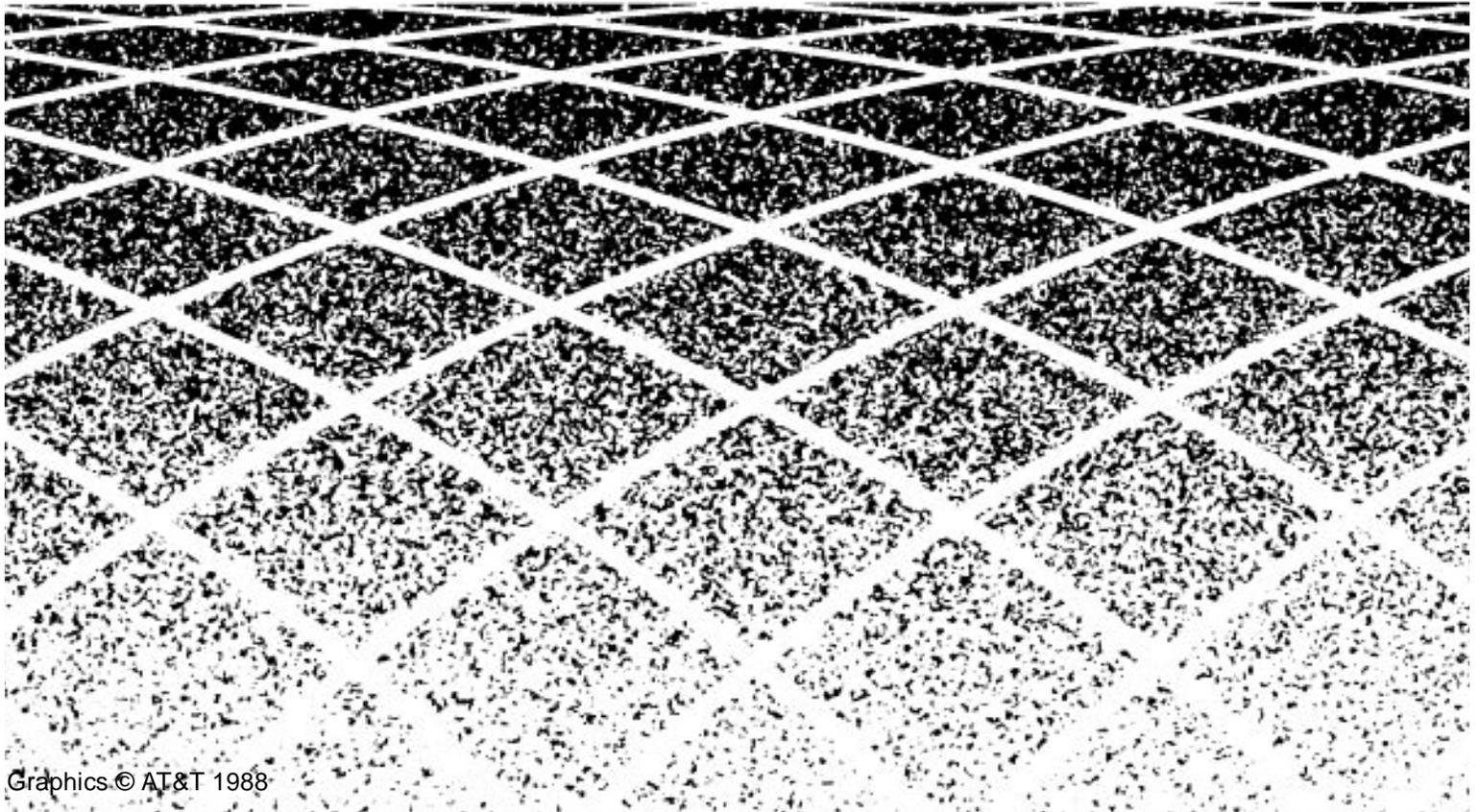




585-350-813
Issue 1
October, 1993

CONVERSANT VIS **Version 4.0** **WholeWord Bilingual Speech** **Recognition**



Contents

About This Document	vii
■ Purpose	vii
■ Intended Audience	vii
■ How to Use This Document	vii
■ Conventions Used in This Document	viii
■ Trademarks and Service Marks	viii
■ Related Resources	ix
■ How to Comment on This Document	ix

