. The system does not look at this file again after it is started, so any modifications you make to the file require you to restart the TCP/IP.

/etc/uucp/Devices

This file contains device information needed for the connection server and PPP to communicate with the serial device (modem).

Example:

```
Direct tty01h,M - 9600NP direct_modem
```

/etc/inet/ppphosts

The Intuity CONVERSANT system uses this file to configure PPP connections and negotiations.

Example:

`*<mylogin> - - VJ`
where `<mylogin>` represents the name of the login

The file itself contains comments that help build these lines with any options, such as, for example, Van Jacobson compression (**VJ** in the example above) or connection timeout options.

The **paptmout** option should not be in this file unless PAP is enabled on the target system.

/etc/hosts

This file contains the required entry for each other computer with which you want to establish PPP connections.

Example:

`192.0.0.1 PC_alias`

Debugging PPP Problems in UnixWare 2.0+

The following information is taken from Novell's technical information document, TID021586.

UnixWare 2.0+ PPP has the ability to output debugging information and packet traces to help you debug PPP problems. The following information describes how to use this feature:

1. Set up the **syslogd** configuration file by doing the following:
    a. Log in as “root” on the target INTUITY CONVERSANT system.
    b. At the system prompt, enter **vi /etc/syslog.conf**

       Depending on your current configuration, this may be a new file.
    c. Add the following line to the file:
       ***.info /var/adm/log/info**

       NOTE:
       The white space between the two fields must be a tab. Regular spaces will not work.
2. Save the file and close the **vi** editor.

3. At the system prompt, enter **syslogd**
4. In the **ppphosts** file, set the debug flag by doing the following:
   a. At the system prompt, enter **vi /etc/inet/ppphosts**
   b. In the line for the system you want to dial, or the "incoming line", add the following option:
      debug=1

      The following is an example of what this line might look like:

      ```
      137.37.37.37 - testsystem debug=1
      ```

      NOTE:
      The debug levels go from 1 to 15. 1 is the lowest level you can set, but still very useful. 15 is a full packet dump, which can be useful at times but which is very verbose and consumes a great deal of disk space.

      The level 1 debug information is displayed only until the negotiation phase is completed. This is where most problems occur, so level 1 is best for most situations. Level 9 provides a full packet dump even after the negotiation phase has been completed. The packet traces are in hex, with no further information provided, so it is very difficult to interpret.

   c. Save the file and close the **vi** editor.
5. In a separate window, do a **tail** on the debug file to view the log in real time.

   To do a **tail** on the debug file, at the system prompt enter
   tail -f /var/adm/log/info

   NOTE:
   If you receive an error indicating that the "info" file does not exist, create it before the tail command is issued, by entering the following command at the system prompt:
   touch /var/adm/log/info

6. Establish a PPP connection.

   The system displays the debugging information in the "tail" window.

Custom WholeWord Grammars

I

Overview

Standard WholeWord grammars are best suited for applications that require single-digit number or yes/no responses from the caller. For other applications requiring other types of responses, you may want to consider obtaining and adding your own custom WholeWord grammars. Examples of application responses that might require the use of a custom grammar include the following:

- Days of the week
- Telephone area codes
- Postal zip codes
- Times of the day

Voice@Work supports the use of custom WholeWord grammars. The Custom WholeWord Grammars tab of the Preference Editor provides options for selecting, adding, and removing custom WholeWord grammars in Voice@Work. Once they are available in Voice@Work, they can be imported into applications and shared with other applications. In general, they are treated like any other (standard) WholeWord grammar.

Purpose

This appendix provides information about the use of custom WholeWord grammars in Voice@Work. Topics include the following:

Obtaining Custom WholeWord Grammars

To obtain custom WholeWord grammars, you can either:

- Purchase them "ready-made" from Lucent Technologies or an ISV
- Create your own using the Advanced WholeWord Grammar Tool (AWGT) available from Lucent Technologies

For more information about these options, contact your Lucent Technologies account representative.

Either way, your custom WholeWord grammar package should include a Voice@Work grammar (.grm) file to go with the custom grammar that is installed on the Intuity CONVERSANT system. It is your responsibility to get the custom grammar installed on the Intuity CONVERSANT system and to make sure the .grm file can be accessed from your Voice@Work PC.

