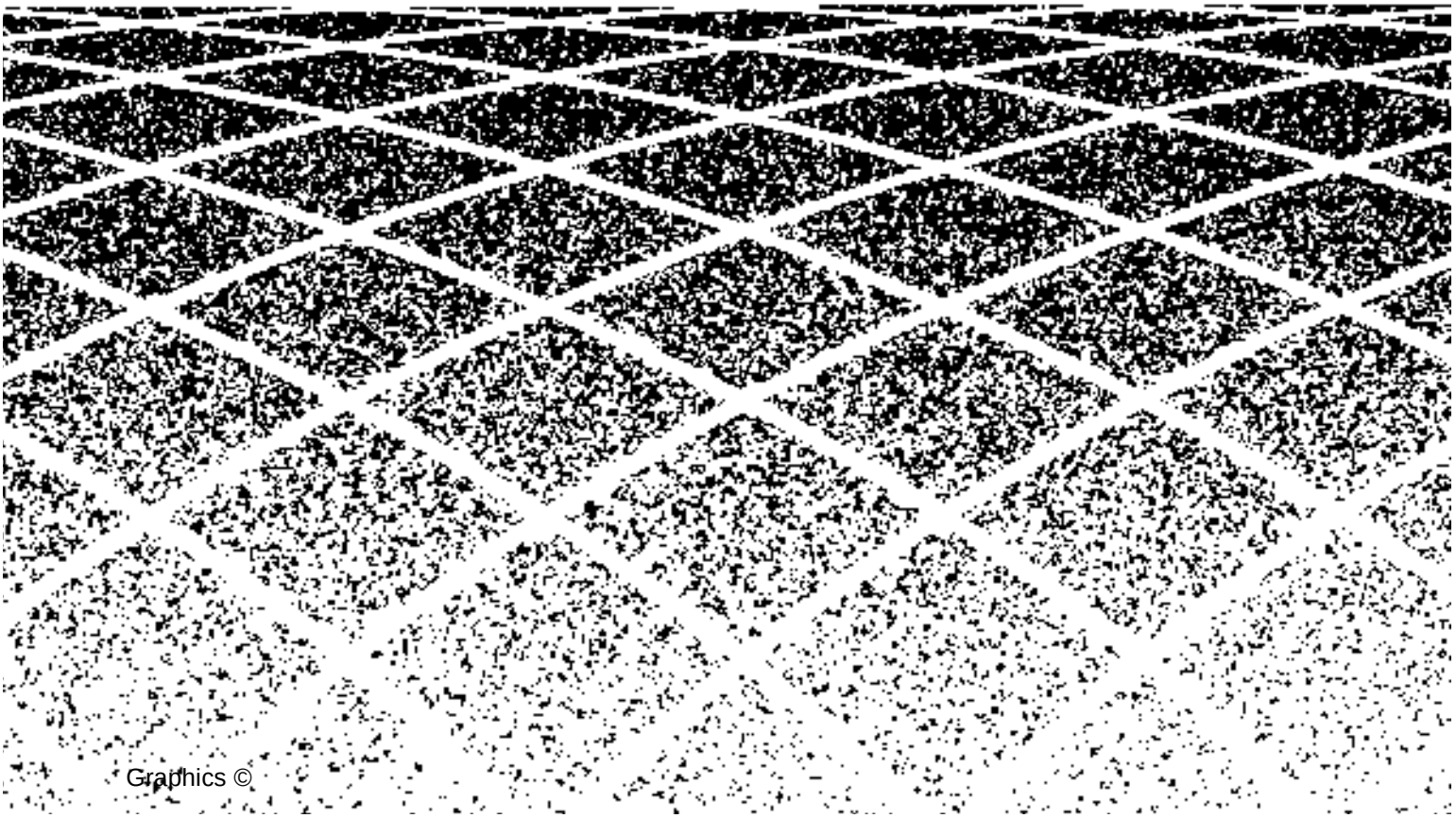
585-350-814
Issue 1
October, 1993

CONVERSANT VIS Version 4.0 FlexWord Speech Recognition



Blank Page

Contents

Contents

Contents

Contents

About This Document

Purpose

This document, *CONVERSANT VIS Version 4.0 FlexWord Speech Recognition*, 585-350-814, describes the procedures for installing and administering the WholeWord Bilingual Speech Recognition package, an optional package which works with the CONVERSANT VIS 4.0 platform.

Intended Audience

This document is primarily intended for customers. Secondary audiences include the following: field support, customer support, and test personnel. This book can be used as a reference for obtaining specific information about various FlexWord features, software, and hardware.

How to Use This Document

This book is organized in five chapters, and each chapter contains the following information:

- Chapter 1, "FlexWord Speech Recognition", contains an overview of the FlexWord speech recognition feature and its different capabilities.
- Chapter 2, "Installing FlexWord Hardware and Software", details how to load the software package onto the platform and how to assign speech recognition hardware resources. There is also a section on how to remove the software package.
- Chapter 3, "Using FlexWord Speech Recognition with Script Builder", explains ways in which Script Builder can be used to write FlexWord applications.

- Chapter 4, "Using Speech Recognition in Scripts", contains changes to existing script instructions as well as new script instructions for speech recognition.
- Chapter 5, "Summary of FlexWord Script Instructions", summarizes the script instructions which are specifically related to speech recognition in Version 4.0.
- Appendix A, "FlexWord Speech Recognition", contains tables.

Conventions Used in This Document

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

  Enter **y** to continue.

- Two or three keys that you press at the same time (that is, you hold down the first key while pressing the second and/or third key) are shown as two separate rounded boxes connected together by “and”. For example, an instruction to press and hold ALT while typing the letter d is shown as

  ALT and D

- Commands and text you type or enter appear in **bold**.
- File and directory names appear in **bold**.
- Values, instructions, and prompts that appear on the screen are shown in traditional typewriter type as `constant-width`

Trademarks and Service Marks

The following trademarked products are mentioned in this book:

- CONVERSANT® is a registered trademark of AT&T.
- FlexWord™ and ScriptBuilder™ are trademarks of AT&T.

Related Resources

The following books are expected to be used in conjunction with this book:

- The hardware installation manual for your platform
- *CONVERSANT VIS Version 4.0 Maintenance*, 585-350-112

  The maintenance guide contains detailed information on troubleshooting and replacement procedures for speech recognition hardware.
- *CONVERSANT VIS Version 4.0 Software Installation*, 585-350-111

  The software installation guide offers in-depth information about software installation and necessary system configurations.

How to Comment on This Document

A reader comment card is located behind the title page of this book. We are interested in your suggestions for improving this book and urge you to complete and return a reader comment card.

If the reader comment card has been removed, send your comments to:

AT&T
Product Documentation Development Department
Room 22-2C11
11900 North Pecos Street
Denver, Colorado 80234

Please include the name and the order number of this book.

FlexWord Speech Recognition

1

Introduction

FlexWord recognition processes and types, features, necessary hardware and software, and factors which influence recognition accuracy are detailed in the following pages and chapters.

Overview of FlexWord Speech Recognition

The “Overview” section explains how FlexWord speech recognition works and describes ways that speech recognition can be used to enhance your organization’s telecommunication transactions.

What is Speech Recognition?

Speech recognition is a CONVERSANT hardware and software feature package which allows callers to speak or enter touch-tone requests over the phone during an application transaction. FlexWord speech recognition requires an installed and operational FlexWord speech recognition package, as well as various pieces of hardware including signal processing (SP) and Companion (CMP) circuit cards.

Speech recognition is offered as either sub-word recognition or whole-word recognition:

- Sub-word speech recognition technology relies on phonemic recognition. The English language is made of approximately 40 phonemes. These phonemes are blocks of sound which, when strung together in particular orders, form recognizable words. (The word “one,” for example, consists of three phonemes: “w-uh-n.”) Sub-word technology analyzes and recognizes words according to their phonemes.
- Whole-word technology recognizes “whole” words, not phonemes or parts of words. Thus, for each word to be recognized, thousands of samples of that word are gathered, and incoming speech is compared to mathematical models made from the word samples.

The FlexWord Speech Recognition package relies on sub-word technology and provides customers with a cost-effective way of designing large, customized vocabularies and menu options. FlexWord is ideal for word/phrase intensive applications. For example, a name dialer, an application which allows employees to speak the name of another employee instead of dialing a telephone extension could be designed using FlexWord. Since names can be built from phonemes, whole-word data collection processes are not necessary. The WholeWord Bilingual Speech Recognition package uses whole-word technology, and it is best suited for number intensive bilingual applications. For example, a banking application, which requires callers to enter checking account number digits and respond “yes or no” to prompts, would benefit from WholeWord recognition features.Applications which combine WholeWord and FlexWord capabilities offer callers the opportunity to enjoy the convenience of speaking numbers as well as preplanned words and phrases.