1	WholeWord Bilingual Speech Recognition	1-1
	■ Introduction	1-1
	■ Overview of WholeWord Bilingual Speech Recognition	1-2
	What is Speech Recognition?	1-2
	How Do WholeWord Bilingual and FlexWord Speech Recognition Packages Work?	1-3
	WholeWord Bilingual Recognition Types	1-5
	Supported Recognition Types	1-8
	Vocabularies of Supported Languages	1-9
	■ Hardware Requirements and Estimating Speech Recognition Resources	1-9
	■ WholeWord Bilingual Speech Recognition Features	1-11
	Barge-In Feature and Using “Recognize During Prompt”	1-11
	Word-Spotting Feature	1-14
	Number Entry	1-14
	■ WholeWord Bilingual Speech Recognition Performance	1-16
	Recognition Accuracy Rate Factors	1-16
	How to Improve Recognition Accuracy Rates	1-18
	Misrecognition vs. Recognition	1-23

Contents

2	Installing WholeWord Bilingual Speech Recognition Hardware and Software	2-1
■	Introduction	2-1
■	Installing Speech Recognition Hardware	2-2
	Companion Card Switch Settings	2-2
■	Installing Speech Recognition Software	2-4
	Stopping the VIS	2-4
	Loading Software for One Speech Recognition Language	2-4
	Loading Software for a Second Speech Recognition Language	2-5
	Removal of Any Speech Recognition Software	2-5
■	Assigning Speech Recognition Functionality to an SP Card	2-6

3	Using WholeWord Bilingual Speech Recognition with Script Builder	3-1
■	Introduction	3-1
■	Specifying Speech Recognition	3-2
	Specifying Input Mode	3-2
	Specifying US English CONFIRM Prompt	3-4
	Return Values	3-6
■	SR_Prompt External Action	3-7
■	SP_Allocate External Action	3-10

4	Using WholeWord Bilingual Speech Recognition in Scripts	4-1
■	Introduction	4-1
■	Script Instruction for Speech Recognition	4-2

Contents

The getdig Instruction for Speech Recognition	4-2
The sp_alloc Script Instruction for Speech Recognition	4-4
■ The sr_talkoff Script Instruction for Speech Recognition	4-5

5	Summary of WholeWord Bilingual Speech Recognition Commands	5-1
■	Summary of Script Instructions	5-1
■	getdig	5-2
	Synopsis	5-2
	Command Format	5-2
	Description	5-2
■	sp_alloc	5-4
	Synopsis	5-4
	Command Format	5-4
	Description	5-4
■	sr_talkoff	5-6
	Synopsis	5-6
	Command Format	5-6
	Description	5-6
	Example	5-7

A	WholeWord Bilingual Speech Recognition	A-1
■	Supported Recognition Types	A-1
■	Vocabularies of Supported Languages	A-3

B	WholeWord Bilingual Speech Recognition	B-1
■	\$CI_VALUES	B-1
■	'type' Field Values	B-3
■	Null Terminated Characters	B-5

Contents

■ Touch-Tone and Speech Input Values	B-6
■ Resource Values	B-7
Adding SP Resource Values Together	B-7

IN	Index	IN-1
-----------	--------------	------

About This Document

Purpose

This document, *CONVERSANT VIS Version 4.0 WholeWord Bilingual Speech Recognition*, 585-350-813, 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 WholeWord 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, "WholeWord Bilingual Speech Recognition", contains a definition of terms used in this book and an overview of the WholeWord speech recognition feature and its different bilingual capabilities.