Adding the Custom WholeWord Grammar to Voice@Work

Once the custom WholeWord grammar has been obtained and installed on the Intuity CONVERSANT and is accessible to the Voice@Work PC, you must still add it to Voice@Work before it is available in your applications.

To add the custom WholeWord grammar to Voice@Work, do the following:

1. Open the Preference Editor (see Chapter 27, “The Preference Editor”, Opening and Using the Preference Editor).
2. Click the **Custom WholeWord Grammars** tab.

3. Select the custom WholeWord grammar file to add by doing one of the following:

   - Click the **Browse** button and use the **Specify Custom WholeWord Grammar File** dialog box to locate the desired *.grm file. Then click Open.
   - Type the full path- and filename of the desired *.grm file in the **WholeWord Grammar File:** field.

   The **WholeWord Grammar File:** field displays the path- and filename of the selected *.grm file.

   If an application is open, the **WholeWord Grammar Language:** field displays the name of the language associated with that application. If no application is open, the **WholeWord Grammar Language:** field displays the name of the default language for Voice@Work.

4. (Optional) If you want the custom grammar associated with a language different from that displayed in the **WholeWord Grammar Language:** field, select the desired language from the pulldown menu.

5. Click **Add Custom WholeWord**.

   Voice@Work adds the custom grammar, placing a copy of it in the appropriate language grammar directory, and displays a message similar to the following (Figure 241):

Figure 241. Custom WholeWord Grammar Successfully Installed Message

6. Click **OK** twice.

   This makes the custom grammar available to any application created using the selected language.

Importing the Custom WholeWord Grammar into an Application

Once a custom WholeWord grammar has been made available within Voice@Work (see Adding the Custom WholeWord Grammar to Voice@Work above), you must import it into your application before you can use it.

To import the custom grammar, do the following:

1. Open the Application Resources tool (see Chapter 21, "The Application Resources Tool", Importing the Custom WholeWord Grammar into an Application).
2. Click the **Import** tab.
3. From the pulldown menu, select **Custom WholeWord Grammar**.

   Voice@Work displays a list of the custom WholeWord grammars available for the application language.
4. Click the grammar you want to import into the application.
5. Verify that the **Create Private Copy** radio button is selected.
6. Click **Import**.

   Voice@Work makes a local copy of the custom WholeWord grammar and places it in the application directory. The custom grammar is now available for use within the application. It subsequently is treated by Voice@Work like any other WholeWord grammar and appears in any list of WholeWord grammars for that application, for example, in the Response and Confirmation tabs of IVR nodes or in the Application Resources tool.

   NOTE:
   Even if you later remove the custom WholeWord grammar from Voice@Work (see Removing a Custom WholeWord Grammar from Voice@Work below), this local copy ensures that the custom grammar will still be available to the application.

Replacing a Custom WholeWord Grammar

If for some reason you find it necessary to change or replace a custom WholeWord grammar file, it must be re-added to Voice@Work before the change will be effective.

To change or replace a custom WholeWord grammar in Voice@Work, follow the procedure for Adding the Custom WholeWord Grammar to Voice@Work. When you click the **Add WholeWord Grammar** button, the system displays a message similar to the following (Figure 242):

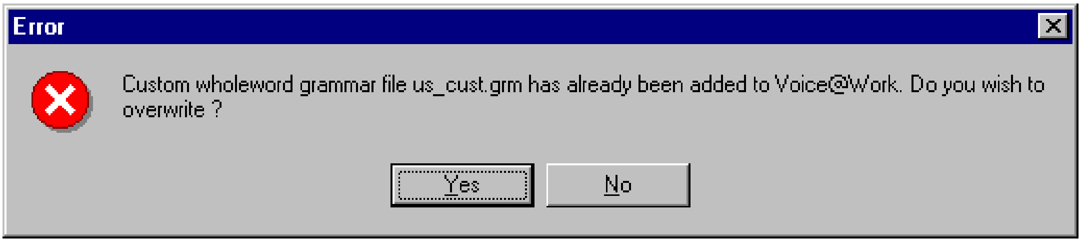


Figure 242. Custom WholeWord Grammar Replacement Error Message

To update Voice@Work with the new version of the grammar file, click Yes.

NOTE:
If you replace a custom WholeWord grammar in Voice@Work, you must also re-import it into any application that uses it before the changes become effective in that application (see Re-importing a Custom WholeWord Grammar below).