How Do the FlexWord and WholeWord Bilingual and Speech Recognition Packages Work?

A speech recognition transactions begin when a caller dials the CONVERSANT Voice Information System (VIS). The VIS answer the phone with a greeting, questions the caller with a prompt, collects information from the caller, and then directs the call, according to this prompt and collect action and the caller's request. If the caller begins by entering touch-tones, the VIS will assume no speech recognition resources are required for this particular prompt. However, if the caller begins by speaking a "target" word, digit, or phrase (the response for which the application designer is prompting), the VIS locates a free signal processing (SP) resource to handle the speech recognition.

The SP card, which contains the speech recognition software, is, in effect, the "recognizer". It compares the incoming speech sample to other word "models," which are mathematical patterns representing specific words that have been built after gathering thousands of samples of the target word or phrase. For WholeWord, these word models are then grouped together as desired, and this patterned group of words is called a "grammar".

For FlexWord, there are no grammars per se. FlexWord matches spoken input to the feature's provided speech recognition algorithms which correspond to specific words on a "wordlist," a predesignated group of words which could be possible answers to a prompt. These wordlists can be used by any of your FlexWord applications. (All of the wordlists you specify for an application will be constructed by AT&T's speech recognition technical staff.)

In both the case of FlexWord and WholeWord speech recognition, each word which possibly could match the caller's request gets a "score". CMP card(s) are responsible for scoring the probable matches. Once the word with the highest score has been determined, it is returned to the script. For WholeWord, if nothing matches the spoken input, the script will reject the input, and the script should reprompt the caller to give her or him another chance to speak a request. For FlexWord, the script will return the closest match from the wordlist, so it's important that callers are given the chance the confirm their requests.

WholeWord language packages offer capabilities that are unavailable through the FlexWord packages. They are as follows:

- ***Barge in***. Callers do not have to wait until a prompt is finished before speaking. If they know where they want their call directed, they can begin to speak immediately after the recognizer becomes active.
- ***Word-spotting*** capabilities. If a caller utters more than the target word, WholeWord will extract the target word from the extraneous phrase.
- ***Bilingual recognition***. The package can recognize two supported languages.

In addition, some WholeWord language packages have connected-digit recognition.

NOTE:

Connected-digit recognition. WholeWord allows callers to string together digits, for example, a caller can say his or her account number, “one-three-two-four,” and it can be recognized as “1324”. Please note that at this time FlexWord does not support alpha-string or alpha-numeric-string recognition. Thus, FlexWord cannot recognize, for example, a string comprised of letters from the alphabet, “abcde,” or of numbers combined with letters from the alphabet, “a2b3c4.” Any digit recognition (0-9) should be done using WholeWord.

Getting Started with FlexWord Speech Recognition

The following procedure gives examples of how you can begin to prepare a FlexWord application:

1. First, draft a design of your desired FlexWord application. When a caller responds to a prompt, she or her will answer with specific words. You must define the words you wish the system to recognize and group these words into “wordlists”. Each prompt should have its own wordlist. For illustrative purposes, look at Figure 1-1. In this banking example, each menu prompt has a corresponding wordlist. The first menu prompt looks to the “Information” wordlist to verify the caller’s first request. The second menu prompt looks to the “Loan” wordlist to verify the caller’s second request.

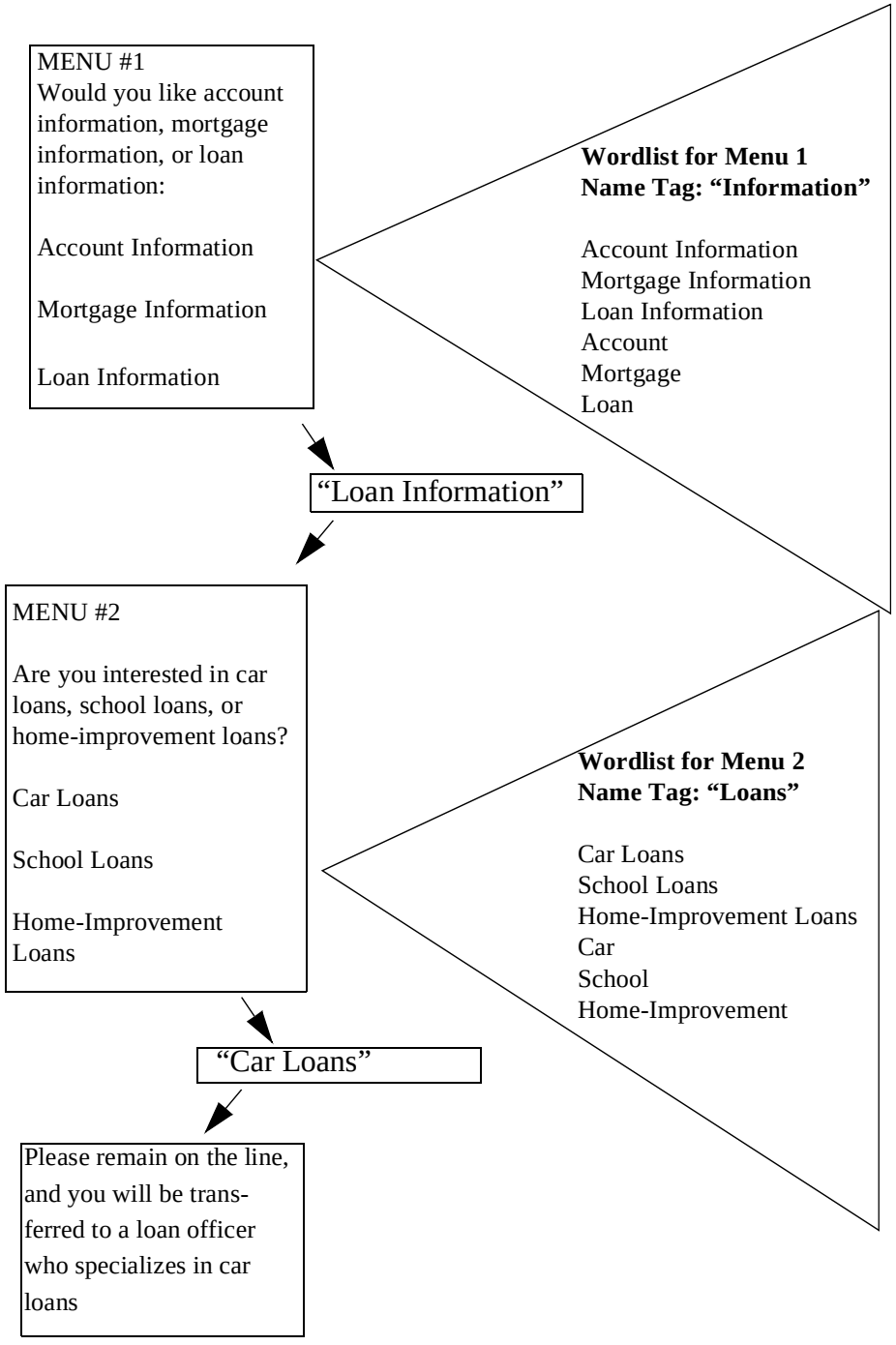


Figure 1-1. FlexWord Application Example with Menu Prompts

The following figure, Figure 1-2, illustrates how wordlists can be constructed for applications using open-ended prompts (those prompts which do not offer a list of choices) and menu prompts. In Figure 1-2, both open-ended and menu prompts are used to collect information from the caller.

.