- Chapter 2, "Installing WholeWord Bilingual Speech Recognition 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 WholeWord Bilingual Speech Recognition with Script Builder" explains ways in which Script Builder can be used to write speech recognition applications.
- Chapter 4, "Using WholeWord Bilingual Speech Recognition in Scripts" contains changes to existing script instructions as well as new script instructions for speech recognition.
- Chapter 5, "Summary of WholeWord Bilingual Speech Recognition Commands" summarizes the script instructions which are specifically related to speech recognition in Version 4.0.
- Appendix A contains tables.
- Appendix B 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.

WholeWord Bilingual Speech Recognition

1

Introduction

WholeWord 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 WholeWord Bilingual Speech Recognition

The “Overview” section explains how WholeWord 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. WholeWord speech recognition requires an installed and operational WholeWord 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 whole-word recognition or sub-word recognition:

- 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.
- 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.

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. 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. 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 WholeWord Bilingual and FlexWord 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 WholeWord and FlexWord, 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 was developed to permit bilingual speech recognition and can receive selected spoken input and speak preprogrammed phrases and words in one or two of the following supported languages::

United Kingdom English

United State English

Canadian French

Mexican Spanish

The UK English recognition feature supports spoken input from England, Scotland, Wales, and Northern Ireland. The US English speech recognition feature supports spoken English from any region of the United States. Canadian French supports French spoken in the province of Quebec. Mexican Spanish supports spoken input collected throughout the country of Mexico.

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.

- **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”.

WholeWord Bilingual Recognition Types

A “recognition type” is a word (for example, “no,” “ten,” one,” “yes”), which is attached to a corresponding packet of information so that it can be more easily recognized. A complete recognition type includes a mathematical “model” of the word, a “grammar” or a set of rules that specifies allowable words and word combinations at any one point in the script, and some recognition types include data interface processes. All of the recognition types, or words used by an application, constitute that application’s “vocabulary”. The standard speech recognition package provides several commonly used recognition types. However, if, for example, your banking application requires customers to input their account numbers, and all of these numbers are 7 digits long, begin with 8, and end with 0 or 1, AT&T can provide specific recognition types as part of a custom development agreement.

Models

Each word of the system’s vocabulary is represented by a mathematical “model” which contains the speech signal characteristics of the word. The model is created from thousands of samples of the spoken word. Vocabularies have one or more models for each word.

Grammars

A “grammar” is a set of rules that specifies allowable vocabulary words and vocabulary word combinations at any one point in the script, for example, {four, five, six, no}. After collecting spoken input, the speech recognition algorithm uses models and grammars to generate a list of candidates that most closely resemble this spoken input. The algorithm returns the most likely match to the script or, if no match fits, rejects the input. When input is rejected, the algorithm returns a “?” to the script. This return message is the same for all of the languages supported. All grammars provided in your speech recognition package share the same set of models for words that are common to their package’s vocabulary. If you need additional grammars for your applications. For example, if you are designing an application in which all of the identification numbers in the application will start with “068,” contact AT&T.

Grammars are constructed for fixed-length strings which are one digit or one word in length (“yes” or “no”). An example of a grammar might be as follows:

- {one, two, three, no}

- {un, deux, trois, non}

- {uno, dos, tres, no}

⇒ **NOTE:**

For packages which support connected-digit recognition: A grammar can be general, for example, any seven-digit number. Or grammars can be customized, for example, a seven-digit number beginning with the sequence 1, 2.

⇒ **NOTE:**

US English grammars allow for constant or fixed variable-length strings.

Data Interface Processes

You may use a custom data interface process (DIP) to process information not available to the SP. Some credit card numbers, for example, have check digit numbers built into them. Since DIPs go to databases to look things up or to perform calculations, DIPs are the most efficient way to do check digit verification. The DIP will select the best digit strings and scores from the set of possible matches.

However, a DIP may not be necessary for many recognition types. If there is no further information to consider for a string other than what is in the grammar, a DIP is not needed.

Specifically, none of the vocabulary word types, such as “yes” and “no,” need a DIP.

For some recognition types, there may be trade-offs that determine how much of the structure of the input is in the grammar and how much is left for the DIP. It also is possible for several different recognition types to use the same grammar but to have different DIPs.