Re-importing a Custom WholeWord Grammar

When you replace a custom WholeWord grammar in Voice@Work, the changes in the custom grammar are not automatically recognized by the applications using that custom grammar, because of the fact that each application uses a local copy of the custom grammar. Thus, before the changes can become effective in a given application using that custom grammar, you must re-import it into the application.

To re-import a custom WholeWord grammar, follow the procedure for Importing the Custom WholeWord Grammar into an Application. When you click **Import**, the system displays a message similar to the following (Figure 243):

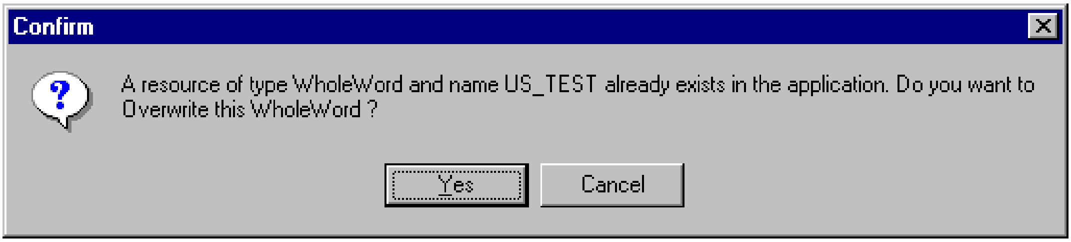


Figure 243. Custom WholeWord Grammar Re-import Confirmation Request

To update the application with the new version of the custom grammar, click **Yes**.

NOTE:
If you have used a custom grammar with an enumerated list and you have used a list item as an ASR branch name, and then you later re-import an updated version of the custom grammar that no longer supports the item used for the ASR branch, then you must modify the obsolete branch(es) accordingly.

Removing a Custom WholeWord Grammar from Voice@Work

There may be times when you no longer want a custom WholeWord grammar available and you want to remove it from Voice@Work.

To remove a custom WholeWord grammar from Voice@Work, do the following:

1. Open the Preference Editor (see Chapter 27, "The Preference Editor", Opening and Using the Preference Editor).
2. Click the **Custom WholeWord Grammars** tab.
3. Verify that the language associated with the grammar you want to remove is selected in the **WholeWord Grammar Language:** field.
4. Click **Remove Custom WholeWord...**

   Voice@Work displays the **Select Grammar File to Remove** dialog box (Figure 244).

Figure 244.

5. Click the name of the custom grammar you want to remove.
6. Click **OK**.

   Voice@Work displays a message similar to the following (Figure 245):

Figure 245. Custom WholeWord Grammar Successful Removal Message

7. Click **OK** twice.

NOTE:
Even after you remove a custom WholeWord grammar from Voice@Work, any application you created using it will still have it available, because a local copy of the grammar is made for each application using it. Therefore, if you want to also make the grammar unavailable for such applications, you must delete it from each application (see Deleting a Custom WholeWord Grammar from an Application below).

Deleting a Custom WholeWord Grammar from an Application

There may be times when you no longer want a custom WholeWord grammar available within a particular application. To make it unavailable, you must delete it from the application.

NOTE:
Even if you have already removed the custom grammar from Voice@Work, a local copy still exists for each application into which it was imported. Therefore, if you want to also make the grammar unavailable for such applications, you must use this procedure to delete it from each application.

To delete a custom WholeWord grammar from an application, do the following:

1. Open the application using the specified custom WholeWord grammar.
2. Open the Application Resources tool (see Chapter 21, “The Application Resources Tool”, Importing the Custom WholeWord Grammar into an Application).
3. Click the **Usage** tab.
4. Select (highlight) the desired custom WholeWord grammar on the WholeWord Grammar list.

   NOTE:
   You may need to expand the WholeWord Grammar list to find the desired grammar.