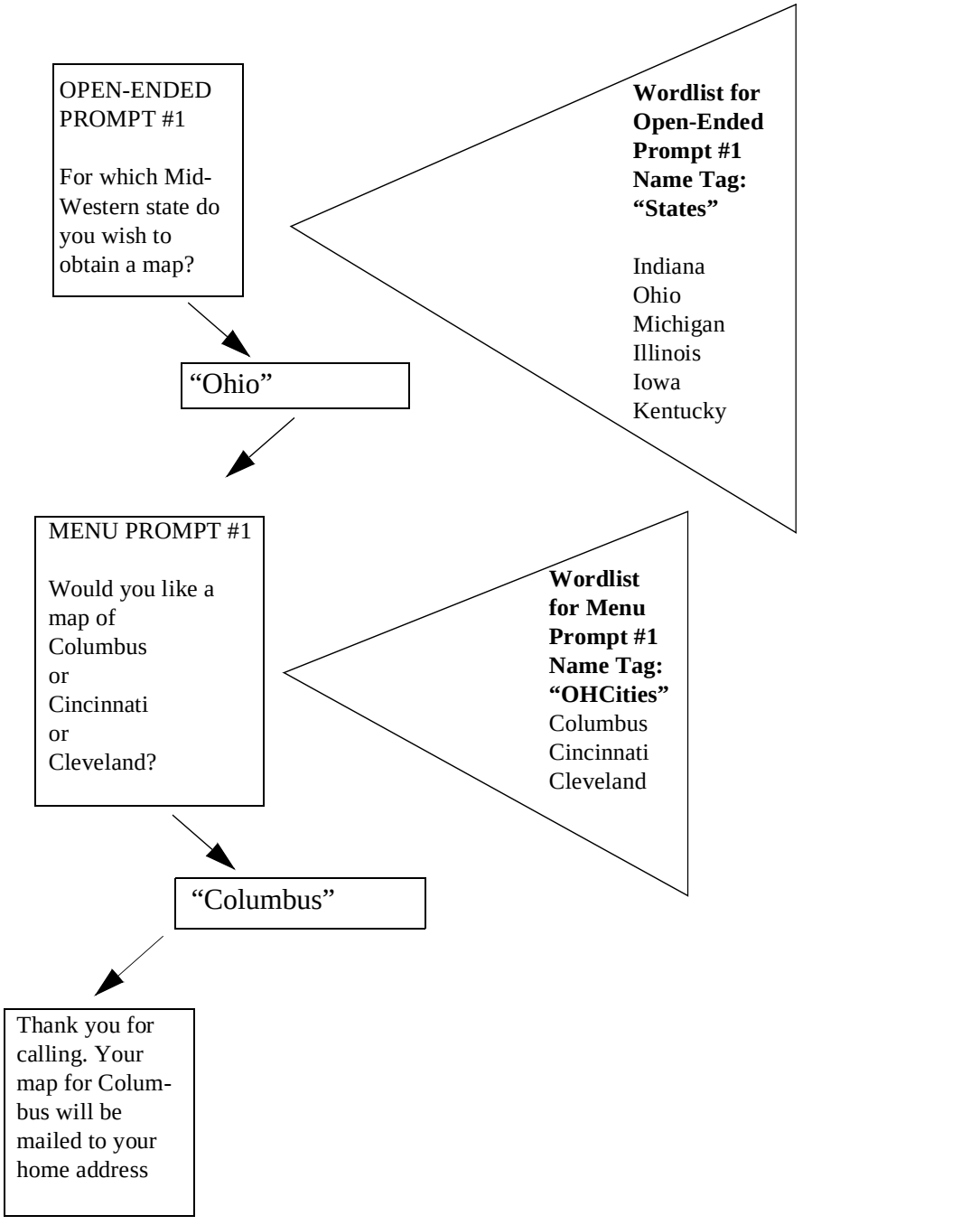


Figure 1-2. FlexWord Application Example Using Open-Ended and Menu Prompts

The wordlists in both Figure 1-1 and Figure 1-2 contain groups of possible words from which callers can choose. One application may use several wordlists. You must name each wordlist, since after callers speak into the phone, this feature will look to the one wordlist which you have designated by name in the Mode field of the Prompt and Collect screen.

All of the wordlists for all of the applications you design with this package constitute the “vocabulary”.

2. After deciding upon and naming your wordlists, contact AT&T’s FlexWord technical support staff by phoning (404) 242-1551. Inform us of your wordlists and vocabulary, and we will work with you to customize your vocabulary package. We will also try to answer any questions you might have.
3. When the customizing process is complete, AT&T will send you your Flex-Word package with your customized wordlists.
4. Now you can begin to implement your application.

Hardware and Software Requirements

The FlexWord feature requires a standard CONVERSANT platform with a signal processor/companion card set (SP/CMP). Please check to make sure that all of the required hardware and software components are in place before attempting to install FlexWord.

Signal Processing and Companion Card Specifications

FlexWord recognition is performed by a CMP, under the control of an SP card. Each available CMP supports 4 simultaneous channels of recognition. For 8 channels of speech recognition, 1 dedicated SP and 2 CMPs are needed.

1 SP + 1 CMP = 4 Channels

1 SP + 2 CMPs = 8 Channels

NOTE:
If your system is concurrently running AT&T's WholeWord speech recognition package, please be sure to dedicate separate SP and CMP cards for FlexWord. FlexWord does not support the WholeWord barge-in feature, and FlexWord *cannot* share SP cards with any other speech function. To play voice prompts, you will need an SP board separate from the hardware resources being used for FlexWord recognition or IVP Tip ring boards, which are set to play their own prompts.

Words and Wordlist Specifications

As mentioned in the "Overview" section, "wordlists" are specific groupings of words and/or phrases that function much the same way as WholeWord speech recognition types. After receiving spoken input, the FlexWord recognition algorithm scans the appropriate wordlist and generates a group of candidates that most closely resemble the caller's response. The algorithm returns the most likely match to the script. Since FlexWord does not have garbage rejection or misrecognition capabilities, FlexWord can not reject bad input. The feature will *always* try to match the caller's request with a word from the specified wordlist. Defining your wordlist is very important for application success.

The maximum number of words or phrases which can be loaded onto the FlexWord recognizer is 2000. FlexWord supports 100 wordlists. However, *every* word or phrase in a wordlist is counted as a distinct and separate word. If the word "help" appears in 10 of your wordlists, "help" will be counted as 10 separate words. However, phrases, such as "Loan Information," count only as one word. Each wordlist can contain 200 words, and each must be given a name tag consisting of 1-14 upper-case characters.

FlexWord's Capacity Maximums	
Words/phrases	= 2000
Wordlists	= 100
Words per word list	= 200
Characters per wordlist name tag	= 14

Finding Wordlist Information

The directory **/att/asr/wordlists** contains a file for each of the application's wordlists. Wordlist files are named according to wordlist name tags; thus, a wordlist with the name tag "Loans" will be listed as the file **Loans** in the Mode field of the Prompt and Collect screen. The actual content of each wordlist file includes the English spelling and the phonetic breakdown of all of the words on the specific wordlist. The words "Account Information," "Mortgage Information," and "Loan Information," for example, would be included in the **Loans** wordlist file in the wordlist directory.

Using Wordlists in Scripts

On the second screen of your Prompt and Collect action, you must specify the name of the wordlist to be used for this prompt in the Mode field. Please note that FlexWord scans a specified wordlist; it can *not* scan for single words. For example, if you give a caller the menu choice “account information” and he or she instead says “account,” this utterance may not be recognized. For this example, it is important to have both “account information” and “account” on the wordlist. FlexWord and WholeWord speech recognition can be used within the same script; however, each Prompt and Collect action must specify FlexWord or WholeWord. Both packages can not be performing at the exact same time. To see a model of a retail application example which combines FlexWord’s wordlists and WholeWord’s “yes/no” speech recognition feature, see Figure 1-3.