NOTE:

For packages which support connected-digit recognition: A general grammar for strings of various lengths may need a simple DIP to select a string of the expected length.

Supported Recognition Types

The recognition type is selected in the “Mode” field on Page 2 of the Prompt and Collect screen. The recognition types for supported language packages are listed in Table A-1 of Appendix A.

The WholeWord recognition types are for single words or digits only. That is, callers are expected to speak only 1 word or 1 digit. The US_1_3 recognition type, for example, means that callers are expected to say a single word, either “one,” “two,” or “three.” A prompt that might use the 1-3 digit grammar type could be structured, respectively, as follows:

- “For checking account balance, say ‘one.’ For savings account balance, say ‘two.’ For interest rates, say ‘three.’”

- —Pour connaître la position de votre compte courant, dites “un”. Pour la position de votre compte d’épargne dites “deux”. Pour les taux d’intérêts, dites “trois”.

- “Para obtener el saldo de su cuenta de cheques, diga ‘uno’. Para el saldo de su cuenta de ahorros, diga ‘dos’. Para las tasas de interés vigentes, diga ‘tres’”.

NOTE:

For packages which support connected-digit recognition: The connected-word recognition type specifies a string of naturally spoken digits (words). The spoken string can be fixed in length from 1-10 digits, or it can be of a variable length of 1-24 digits. Grammars are provided for the 1-10 digit fixed-length and the 1-24 variable-length strings. The string length is specified by the Min/Max values in the Prompt & Collect screen in Script Builder.

The desired string length must be specified when selecting the recognition type for a fixed-length string. An area code (614), for example, consists of 3 words; therefore, the minimum and maximum values are “3.” If a variable-length string of 1-4 words is required, “1” is the minimum value and “4” is the maximum value.

Vocabularies of Supported Languages

The vocabularies for all supported languages are listed in Table A-2 of Appendix A.

Hardware Requirements and Estimating Speech Recognition Resources

The VIS supports a maximum of 6 channels of simultaneous speech recognition of connected digits, a string of digits spoken one after the other, per companion (CMP) card. It also supports shared access to hardware resources for applications that do not require continuous speech recognition.

Speech recognition is performed by the signal processor card (SP). For 12 channels of speech recognition, 1 dedicated SP and 2 companion cards (CMP) are needed. One SP with 1 CMP supports up to 6 channels of speech recognition.

$$1 \text{ SP} + 1 \text{ CMP} = 6 \text{ Channels}$$

$$1 \text{ SP} + 2 \text{ CMP} = 12 \text{ Channels}$$

To understand the number of “supported channels” (the number of simultaneous calls that can be handled which require some speech recognition), you must be familiar with your VIS’s number of recognition resources. If your script does not allow for barge-in (callers speaking before the prompt is finished, where SR_Prompt is set to “no” or sr_talkoff uses [im.0]), then the recognition resources can be shared. When barge-in is disabled, the recognition resource is only allocated when a prompt that allows for speech recognition is completed. The resource is used to analyze the caller’s response, and then the resource is freed. Because of this, the VIS can handle more channels than there are resources. However, the number of simultaneous calls that can be handled will depend on the amount of time that the resource is being used during each script.

The “barge-in feature” lets callers speak before the prompt is finished. For scripts that allow barge-in (where SR_Prompt is set to “yes” or sr_talkoff uses [im.0]), the recognition resources are in use during the entire length of each interruptible prompt. (Interruptible prompts can be created by setting the Recognize During Prompt on page 1 of the Prompt and Collect screen to “yes.”) Typically, scripts with the barge-in option set to **on** potentially handle fewer channels than available recognition resources allow. (For more information about barge-in, see “Barge-In Feature and Using “Recognize During Prompt”” in the following “WholeWord Bilingual Speech Recognition Features” section.)