5. Right-click the name of the desired custom grammar.
6. From the popup menu, click **Delete**.

   Voice@Work displays a message similar to the following (Figure 246):

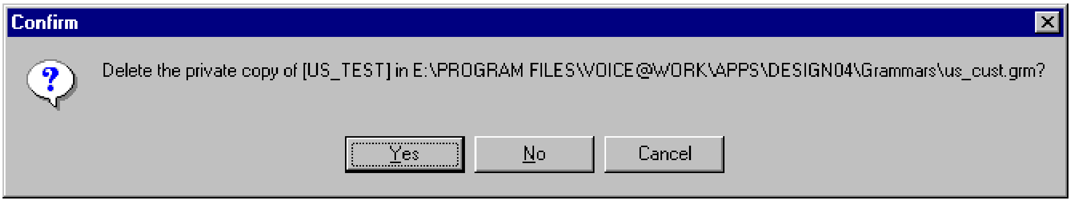


Figure 246. Custom WholeWord Grammar Delete Confirmation Message

7. Click **Yes**.

   Voice@Work deletes the custom grammar from the application and updates the Application Resources tool display.

Glossary

Numerics

3270 interface

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

A

active call flow

Call flow currently showing in the main window (design pad) and available for editing. See also "call flow" and "design pad."

adaptive differential pulse code modulation

(ADPCM) A means of encoding analog voice signals into digital signals by adaptively predicting future encoded voice signals. This adaptive modulation method reduces the number of bits required to encode voice recordings.

Adjunct/Switch Application Interface

(ASAI) An optional feature package that provides an Integrated Services Digital Network-based interface between Lucent Technologies PBXs and adjunct processors.

ADPCM

adaptive differential pulse code modulation

advanced speech recognition

A speech recognition ability that allows the system to understand spoken inputs from callers. See also "WholeWord speech recognition" and "FlexWord™ speech recognition."

American Standard Code for Information Interchange

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

ANI

automatic number identification

announcement

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

application

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

application resource

See "global resource."

application script

The computer program that controls the application (the transaction between the caller and the system). See also “application.”

application source files

The files that are used by Voice@Work to generate TAS code during the code generation process. While they are not used by the target system, these files can be transferred to the target voice response system to assist technical support personnel in troubleshooting applications.

application transfer

An option that allows you to transfer files interactively or directly to/from another computer using the file transfer system (FTS).

arguments

Variable values that are passed from one function, method, or application script to another.

ASAI

Adjunct/Switch Application Interface

ASCII

American Standard Code for Information Interchange

ASR

advanced speech recognition

asynchronous event

An event detected by the system that disrupts the normal flow of the application. At present, the only asynchronous event recognized by INTUITY CONVERSANT systems is “inbound call termination,” encountered when the caller hangs up.

attributes

Properties or values assigned to or characteristic of variables.

automatic number identification

(ANI) A method of identifying the calling party by automatically receiving a string of digits that identifies the calling station of a particular customer.

B

backup
The preservation of the information in a file in a different location, so that the data is not lost in the event of hardware or system failure.

backing up an application
Using a utility to make an archive copy of a completed application or an interim copy of an application in progress. The backup copy can be restored to the system if the online version is damaged, or if you make revisions and want to go back to the previous version.

barge-in
A capability provided by advanced speech recognition that allows callers to speak or enter their responses during the prompt and have those responses recognized. See also "advanced speech recognition."

blind transfer protocol
A protocol in which a call flow is terminated as soon as the phone number is dialed, without waiting to see if the telephone is busy or if the called party answers.

bps
Bits per second.

branches
Connections to a node, indicated in Voice@Work by lines leading down and to the right from a node.

breakpoints
Designated places within a call flow where the application temporarily stops running during a simulation. A breakpoint is designated in a call flow with the  icon.

buttons
See "icons."

C

call classification analysis
(CCA) A process that enables application designers to use information available within the system to classify the disposition of originated and transferred calls.

call data events
Variables in the application that you ask your system to store in a database for retrieval later on. Any data can be designated as call data events, though typically they are counters of some kind. When a call is made to the INTUITY CONVERSANT system, the application stores the values of call data events at the end of each call, for use in generating Call Data Reports and Call Data Summary Reports.

call progress tones
Standard telephony sounds that indicate the status of the call. These sounds include busy, fast busy, ringback, reorder, etc.

call flow

1. A series of nodes linked together. **2.** The “path” that a caller is directed through when interacting with a voice response system such as an INTUITY CONVERSANT system. See also “transaction.”

called party number

The number dialed by the person making a telephone call. Telephone switching equipment can use this number to selectively route an incoming call to a particular department or agent.

caller

The party who calls for a service, gets connected to the “target voice response system,” and interacts with it. As the target system can also make outbound calls for service, the caller can also be the person who responds to those outbound calls.