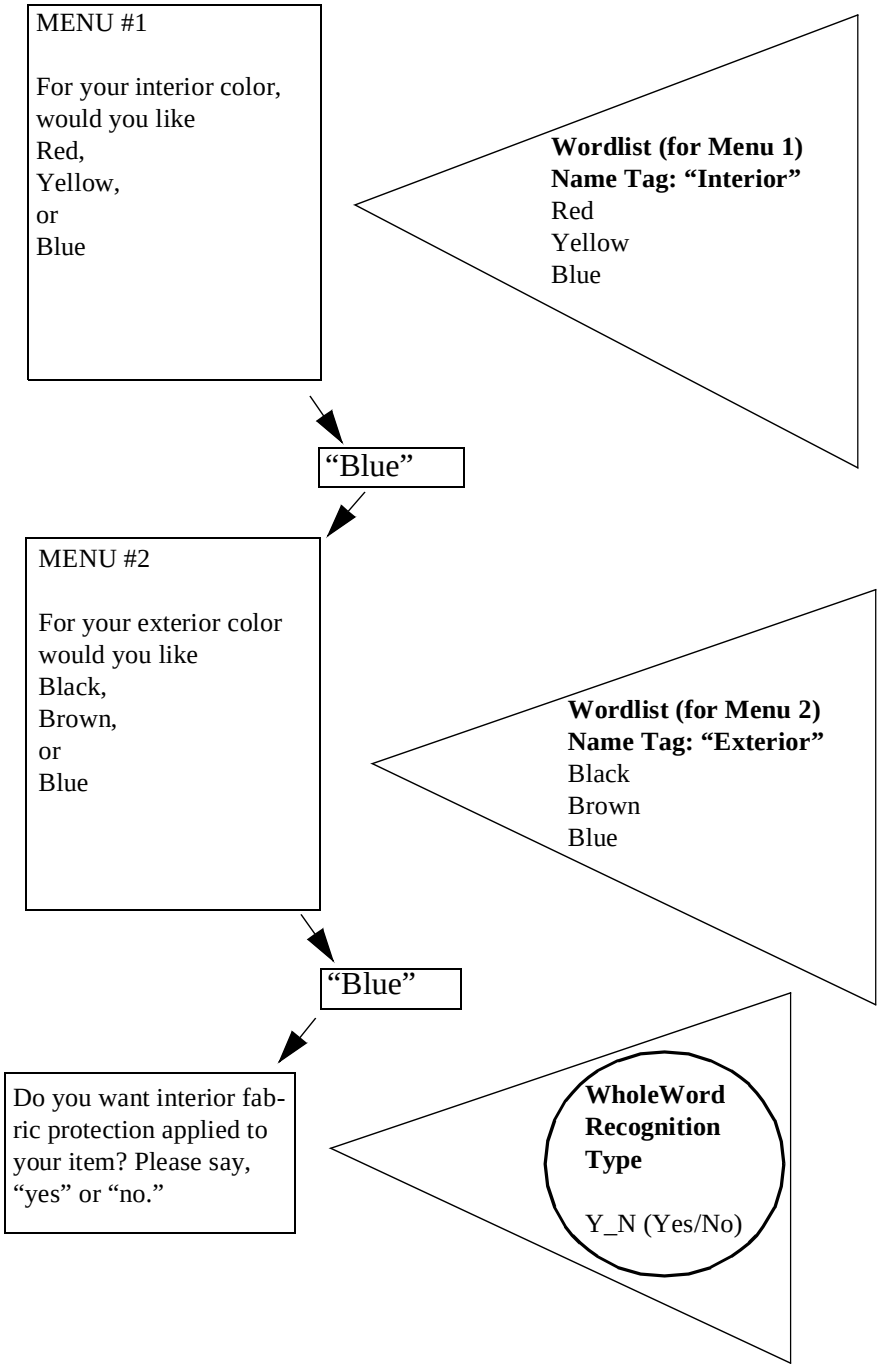


Figure 1-3. Example with FlexWord and WholeWord

In the Figure 1-3 application, the color “blue” must appear on both the “interior color” wordlist and the “exterior color” wordlist. Since the word “blue” appears on two wordlists, it counts as 2 words. If the word “blue” is not on the “interior color” wordlist, this feature *will not* look to the “exterior color” wordlist to find it. The feature will return the word from the wordlist that it thinks is the most suitable match. In that the feature can only look at one wordlist per Prompt and Collect screen, it is of the utmost importance that each wordlist contains *all* possible choices for a single prompt.

NOTE:

It is possible to consolidate wordlists. For example, if the first Prompt and Collect screen allows for colors “red” and “blue” and the second Prompt and Collect screen allows for choices “green” and “purple”. You can put “red, blue, green, purple” on the same wordlist and direct the application to look at the wordlist with the name tag “Colors” for both prompts. Then, if the caller chooses “green” at the first prompt (which only allows for choices “red” and “blue”), the script could evaluate “green” as an unacceptable choice and you can reprompt the caller.

The following sections contain information concerning the performance of the FlexWord feature on the VIS. Included are helpful hints and tips for using the feature.

Accuracy

Given the variance in human speech and the statistical properties of recognition algorithms, the speech recognizer may make errors. Speech recognition accuracy depends not only on the recognition algorithms, but also on prompts, calling populations, the words to be recognized, and application designs.

Factors which positively influence accuracy rates include effective wordlist and prompt construction as well as educated calling populations. Low recognition accuracy rates are usually caused by inexperienced callers, who say extraneous phrases and/or speak before the prompt is finished, and wordlists which contain short words, rhyming words, or a large number of words.

Positive Influences on Recognition Accuracy

The following items factor into positive recognition accuracy:

- ***Effective Wordlists***. Choosing word for your wordlist which have different sounds and are of medium length will help to increase recognition accuracy. In the case of name dialers, applications which allow employees to speak a colleagues name rather than enter an extension number, using the last name and the first name on wordlists increases recognition accuracy. In other types of applications, syllabic and vowel similarities may contribute to recognition confusion. For example, “women’s wear” and “men’s clothing” are more effective wordlist phrases than “women’s clothing” and “men’s clothing.” The latter pair sound too much alike; both have “men” embedded in the word and both share the word “clothing”. Please pay special attention to the words your customers actually use when they ask for a service. If you are automating an existing transaction which has in the past taken place between a customer and a service professional, use your service professionals as resources and try to mimic customers’ request words.
- ***Experienced calling population***. Recognition improves for applications in which the calling populations are closed, and the callers are experienced with and/or trained to interact with the application.
- ***Prompt Structure and Design***. Prompts offered in calm, clear voices greatly affect recognition accuracy as do the specific structures of the prompts. Prompts should guide the caller to say desired words/phrases.
  - — Prompts which are set up to dissuade callers from barging-in or speaking before the prompt is finished increase recognition accuracy. For example:

    “Please say the month of your birth, now.”

— Menu prompts. For best results, menu prompts should be built with the structure: <desired result> <action required>. For example:

"To hear your checking account balance, say 'checking.'

To hear your savings account balance, say 'savings.'"

By placing the action required at the end of the prompt, the caller is able to remember the <action required> of her or him. Do not list the <action> before the <desired result>. For example, the prompt: "Say 'one' for a description of the up-coming gallery events," is a bad prompt. It encourages the caller to forget the specific <action required>, since the last thing related is the <desired result>.

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

  "You said 'swordfish.' Is this correct? Please say 'yes' or 'no.'"

  Thus, the caller can make sure that her/his word matched the returned recognition word. See Figure 1-4 for an illustration of a confirmation path.

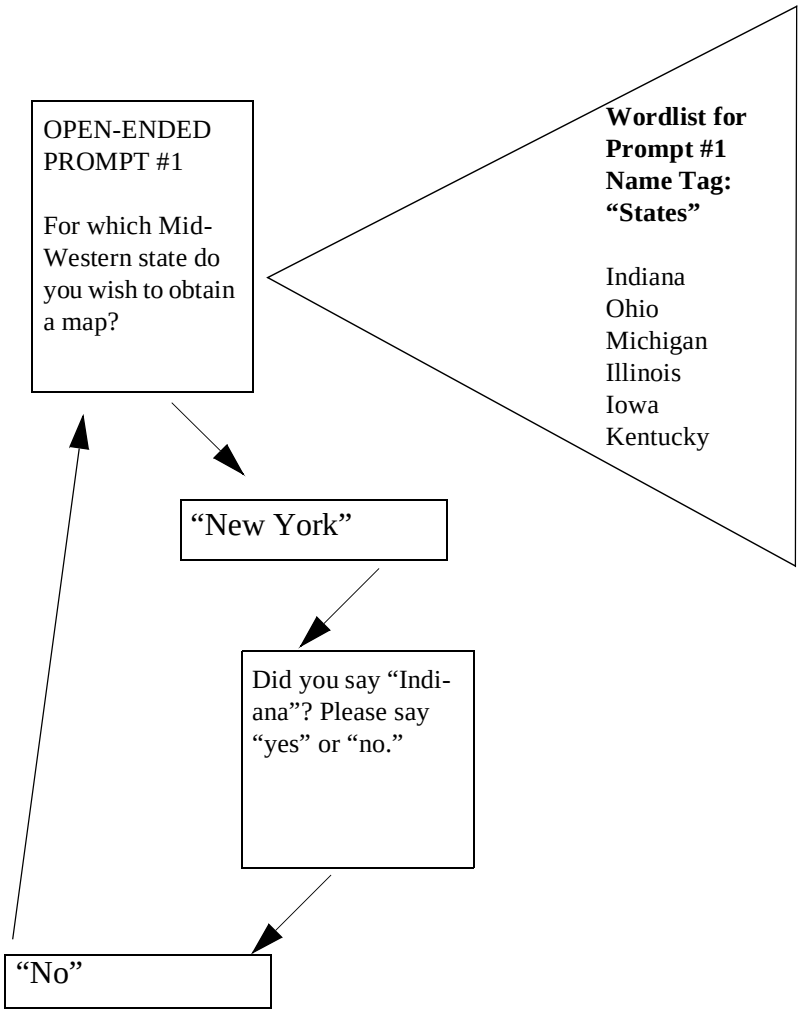


Figure 1-4. FlexWord Application Example of a Reprompt

- ***Informative prompts***. Lengthy prompts providing detailed response instructions may improve recognition accuracy. But generally, from the perspective of frequent users, these types of prompts are not optional, since experienced users will have to listen to the complete prompt before responding. One solution may be to provide more informative prompts for first-time callers only. However, for applications with infrequent users, lengthy prompts may be more acceptable and help improve the accuracy.

Negative Influences on Recognition Accuracy

Recognition rates are lower when callers barge in before prompts are finished, since FlexWord does not support barge in, and/or when callers speak extraneous words and phrases. Also, callers who do not clearly enunciate their words will experience lower recognition accuracy. The following items have a negative influence on recognition accuracy.

- ***Callers Speaking Extraneous Words***. FlexWord does not have word-spotting capabilities; the feature can not pick a target phrase out of a stream of spoken words. The following example illustrates an accuracy problem caused by extraneous speech:

  PROMPT: “Please say ‘checking account’ or ‘savings account, now’”

  CALLER: “I want to find out my checking account balance.”