Application developers who want to increase the VIS's ability to share recognition resources should consider the following channel capacity factors:

- The percentage of time: It's important to weigh the number of prompts needed to recognize input during the script versus the time spent doing other things, such as database reads.
- The use of barge-in: Setting the SR_Prompt to "yes" or setting the sr_talkoff (im.0) limits the ability to share resources.
- The percentage of callers who use touch tone input: If your script handles touch-tone callers with separate prompts (or an 'exec'ed script), or if your script disables barge-in when a caller uses a touch tone, you can increase the number of speech recognition channels supported.

Setting the "talk" option for the card to off load non-essential speech processing from the SP card whenever possible permits application designers to make better use of speech recognition resources in a IVP4/6 card system. For IVP4 and IVP6 cards, if the "talk" option is set and **sr_talkoff** or SR_Prompt is turned on, prompts and announcements will be played as if the card were configured with the "tdm" option. If talkoff or SR_Prompt is not on, the prompt will be played with the "talk" option. (When the "talk" option is set, the IVP4/6 card plays prompts and announcements; when the "tdm" option is set, the SP card plays the prompts and announcements.)

Application designers must use good judgement when deciding when to turn on and turn off **sr_talkoff** and SR_Prompt. If speech processor resources are being strained, turning off **sr_talkoff** or SR_Prompt and moving speech playback to the IVP4/6 cards will relieve some of the load on the SP card. This means, however, that callers will not be able to talk during the prompt and have their responses recognized.

Applications should turn off sr_talkoff and SR_Prompt as soon as speech recognition services are no longer required. If only one **getdig** instruction is required, sr_talkoff and SR_Prompt should be turned off after the instruction has been completed. If a caller opts to enter a touch tone, the application may assume that the caller prefers to enter touch tones, so speech recognition is not required and sr_talkoff and SR_Prompt can be turned off.

WholeWord Bilingual Speech Recognition Features

Features included in this package include barge-in, word-spotting, and number entry capabilities. These features are discussed in detail in the following subsections.

Barge-In Feature and Using “Recognize During Prompt”

With speech recognition, callers have the option of speaking a vocabulary word or entering touch-tones in response to a prompt. (See Chapter 4, "Using WholeWord Bilingual Speech Recognition in Scripts".) “Barge-in,” also referred to as “Recognize During Prompt,” allows callers to interrupt or “barge in” during voice playback by speaking a vocabulary word. As soon as the system recognizes a vocabulary word, the prompt stops playing. Barge-in operates much like the “talkoff” feature for touch-tone input. Experienced callers appreciate being able to shorten the transaction time by not listening completely to each prompt. However, barge-in can monopolize speech recognition resources. If resources are over extended, you may want to turn off the barge-in feature.

For isolated word recognition, the prompt will not stop until the SP recognizes a valid vocabulary word. Once the prompt completes playback, the initial time-out eventually ends the recognition if no valid input is received.

⇒ NOTE:

For packages which support connected-digit recognition: The prompt playback stops between the recognition of the first word and last words of the input, at a point determined by the recognizer’s ability to decide when valid input has started.

At the script level, the Recognize During Prompt is enabled using the external action SR_Prompt or the script instruction **sr_talkoff**. This prompt or instruction prepares the system for speech recognition during the prompt. Then, any prompt with Recognize During Prompt set to “yes” on page 1 of the Prompt and Collect screen will be interruptible by voice or by touch tones. (See Chapter 3, "Using WholeWord Bilingual Speech Recognition with Script Builder"; and Chapter 4, "Using WholeWord Bilingual Speech Recognition in Scripts".) If your script does not use Recognize During Prompt, be sure your prompts are worded carefully in order to make sure that callers know they can’t speak until the prompt is finished. For instance, look at the wording and the <pause> length of the following prompts. There are no large gaps of silence to tempt callers to speak during the menu prompts. These prompts are designed for use when barge-in is **off** so that callers can’t barge-in and must wait until the prompt is finished:

- “For sales say, ‘one,’ now. <a short pause>
For service, say, ‘two,’ now. <a short pause>
For a representative, say ‘three,’ now.”

- —Pour les ventes, dites “un”, maintenant. <une pause courte>
Pour des questions techniques, dites “deux”, maintenant. <une pause courte>
Pour parler avec un représentant, dites “trois”, maintenant.