CCA

call classification analysis

code generation

The process in Voice@Work of generating the TAS code necessary to transfer and install an application to the target voice response system. See also “Transaction Assembler Script language (TAS).”

command

An instruction or request the user issues to the system software to make the system perform a particular function. An entire command in UNIX consists of a command name and options.

configuration

The arrangement of the software and hardware in a computer system or network.

contiguous nodes

Any selected series of nodes in a call flow, in which the selected nodes are all directly adjacent to one another without any unselected nodes in between. Any unselected node that breaks up the series makes the series discontiguous. See also “node” and “branches.”

control

A component of a prompt used to alter the way a prompt is played to the caller. Controls can be used to regulate the speed at which a phrase is played, for example, or to insert a silence or pause between phrases.

control bar

The area at the top of the Globals Manager where tabstops are created and placed.

custom speech

Unique words or phrases to be used in INTUITY CONVERSANT system voice prompts that Lucent Technologies custom records on a per-customer basis.

D

database

A structured set of files, records, or tables.

database field

A field used to extract values from a local database and form the structure upon which a database is built.

database record
The information in a database for a person, product, event, etc. The database record is made up of individual fields for each information item.

database table
A structure, made up of columns and rows, that holds information in a database. Database tables provide a means of storing information that changes too often to “hard-code,” or store permanently, in the transaction outline.

debug
The process of locating and correcting errors in computer programs; also referred to as “troubleshooting.”

default
The way a computer performs a task in the absence of other instructions.

design pad
The area in the main window that opens when you first open Voice@Work. It is where you do most of the work in creating call flow applications.

dial ahead
The ability to collect and process touch-tone inputs in sequence, even when they are received before the prompts.

dialed number identification service
(DNIS) A service that allows incoming calls to contain information about the telephone number for which it is destined.

dial pulse recognition
(DPR) A feature that allows the target voice response system to recognize digits dialed by callers with rotary dial telephones.

dial through
A capability provided by touchtone recognition that allows callers to enter responses during the prompt and have those responses recognized (similar to the Speak with Interrupt capability). See also “barge-in.”

dialog box
A secondary window in which you are asked to fill in information and/or make choices before closing it and returning to the main window.

directory
A type of file used to group and organize other files or directories. Also called “folder.”

discontiguous nodes
Two or more nodes selected and grouped together that have other unselected nodes in between. See also “contiguous nodes.”

DNIS
dialed number identification service

docking
The act of attaching a palette or window to another window so that it appears to become part of the second window. In Voice@Work, the term docking applies to both palettes and global resource managers.

DPR
dial pulse recognition

E

empty branch
A node branch that has no node connected to it. See also “branches.”

Enhanced Basic Speech
(EBS) Pre-recorded speech available from Lucent Technologies in several languages. Sometimes called “standard speech.”

error message
A message on the screen indicating that something is wrong, often with a suggestion for correcting it.

event
The notification given to an application when some condition occurs that is generally not encountered in normal operation.

external functions
Specific predefined (or customer-created) system tasks that Voice@Work can call or *invoke* to interact with other products or services. The function allows the application developer to enter arguments for the function to act on. Examples include concat, getarg, length, substring, etc.

F

field
See “database field.”

file
A collection of data treated as a basic unit for storage.

file transfer
An option that allows you to transfer files interactively or directly to/from another computer using the file transfer system (FTS).

File Transfer Protocol
(FTP) A standard protocol for transferring files using a modem and phone lines.

filename
Alphanumeric characters used to identify a particular file.

flash
See “switch hook flash.”

FlexWord™ speech recognition
A type of speech recognition based on subword technology that recognizes phonemes, or parts of words, in a specific language. Compare to “WholeWord speech recognition.” See also “phoneme.”

folder
A type of file used to group and organize other files or directories. Also called “directory.”

FTP
See “File Transfer Protocol.”

Full CCA
A feature package that augments the types of call dispositions that Intelligent CCA can provide. See also “call classification analysis” and “intelligent CCA.”

function arguments
See “arguments.”