  Since the feature can not spot “checking account” in the callers request, it will return a phrase from the word list, and this phrase will probably not be “checking account”. Make sure that prompts encourage callers to say one isolated word or phrase. FlexWord is not capable of screening out improper or extraneous speaker input.

  NOTE:
  If you expect that your callers may say “I don’t know,” “What am I supposed to do?” or ““Drat!” think about including these phrases on your wordlists. Then if the script recognizes one of these phrases, make sure it reprompts the caller by saying, for example, “Sorry, please choose savings or checking.”Anything on your word list may be returned to your script as a recognition response. So write your prompts accordingly, and be sure to allow for callers to confirm their entries. Regardless of how obscure caller input may be, the feature will attempt to match the input with a word on the wordlist. For example, the prompt may ask: “Choose a color—red, blue, or yellow.” If the caller responds, “I hate talking on the telephone,” the script could reprompt: “Yellow. If yellow is the color you wish say ‘yes.’”

- ***Callers Barging-In Before the Prompt is Finished***. FlexWord recognition *does not* support barge-in capabilities. Thus, prompts should include some sort of time reference so that callers know when to respond. The first prompt example shows how recognition accuracy can be increased by including a time reference:

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

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

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

  The second prompt example does not have the time reference "now." Without the "now" at the end of the sentence, the caller may speak before the feature is ready to listen.

  NOTE:
  If an application is making use of FlexWord, you must specify **no** at the "Speak with Interrupt" on the Prompt and Collect screen. FlexWord can not support the WholeWord barge-in feature, so setting "Speak without Interrupt" to **yes** will cause problems.

- ***Environment***. Noisy environments, such as an airport, train yard, or an unclear cellular phone connection may contribute to recognition accuracy problems.

- ***Ineffective Wordlists***. Ineffective wordlists cause lower recognition accuracy. Effective wordlists contain different sounding words of medium length. In general, the larger the wordlist, the lower the expected recognition accuracy.

  — Short words. One syllable words which have the same vowel sounds are more difficult to recognize. For example, "on" and "off" both share the short "o" sound. (Long "o" vowel sounds appear in words like "oh" and "no"—words in which you can hear the letter "o".) Short words should be used in moderation.

— Rhyming Words. Wordlists which contain words with similar rhyming vowel sounds can cause a decrease in recognition accuracy. Please look to the examples in Table 1-1 and make sure that you choose distinct sounding words when creating wordlists. Again, the VIS is matching sounds and stringing these sounds together. Pronounce the words on your wordlists and vary the vowel sound of each word if possible.

Table 1-1. Examples of Vowel Sounds

Long "A"	Short "A"	Long "E"	Short "E"	Long "I"	Short "I"	Long "O"	Short "O"	Long "U"	Short "U"
paper	fatten	peaceful	metal	spicy	trigger	postal	foggy	cupid	grunting
traitor	dancer	receipt	header	tiger	kitty	soda	robber	viewer	runner

— Wordlist Size. As the size of wordlists increase, accuracy decreases. The best accuracy results can be achieved by structuring an application to make use of several smaller wordlists rather than one large wordlist.

Installing FlexWord Hardware and Software

2

Introduction

This chapter contains information on installing FlexWord hardware and software.

NOTE:
This chapter is a supplement to *Voice Processing Hardware Installation and Upgrade*, which specifically addresses your platform. If you wish, you may insert this chapter at the back of that book behind the tab labeled ''Optional Feature Packages.''

Installing FlexWord Hardware

For information on installing the companion card (CMP) or signal processing card (SP), refer to Chapter 7, ‘‘Installing CONVERSANT Circuit Cards,’’ of the Voice Processing Hardware Installation and Upgrade document for the appropriate platform. To make sure that the switch settings are correct, see "Companion Card Switch Settings" later in this chapter.

Companion Card Switch Settings

A CMP card is a signal processing card (SP) extension that helps the SP to process information. The SP and CMP cards are connected through a specialized bus.

The CMP card is shown in Figure 2-1 with the location of its jumper and switches. The required switch settings are shown in Figure 2-2 through Figure 2-3.

NOTE:
It is possible to have 1 SP supporting either one or two CMPs. The CMP(s) connected to the SP(s) must always start addressing with card 0. For example, if there are four SPs, each with one CMP, each CMP will be addressed as card 0.

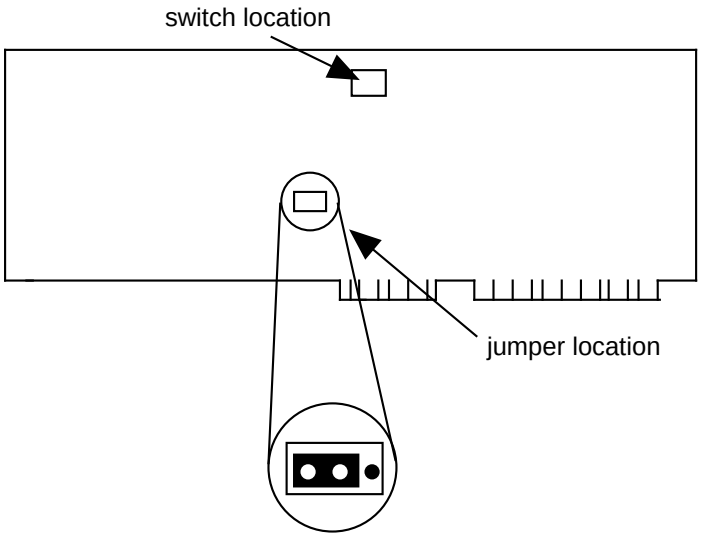


Figure 2-1. Companion Board Jumper and Switch Locations

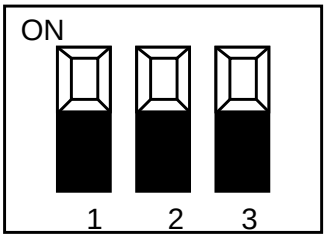


Figure 2-2. Companion Board 0 Switch Setting

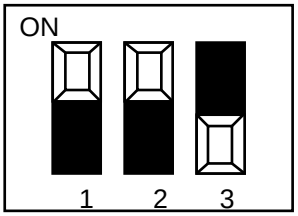


Figure 2-3. Companion Board 1 Switch Settings

Installing and Removing FlexWord Software

If you are installing the FlexWord feature package as part of the initial software load, refer to Chapter 4, ''Installing Software for Optional Features,'' of *CONVERSANT VIS Version 4.0 Software Installation and Upgrade*, 585-350-111

If you are installing the FlexWord feature package on a running system, follow the procedures to stop the VIS, then load the software.

Stopping the VIS

1. From the CONVERSANT VIS Version 4.0 screen, highlight Voice System Administration, then press ENTER.
2. From the Voice System Administration screen, highlight Configuration Management, then press ENTER.
3. From in the Configuration Management screen, highlight System Control, then press ENTER.
4. From the System Control screen, highlight Stop Voice System, then press ENTER.
5. Press CANCEL until you are out of the menu windows, and the system prompt is displayed.

Loading Software for FlexWord

1. Insert the first disk of the FlexWord feature package set into the floppy disk drive.
2. Enter **installpkg**

The instructions displayed on the screen will tell you when or if to insert subsequent disks. When the installation is complete, you may either restart the VIS, or continue with other tasks.

Removal of FlexWord Software

Before you remove the FlexWord software, make sure that the FlexWord functionality is not being used by any SP cards in the system. (See the next section, "Assigning FlexWord Functionality to a Card", for more information.)

1. Stop the VIS using the procedure "Stopping the VIS" given earlier in this chapter.
2. Enter **removepkg**

   The system will respond by displaying a numbered list of installed packages.
3. Enter the number associated with the FlexWord software package that you wish to remove.

When the prompt is returned, the FlexWord software package has been removed.

Assigning FlexWord Functionality to a Card

In order for this feature to operate properly, you must assign FlexWord functionality to an SP card.

NOTE:
Make sure the SP card is in the MANOOS state before beginning this procedure. For more information about changing the state of the card, refer to Chapter 3, ''Configuration Management,'' of *CONVERSANT VIS Version 4.0 Operations*, 585-350-703.

1. From the Voice System Administration window, highlight Configuration Management, then press ENTER.
2. From the Configuration Management window, highlight Voice Equipment then press ENTER to open the Voice Equipment window.
3. Press CHG-KEYS to display the alternate set of function keys. Use the key ASSIGN to assign functions to SP cards as well as to change the current functions assigned to SP cards.
4. From the Voice Equipment window, press ASSIGN to display the Assign screen. The key ASSIGN assigns functions to SP cards as well as changes current functions assigned to SP cards.
5. At the Assign screen, highlight Functions to SP Cards, then press ENTER to display the Assign Functions to SP Cards screen (See Figure 2-4).