G

global
See “global resource.”

global resource
Any of several types of components that can be used in building Voice@Work applications. Types include but are not limited to phrases, prompts, (database) tables, variables, host definitions, and external functions. See also “shared resource.”

global resource manager
Any of several windows that can be opened to display information about global resources in an application. For example, information about variables can be viewed in the Variables Manager. Right-clicking in a global resource manager brings up a popup menu which presents a different set of actions and options for each global resource manager. See also “global resource.”

grammar
The set of inputs that a speech recognizer can match (identify) from a caller.

graphical user interface
(GUI) A computer interface that allows a user to interact with the computer using a graphically oriented environment. This type of interface typically makes extensive use of graphic “objects” such as icons, buttons, and cursors.

group/tab
On the external functions palette, a set of external functions that are grouped together and whose icons appear grouped together.

GUI
graphical user interface

H

hard disk drive
A high-capacity data storage/retrieval device that is located inside a computer. A hard disk drive stores data on nonremovable high-density magnetic media based on a predetermined format for retrieval by the system at a later date.

hardware
The physical components of a computer system, such as keyboards, monitors, and media disk drives.

host
See “host computer.”

host computer

A computer linked to a network to provide a range of services, such as database access and computation. The host computer operates in a time-sharing manner with other computers linked to it via the network.

host flow

A specialized call flow, the primary purpose of which is to set up the interactions between the target voice response system and the host. There is one host flow for each host definition in the application.

I

icons

Graphical images, often designed to look like push buttons, used to activate commands or perform actions within a graphical user interface (GUI).

import

To incorporate a global resource or external file and make it part of or available to the current application.

independent software vendor

(ISV) A company that has an agreement with Lucent Technologies to develop software to work with the INTUITY CONVERSANT system to provide additional features required by customers.

Integrated Services Digital Network

(ISDN) A network that provides end-to-end digital connectivity to support a wide range of voice and data services.

intelligent CCA

A type of Call Classification Analysis (CCA) in which the line is monitored after dialing is complete, to determine whether a busy, reorder (fast busy), or other failure has been encountered. It also recognizes when the extension is answered or if the extension is not answered after a specified number of rings. The monitoring capabilities are dependent on the network interface circuit card and protocol used. See also “call classification analysis.”

interface

The access point of and means of interacting with a computer system.

interrupt

The termination of voice and/or telephony functions when a particular condition occurs. For example, if an interrupt option is turned on, and a caller speaks a response before a prompt is done playing, the prompt terminates immediately, and the caller is moved to the next step in the call flow. See also “barge-in.”

ISDN

Integrated Services Digital Network

ISV

independent software vendor

ITAC

International Technical Assistance Center

K

Kbyte
kilobyte

keyboard mapping
The method used to assign values to keys on a computer keyboard. Keyboards can be mapped for a particular type of computer interface emulation or for a particular language's alphabet.

L

LAN
local area network

local area network
(LAN) A data communications network in a limited geographical area. The LAN provides communications between computers and peripherals.

local database
A database residing on the "target voice response system."

logical unit
(LU) A type of Systems Network Architecture (SNA) Network Addressable Unit.

logging on/off
Entering or exiting the target voice response system software.

LU
logical unit

M

main window
The window that appears when you first start Voice@Work.

main topic window
The window that appears on the screen when the online help is first accessed. Most of the online help topics appear in this window.

Mbyte
megabyte

menu
Options presented to a user on a computer screen or with voice prompts.

menu bar
The area at the top of a window containing the various menus.

N

name bar

A field in a global resource manager which contains the name of the global resource for that resource manager and which is distinguishable from other fields by its magenta color.

EXAMPLE: 

is the Phrases Manager name bar.

node

A graphic object considered a basic “building block” for Voice@Work call flows. Each node represents a bit of code that is to be generated in the final application.

non-modal window

A window in the Windows operating system that allows other windows to simultaneously be open and active. This allows you to easily move back and forth between multiple windows in an application.

null value

An entry containing no value. A field containing a null value is normally displayed as blank and is different from a field containing a value of zero.

O

online help

Messages or information that appear on the user’s screen when the F1 “function key” is pressed or when the Help menu is accessed.

open branch

See “empty branch.”

option

An argument used in a command line to modify program output by modifying the execution of a command. When you do not specify any options, the command executes according to its default options.

ORACLE

A company that produces relational database management software. It is also used as a generic term that identifies a database residing on a local or remote system which is created and maintained using an ORACLE RDBMS product.

P

palette

An area, usually at the top of a window, in which icons that perform a variety of commands are displayed. In Voice@Work, there are three palettes, each with its own set of icons: the Toolbar, the Node Palette, and the External Functions Palette.

PAP

Peer Authentication Protocol

parent nodes
Nodes that have one or more subordinate nodes connected to them, which appear below them in the call flow. See also “node” and “subordinate nodes.”

PBX
private branch exchange

PC
Personal computer.

peripheral (device)
Equipment, such as a keyboard or monitor, that is added to the basic processing device in a computer system.

phoneme
A single basic sound of a particular spoken language. For example, the English language contains 40 phonemes that represent all basic sounds used with the language. The English word “one” can be represented with three phonemes, “w” - “uh” - “n.” Phonemes vary between languages because of guttural and nasal inflections and syllable constructs.

phrase
In Voice@Work, a bit of recorded speech used in constructing prompts. A phrase in Voice@Work may be a single word, a group of words, or a complete sentence or group of sentences. See also “prompt.”

phrase list
A list of phrases contained in a phrase table in Voice@Work. Each phrase list entry consists of two parts: the phrase number and the phrase tag. These phrase lists become the basis of talkfiles on the target voice response system. See also “talkfile.”

phrase numbers
Numbers used to identify individual phrases. Voice@Work assigns phrase numbers automatically and consecutively when you create new phrases.

phrase table
A set of phrases grouped together as a set.

phrase tag
A string of up to 50 characters that identifies the contents of a speech phrase used by an application script.

phrase text
The text used to designate a particular phrase. The first 50 characters of the phrase text become the phrase tag. See also “phrase tag.”

popup
A window or menu that appears on the screen when a button or popup topic link is clicked.

popup topic link
A hyperlink in an online help system that causes a popup window to appear. It is usually distinguished by a dashed underline under a term or phrase.

popup menu
A menu that appears on the screen when the secondary, usually the right, mouse button is clicked. See also “right-click menu.”

popup topic window
A window that appears on the screen when a popup topic link is clicked. Popup topics usually explain or define terms.

port
A connection to a device that allows data or information to travel into and out of the device.

PPP
Point-to-Point Protocol

PRI
Primary Rate Interface

Primary Rate Interface
(PRI) An ISDN term for connections over E1 or T1 facilities that are usually treated as trunks.

private branch exchange
(PBX) A private telephone switching system, usually located on a company's premises, which provides a central switching point for a number of individual telephones. Often found in hotels, businesses, government agencies, etc.

prompt
A message played to a caller that gives the caller a choice of selections in a menu and asks for a response. Compare to "announcement."

pulldown menu
A menu that can be accessed by clicking a menu options icon to reveal the menu items.

R

RDBMS
Relational Database Management System

recognizer
The part of the system that compares caller input to a grammar in order to correctly match (identify) the caller input. See also "WholeWord speech recognition" and "FlexWord™ speech recognition."

recovery
The process of trying to establish or re-establish a working connection between the target voice response system and a remote host.

Relational Database Management System
(RDBMS) ORACLE's relational database management system

remote database
Information stored on a system other than the target voice response system that can be accessed by the target system.

resource
See "global resource."

resource manager
See "global resource manager."

return values
Variable values that are passed back to a function, method, or application script from another function, method, or script that has been invoked.

right-click menu

In Voice@Work, the popup menu called by clicking on a window or element with the secondary, usually the right, mouse button.

NOTE: Although we recognize that some computer mouse devices may be configured so that the left mouse button is the secondary button, we have chosen in this documentation to refer to a secondary mouse click as a right-click, according to popular usage and convention.

S

SCI tools

Service Creation Installation tools

script

The set of instructions for the target voice response system to follow during a transaction.

Service Creation Installation tools

(SCI tools) A set of software tools designed to help you install your Voice@Work applications and related files on the target voice response system.

shared resource

A resource, such as a database table or speech pool, that is used by more than one application or developer. See also "global resource."

software

The set or sets of programs that instruct the computer hardware to perform a task or series of tasks.

speech file

A file containing a digitally recorded speech phrase.

speech phrase

A continuous speech segment encoded into a digital string. See also "phrase."

speech recognition

The ability of a system to understand spoken input from callers.