6. At the Assign screen, highlight Functions to SP Cards, then press ENTER to display the Assign Functions to SP Cards screen (Figure 2-4).

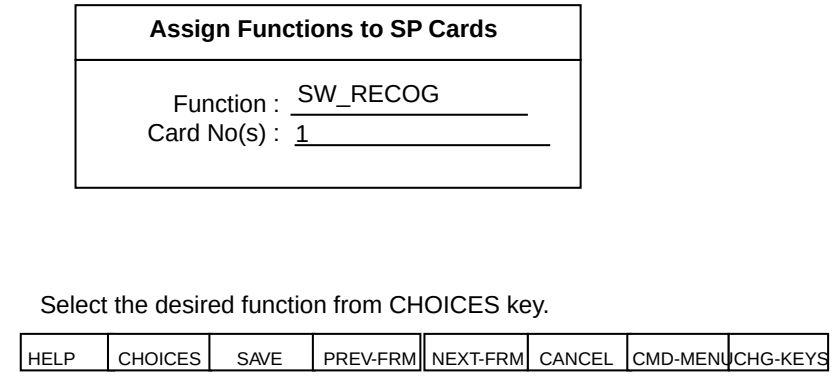


Figure 2-4. Assign Functions to SP Cards

Assignments made in the Assign Functions to SP Cards screen overwrite all current assignments on the specified cards. Take care when making assignments and re-assignments.

7. In the Function field, enter **SW_RECOG** to assign FlexWord functionality to an SP card. SW_RECOG allocates that SP for sub-word recognition only.
8. In the Card No(s) field, specify the card number of the SP that you wish to assign SW_RECOG functionality.

   Figure 2-4 has SW_RECOG assigned to card ‘‘1.’’

9. Press SAVE to save the entered values.
10. Press CANCEL once to return to the Voice Equipment screen.
11. Press CHG-KEYS to display the alternate set of function keys.

    Use the CHGSTATE key to change to the state of the SP card to INSERV.
12. Press CHGSTATE to display the Change State of Voice Equipment screen (Figure 2-5).

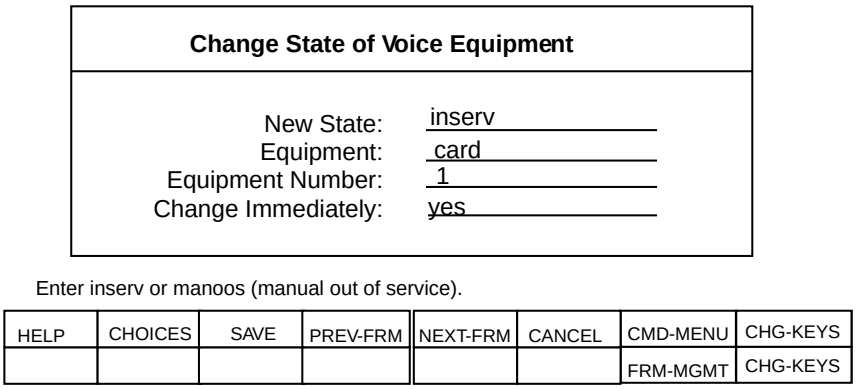


Figure 2-5. Change State Voice Equipment

13. In the New State field, enter **inserv**.
14. In the Equipment field, enter **card**.
15. In the Equipment Number field, enter the number of the SP card.
16. In the Change Immediately field, enter **yes**.
17. Press SAVE to close the window and execute the options specified.

    You will receive a message indicating the SP card has been placed in service.
18. Enter **display card *<sp card number>*** to check the state of the CMP card. If the state shows ''Not_diag'' (not diagnosed), proceed to the next step in this procedure. If the CMP card is in service, you have completed assigning FlexWord functionality to the SP card.

NOTE:
The ''Not_diag'' state occurs only when you first install a CMP card. The following steps (Steps 18-24) are not always necessary.

19. Press CANCEL to exit the Voice Equipment screen and to return to the Configuration Management screen.
20. From the Configuration Management screen, highlight System Control, then press ENTER.
21. From the System Control screen, highlight Diagnose Equipment, then press ENTER to display the Diagnose Equipment screen (Figure 2-6).

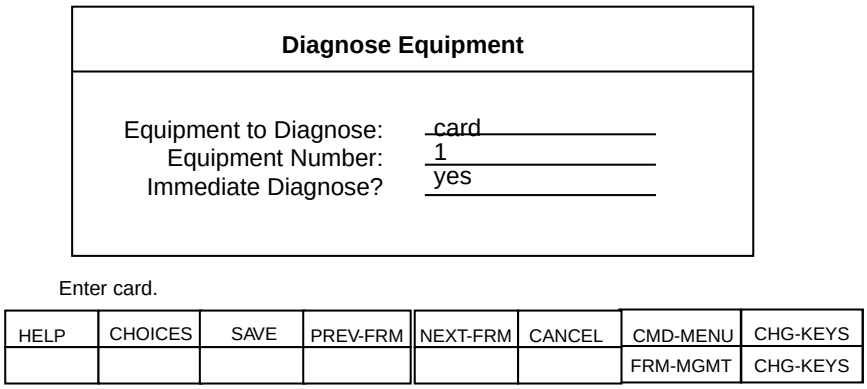


Figure 2-6. Diagnose Equipment Screen

22. In the Equipment to Diagnose field, enter **c** to specify card.
23. In the Equipment Number field, enter the SP card number obtained from the Voice Equipment window. If FlexWord functionality is assigned to the SP, diagnosing an SP card also diagnoses the associated CMP(s). The cards in an SP/CMP cluster cannot be diagnosed independently.
24. In the Diagnose Immediately field, enter **yes**.
25. Press SAVE to save the values you entered and perform the diagnose. Then check to see if the card is in service.

NOTE:
If for some reason the diagnostic fails, please refer to Chapter 2, “Isolating Speech Recognition Troubles,” *in CONVERSANT VIS Version 4.0 Maintenance Guide*, 585-350-112.

Using FlexWord Speech Recognition with Script Builder

3

Introduction

This chapter contains the following information about using the FlexWord feature with the CONVERSANT VIS Script Builder Version 4.0:

- Procedures for "Specifying FlexWord" in a transaction
- A description of external action **"SP_Allocate External Action"** used with FlexWord.

NOTE:
This chapter supplements the *CONVERSANT VIS Script Builder User Guide*, 585-350-704. You may wish to insert this chapter at the back of the user guide, behind the tab labeled ‘‘Optional Feature Packages.’’

Specifying FlexWord

Specifying Input Mode

When the FlexWord software is installed, new options are displayed in the Script Builder menus under Prompt and Collect Caller Input. The Mode field, of the Define Prompt and Collect form (page 2 of 3) now accepts either ‘‘T’’ for touch-tone input or a value to specify a FlexWord wordlist. Figure 3-1 shows this form with an example set of wordlist name tags. Wordlist name tags are typed in all capital letters and are 1-14 characters long. To display these choices from the form, press CHOICES. This is a dynamic menu, which will display all recognition types available on your system, regardless of whether you are planning to use them in the current application.

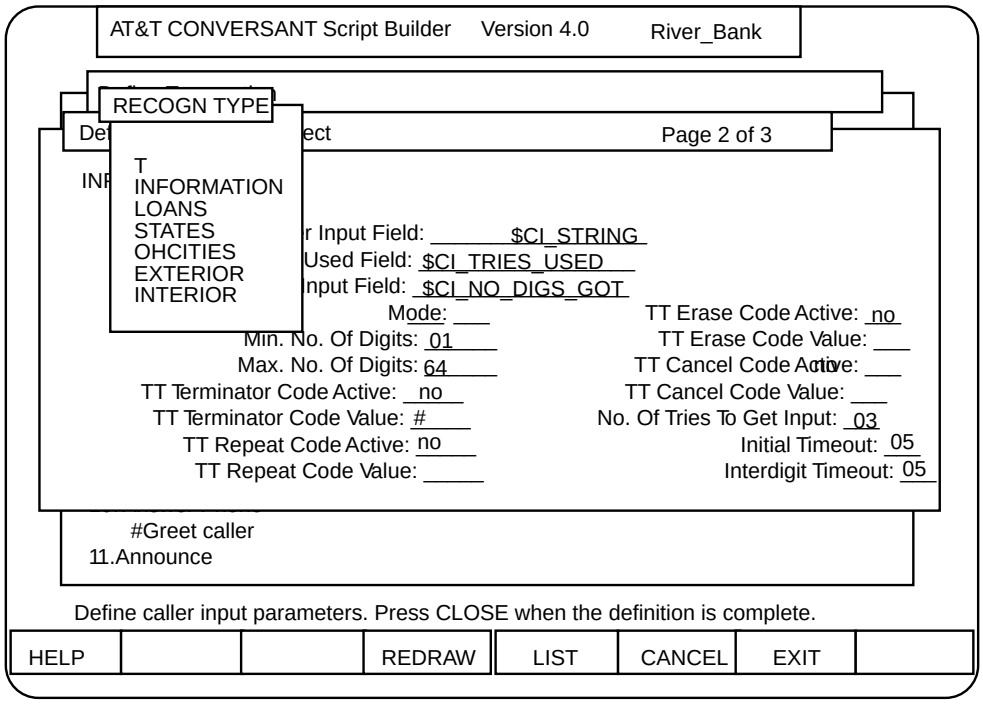


Figure 3-1. Define Prompt and Collect Page 2 - Choices for Mode Field

Script Builder validates the minimum and maximum fields for the speech recognition type selected. The minimum for all FlexWord wordlists is 1 and the maximum is 64. The minimum and maximum fields in this form limit the amount of caller input. If you set the maximum at less than 64, you run the risk of cutting off some of the data returned to the script from the recognizer.