SQL

structured query language

standard external functions

A set of external functions provided by Lucent Technologies as part of the Voice@Work package. Sometimes called "system functions." See also "external functions."

standard speech

The speech package available in several languages containing simple words and phrases, produced by Lucent Technologies for use with INTUITY CONVERSANT systems. This package includes digits, numbers, days of the week, and months, each spoken with initial, medial, and falling inflection. The speech is in digitized files stored on the hard disk, to be used in voice prompts and messages to the caller. This feature is also called Enhanced Basic Speech (EBS).

standard vocabulary

A standard package of simple word speech models provided by Lucent Technologies and used for WholeWord speech recognition. These phrases include the digits "zero" through "nine," "yes," "no," and "oh," in US English, or the equivalent words in another language.

string
A contiguous sequence of characters treated as a unit. Strings are normally bounded by white spaces, tabs, or a character designated as a separator. A string value is a specified group of characters symbolized by a variable.

structured query language
(SQL) A standard data programming language used with data storage and data query applications.

status bar
The box at the bottom of the main window that displays help and status messages to aide in creating applications.

subordinate nodes
Nodes that are connected to one or more “parent” nodes, which appear above them in the call flow. See also “node” and “parent nodes.”

subword technology
A method of speech recognition, used in FlexWord™ speech recognition, that recognizes phonemes or parts of words. Compare to “WholeWord speech recognition.” See also “phoneme.”

switch hook flash
A signaling technique in which the signal is originated by momentarily depressing the “switch hook.”

system administrator
The person assigned the responsibility of monitoring all computer system software processing, performing daily system operations and preventive maintenance, and troubleshooting errors as required.

system functions
See “standard external functions.”

system variables
Variables that are predefined and included as part of the Voice@Work package.

T

table
See “database table.”

table schema
The structure or design of a database table, established by defining data record fields for the table.

talkfile
An ASCII file that contains the speech phrase tags and phrase tag numbers for all the phrases of a specific application. The speech phrases are organized and stored in groups. Each talkfile can contain up to 65,535 phrases, and the speech file system can contain multiple talkfiles. See also “phrase list.”

target voice response system
The system on which the Voice@Work application is to be installed (for this release, only INTUITY CONVERSANT V6 and V7 systems).

TAS
Transaction Assembler Script language

TCP/IP
Transfer Control Protocol/Internet Program

Text-to-Speech
(tts or TTS) An optional feature that allows an application to play US English speech directly from ASCII text by converting that text to synthesized speech. The text can be used for prompts or for text retrieved from a database or host, and can be spoken in an application with prerecorded speech.

trace
A command that can be used to monitor the execution of a script.

transaction
The interactions (exchanges) between the caller and the voice response system. A transaction can involve one or more telephone network connections and voice responses from the target voice response system. It can also involve one or more of the system optional features, such as speech recognition, 3270 host interface, FAX Actions, etc.

Transaction Assembler Script language
(TAS) A proprietary language developed by Lucent Technologies, Inc., designed especially for use on INTUITY CONVERSANT systems.

troubleshooting
The process of locating and correcting errors in computer programs. This process is also referred to as debugging.

TSO
Technical Services Organization

tts or TTS
Text-to-Speech

U

undocking
The act of removing a palette or window from another window so that it becomes a separate window. In Voice@Work, the term undocking applies to both palettes and global resource managers.

UK
United Kingdom

US
United States of America

UNIX Operating System
A multiuser, multitasking computer operating system.

V

variable

An application-specific information holder. Variables may be used to collect and hold data, act as counters, or make use of system performance information.

W

WholeWord speech recognition

An optional feature, available in several languages, based on whole-word technology that can recognize the numbers one through zero, “yes”, and “no” (the key words). This feature is reliable, regardless of the individual speaker. This feature can identify the key words when spoken in phrases with other words. A string of key words, called *connected digits*, can be recognized. During the prompt announcement, the caller can speak or use touch tones.

whole-word technology

The ability to recognize an entire word, rather than just a phoneme or a part of a word. Compare to “subword technology.”

window

A standard Windows® operating system graphical device used as part of the Voice@Work graphical user interface (GUI).

word list

A set of words available for FlexWord recognition by an application during a prompt and collect action step.

Index

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W