The fields for terminator code, repeat code, erase code, and cancel code only have meaning for touch-tone input. The remaining fields in this form have meaning for speech as well as touch-tone input.

Specifying FlexWord CONFIRM Prompt

ATTENTION: The CONFIRM prompt will only work if the US English WholeWord speech recognition package is installed. For more information, refer to Chapter 3, "Using WholeWord Bilingual Speech Recognition with Script Builder," in *CONVERSANT VIS Version 4.0 WholeWord Speech Recognition*, 585-350-813.

Return Values

$CI_MODE

The $CI_MODE variable is defined for applications that use speech recognition. If you set the minimum number of digits to 1 and the maximum to 64, any value from 1-64 will be returned to the script. When a Prompt and Collect action is performed, the $CI_MODE variable is set equal to or greater than 0 for speech input and to a negative number for touch-tone input. Please note that when defining the variable for speech recognition, the $CI_MODE is set to the SP board number that was used to perform the recognition. $CI_MODE is set only for Prompt and Collect actions that specify speech recognition.

$CI_VALUE

The $CI_VALUE variable indicates the word recognized by the system. For example, if the caller inputs the recognized word "savings," and the maximum amount of digits is set equal to or greater than 7 on the Prompt and Collect Screen 2, the return value will be "savings." However, if the maximum number of digits on the Prompt and Collect screen is set equal to or greater than 3 and a caller inputs the recognized word "savings," the return value is "sav"—the first 3 letters.

SP_Allocate External Action

The script uses the SP_Allocate external action to help ensure that a speech recognition system resource is available when it is needed. Normally, this resource is shared between the transactions running on separate channels. It is allocated and deallocated automatically at each point where speech recognition is used during the transaction. If the recognition resource is being heavily used, a Prompt and Collect action employing recognition may fail with ‘‘Too few digits.’’

The SP_Allocate external action may be utilized before any Prompt and Collect actions to check the availability of the recognition resource. It may also be used to explicitly allocate or deallocate the resource, if it is available, to the transaction.

NOTE:
Care should be taken to design each application so that it will free the speech recognition resource as soon as it is no longer needed by the script. SP_Allocate can tie up the resource when it is being used after it is no longer needed. For additional information about this, refer to ‘‘Number of Supported Channels’’ in Chapter 1 of *CONVERSANT VIS Version 4.0 WholeWord Speech Recognition*, 585-350-813.

Follow the procedure below to select the SP_Allocate external action:

1. In the Define Transaction screen, with the cursor on the line above where you want to add SP_Allocate, press ADD.

   System response:

   The Action Choices menu is displayed.

2. Highlight SP_Allocate, then press ENTER.

   System response:

   The external action is inserted in the transaction below the cursor.

3. Press CANCEL to close the Action Choices menu.

4. Highlight External Action: SP_Allocate, then press DEFINE.

   System response:

   The following form is displayed (Figure 3-2):

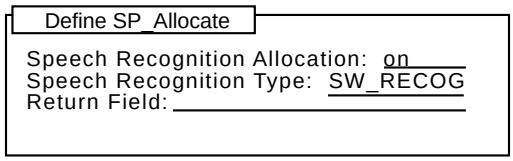


Figure 3-2. Define SP_Allocate Form

5. To enable Speech Recog Allocation, enter **on** in the first field, then press ENTER. To disable Speech Recog Allocation, enter **off** in the first field, then press ENTER. The default value in the field when this form opens is ''on.''

6. An optional return field may be specified in the Return Field: field at this time, then press CLOSE.

   If the field name entered has not been previously defined, the Define Transaction Fields screen opens when CLOSE is pressed. Define the Field Type of the return field as "num," then press CLOSE to exit the Define Transaction Fields window. The return field is set to one of the following values depending on the status of the SP_Allocate:

0	Success
-1	Failure
-2	System resources not available

7. When you press CLOSE the Define SP_Allocate screen is closed, and the Define Transaction screen returns.

8. In the Define Transaction screen, press SHOW to expand the External Action: SP_Allocate action step (Figure 3-3). This example assumes ''RET_VALUE'' was entered in the Return Field: field.

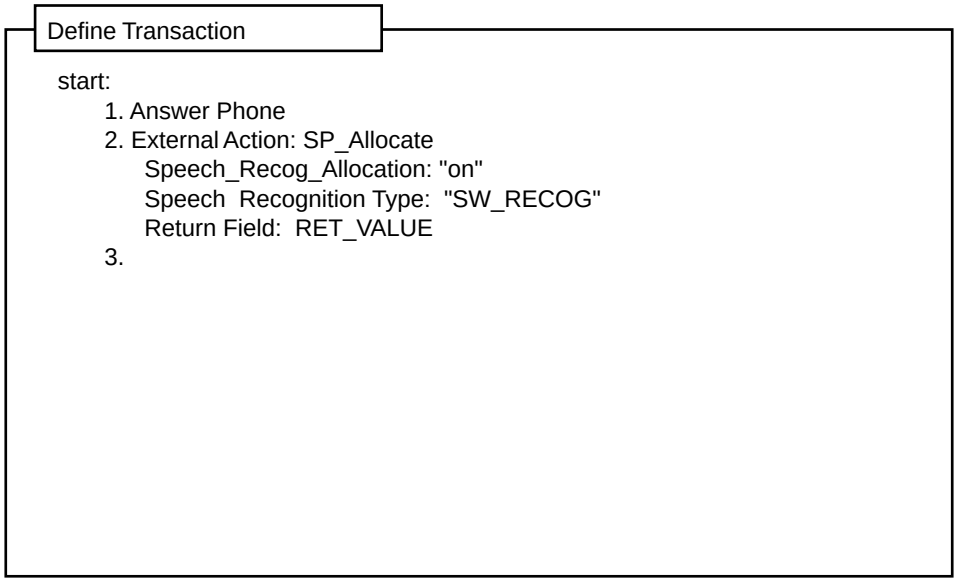


Figure 3-3. Expand External Actions: SP_Allocate Action Step

Using Speech Recognition in Scripts

4

Introduction

This chapter describes changes to existing script instructions as well as new script instructions for speech recognition.

NOTE:
This chapter is a supplement to the *CONVERSANT VIS Version 4.0 Application Development*, 585-350-208. If you wish, you may insert this chapter behind the tab labeled ‘‘Optional Features Packages.’’

Script Instructions for Speech Recognition

The following instructions relate to writing in script language using FlexWord.

The getdig Instruction for Speech Recognition

The generic getdig instruction has some new options for speech recognition arguments. Getdig receives touch-tone or speech information entered by a caller. The getdig format is:

getdig(*type, ctype.dst, number, ctype.mode*)

The first argument, *type*, specifies whether touch-tone or speech input is expected from the caller. Type 0 specifies 12-key telephone touch-tone input. A non-zero value for *type* specifies speech input. The getdig instruction requires the recognition type used for a particular wordlist. The choices available for *type* in this instruction can be found in the **vs/data/sr_file.** The sr_file contains a line for every wordlist installed on your system. If WholeWord is installed, this sr_file will also list all of the grammars on the system. Information in this file will map wordlist names into the symbols that you need to use with this instruction. Therefore, for a specific wordlist, look to the fourth field and use this "defined symbol," which correlates with the wordlist on the same line.

If the *type* argument is 0, the *number* argument specifies the maximum number of touch-tone digits to be received. The maximum value is 128. Received touch tones are stored as a null terminated character string in a buffer specified by the destination argument, *dst*.

If the type argument is other than 0, the number argument specifies the maximum string length for speech input. Received speech input is stored as a null terminated character string in a buffer specified by the destination argument. The characters are defined by the words in the wordlist.

The fourth argument, *ctype.mode*, indicates to the script whether the response is touch-tone or voice. If the response is touch-tone, -1 is stored in *ctype.mode*. If the response is voice, then the number (non-negative) of the SP card that recognizes the voice is stored in *ctype.mode* to be used later by a DIP.

When the **getdig** instruction terminates, a return code is placed in r.0. The following lists show the return values for touch-tone and speech input, where N represents the number of touch tones received.

The return values for touch-tone input and speech input are shown in Table A-1 and Table A-2 of Appendix A.

The sp_alloc Script Instruction for Speech Recognition

NOTE:
For CONVERSANT Version 4.0, only the WholeWord and FlexWord SP resources should be used with the sp_alloc() instruction.

NOTE:
The script instruction **sp_alloc** has replaced the script instruction **sr_alloc**. Since sr_alloc is slowly being phased out, the following warning message will appear on the screen when the sr_alloc script instruction is used:

```
WARNING: sr_alloc is obsolescent. Use sp_alloc.
```

Although the **sr_alloc** instruction will compile, application designers are encouraged to use the **sp_alloc** instruction.

The script instruction, **sp_alloc**, is explicitly used to allocate/deallocate speech recognition on the SP card. The format for the instruction is as follows:

sp_alloc(*type.onoff, type.resource*)

The **sp_alloc()** instruction may be used by a script to allocate the speech recognition resource on the SP card. Normally, this resource is shared by all scripts running on the system, and allocation is done automatically only when the script actually uses the resource. If the SP resource is not available when an instruction that requires it is executed, the instruction will fail. By using **sp_alloc()**, the script may test for the availability of a particular SP resource. If the resource is available, it will be allocated to the script until the script terminates or until the script deallocates the SP resource using **sp_alloc()**.

sp_alloc() may be used to allocate an SP resource for a period longer than the script is actually recognizing speech. Care should be taken to avoid overloading the systems SP facilities, since this can occur if many scripts using **sp_alloc()** are running simultaneously.Script register 0 (r.0) is set to the following values to indicate the status of the **sp_alloc()** execution:

0	Success
-1	Error (sp_alloc already on or off)
-2	System resources not available

The *type.onoff* argument is used to tell sp_alloc() whether to allocate or deallocate resources. Its two valid values are as follows:

1 Allocate the SP resource

0 Deallocate the SP resource

The *type.resource* argument is used to tell the sp_alloc() which SP resource or combination of SP resources to allocate or deallocate. Each SP resource has a unique value. The values for each resource and examples of how resource values can be added are listed in Table A-3 and Table A-4 of Appendix A.

The sr_talkoff Script Instruction for Speech Recognition

WARNING:
*The script instruction **sr_talkoff**, which is used to enable/disable speech recognition during the prompt can not be used with FlexWord.*

Summary of FlexWord Script Instructions

5

Summary of Script Instructions

This section contains summaries of the script instructions specific to the Speech Recognition feature in Version 4.0:

- "getdig"
- "sp_alloc"
- "The sr_talkoff Script Instruction for Speech Recognition"

NOTE:
These pages may be inserted in Appendix A, ‘‘Summary of Script Instructions,’’ of *CONVERSANT VIS Version 4.0 Application Development*, 585-350-208.

getdig

Synopsis

This script instruction receives information entered by a caller.

Command Format

getdig(***type, ctype.dst, number, ctype.mode***)

Description

The generic getdig instruction has some new options for speech recognition arguments.

getdig receives touch-tone or speech information input by a caller.

The first argument, *type*, specifies whether touch-tone or speech input is expected from the caller. Type 0 specifies 12-key telephone touch-tone input. A non-zero value for *type* specifies speech input. The getdig instruction requires the recognition type used for a particular wordlist. The choices available for *type* in this instruction can be found in the **vs/data/sr_file.** The sr_file contains a line for every wordlist installed on your system. If WholeWord is installed, this sr_file will also list all of the grammars on the system. Information in this file will map wordlist names into the symbols that you need to use with this instruction. Therefore, for a specific wordlist, look to the fourth field and use this “defined symbol,” which correlates with the wordlist on the same line.

If the *type* argument is other than 0, the *number* argument specifies the maximum string length for speech input. (The maximum value is127.) Received speech input is stored as a null terminated character string in a buffer specified by the destination argument. The characters are defined by the words on the wordlist.

The fourth argument, *ctype.mode*, indicates to the script whether the response is touch-tone or voice. If the response is touch-tone, -1 is stored in ctype.mode. If the response is voice, then the number of the SP card that recognized the voice (non-negative) is stored in ctype.mode to be used later by a DIP.

When this instruction terminates, a return code is placed in r.0. The following lists show the return values for touch-tone and speech input, where N represents the number of touch tones received.The return values for touch-tone input and speech input are shown in Table A-1 and Table A-2 of Appendix A.

Example

getdig (WL_11, ch.F__CI_VALUE, im.64, int.F__CI_MODE)

In this example, the script accepts words from, for example, the “SAVINGS” wordlist in a string no longer than the length defined in im.64. It stores the received input in **ch.F__CI_VALUE**. It stores the indication of speech or touch-tone input in **int.F__CI_MODE**.

sp_alloc

Synopsis

This script instruction explicitly allocates/deallocates speech recognition resources.

Command Format

sp_alloc(type.onoff, type.resource)

Description

NOTE:
For CONVERSANT Version 4.0, only the WholeWord and FlexWord SP resources should be used with the sp_alloc() instruction.

NOTE:
The script instruction **sp_alloc** has replaced the script instruction **sr_alloc**. Since sr_alloc is slowly being phased out, the following warning message will appear on the screen when the sr_alloc script instruction is used:

```
WARNING: sr_alloc is obsolescent. Use sp_alloc.
```

Although the **sr_alloc** instruction will compile, application designers are encouraged to use the **sp_alloc** instruction.

The script instruction, **sp_alloc**, is explicitly used to allocate/deallocate speech recognition on the SP card. The format for the instruction is as follows:

sp_alloc(*type.onoff, type.resource*)

The **sp_alloc()** instruction may be used by a script to allocate the speech recognition resource on the SP card. Normally, this resource is shared by all scripts running on the system, and allocation is done automatically only when the script actually uses the resource. If the SP resource is not available when an instruction that requires it is executed, the instruction will fail. By using **sp_alloc()**, the script may test for the availability of a particular SP resource. If the resource is available, it will be allocated to the script until the script terminates or until the script deallocates the SP resource using **sp_alloc()**.

sp_alloc() may be used to allocate an SP resource for a period longer than the script is actually recognizing speech. Care should be taken to avoid overloading the systems SP facilities, since this can occur if many scripts using **sp_alloc()** are running simultaneously.Script register 0 (r.0) is set to the following values to indicate the status of the **sp_alloc()** execution:

0	Success
-1	Error (sp_alloc already on or off)
-2	System resources not available

The *type.onoff* argument is used to tell sp_alloc() whether to allocate or deallocate resources. Its two valid values are as follows:

1	Allocate the SP resource
0	Deallocate the SP resource

The *type.resource* argument is used to tell the sp_alloc() which SP resource or combination of SP resources to allocate or deallocate. Each SP resource has a unique value. The values for each resource and examples of how resources can be added are listed in Table A-3 of Appendix A.

The sr_talkoff Script Instruction for Speech Recognition

ATTENTION: The script instruction **sr_talkoff**, which is used to enable/disable speech recognition during the prompt *can not* be used with FlexWord.

FlexWord Speech Recognition

A

Return Values for Touch-Tone and Speech Input

Return values for touch-tone and speech input appear in the following tables:

Table A-1. Return Values for Touch-Tone Input

N > 0	If the *number* argument is greater than N (fewer than the expected number of touch tones were received), an interdigit time-out has occurred.
N = 0	An initial time-out has occurred.
N < 0	A system error occurred during the playing of the prompt or the getdig instruction itself.

Table A-2. Return Value for Speech Input

N > 0	Speech was heard and recognized. In this case, *N* represents the length of the string containing the word recognized.
N = 0	No valid speech was heard by the system. (An initial time-out has occurred.)
N < 0	A system error occurred during the playing of the prompt or the getdig instruction itself.

Resource Values

Values for various resources appear in the following table:

Table A-3. Resource Values:

1	Voice Coding or Playing
2	PRI Function
4	WholeWord Recognition
8	Call Classification
16	Text-to-Speech
64	Echo Cancelling
256	FlexWord Recognition

Adding SP Resource Values Together

The values for the WholeWord and FlexWord SP resources can be added together to allocate or deallocate more than 1 SP resource by using 1 **sp_alloc()** instruction. See Table A-4 for examples.

Table A-4.

Action	sp_alloc Script Instruction
Allocate WholeWord recognition resource for the script	**sp_alloc(im1,im.4)**
Deallocate FlexWord recognition resource for the script	**sp_alloc(im.0,im.256)**
Allocate both WholeWord and FlexWord recognition resource fo the script	**sp_alloc(im.1,im.260)**

Index

M

P

R

S

T

V

W