- “Para comunicarse con ventas, diga ‘uno’, ahora. <una pausa corta>
Para servicios diga ‘dos’, ahora. <una pausa corta>
Para hablar con un representante, diga ‘tres’, ahora”.

- “You may order up to five copies. Please say how many copies you would like, now.”

- —Vous pouvez commander jusqu’à cinq copies. Dites combien de copies vous désirez, maintenant.

- “Usted puede ordenar hasta cinco copias. Ahora diga por favor cuántas copias quiere”.

These next prompts are examples designed to encourage callers to barge-in when barge-in is set to **on**. Note that the pauses are longer, so callers are encouraged to respond after hearing the desired option.

- “For sales say, ‘one.’ <a 1-1.5 second pause>
For service, say, ‘two.’ <a 1-1.5 second pause>
For a representative, say ‘three.’ <a 1-1.5 second pause>

- —Pour les ventes, dites “un”. <pause de 1 à 1,5 secondes> Pour des questions techniques, dites “deux”. <pause de 1 à 1,5 secondes>
Pour parler avec un représentant, dites “trois”.

- “Para comunicarse con ventas, diga ‘uno’. <una pausa de 1.5 segundos>
Para servicios diga ‘dos’. <una pausa de 1.5 segundos>
Para hablar con un representante, diga ‘tres’.

- “You may order up to five copies. How many copies would you like?”
- —Vous pouvez commander jusqu’à cinq copies. Dites combien de copies vous désirez.
- “Usted puede ordenar hasta cinco copias. Diga por favor cuantas copias quiere”.

The first prompt, although asking for the same information as the previous example, clearly leaves the caller enough time to barge-in before the prompt is finished. The second prompt does not include the time reference “now”; therefore, callers can speak at any time. Please note that if there is silence at the end of a prompt, the caller may speak but the system may not be prepared to listen. Make sure that there is no silence at the end of your recorded phrases when barge-in is set to **off**.

In order to use the Recognize During Prompt feature, IVP6 or VRS6 cards should be set to the “tdm” option, and the SP card must be assigned with “WW_RECOG+VOICE” functionality. For information about assigning the “tdm” function to the T/R cards and speech recognition functionality to the SP card, see “Changing Voice Equipment Options” in Chapter 3, “Configuration Management,” of *CONVERSANT VIS Version 4.0 Operations*, 585-350-703, and “Assigning Speech Recognition Functionality to an SP Card” in Chapter 2, “Installing Whole-Word Bilingual Speech Recognition Hardware and Software”, later in this book.

Word-Spotting Feature

Speech recognition supports “word spotting.” For example, a caller says:

- “I want number five, please.”
- “Je voudrais le numéro cinq, s’il vous plait”.
- “Quiero la número cinco, por favor”.

The VIS recognizes the word “five” or “cinq” or “cinco” as a valid response and ignores the rest of the input.

Number Entry

Applications can accept spoken digits in two ways. “Connected digits” are strings of digits, spoken one after the other with no long pause in between. “Tone-paced digits” are strings of digits spoken one at a time. The caller speaks a digit then waits for a tone before saying the next digit. Number entry is an essential feature if you want your callers to input numbers greater than one digit in length, (for example, “1-3-5” or “2-4-6-8”). But the spoken input has to be collected singularly if you do not have connected digit recognition. (See the following note if your package supports connected digit recognition.) If your package does not support connected digits, then you must piece together caller tone-paced input in order to create a digit string. A four-digit number, for example, could be collected as follows:

- “Please say the 4-digit extension of the person you wish to call. After you say the first digit, wait for a tone before saying the next.”
- “Veuillez donner le numéro à quatre chiffres de la personne à qui vous voulez parler. Attendez pour le bip sonore entre chaque chiffre”.
- “Pour favor diga los cuatro digitos de la extensión de la persona con las que quiere hablar”.

Callers inputting the three-digit string, 6-1-4, might say the following:

- (beep) “six” (beep) “one” (beep) “four”
- (bip) “six” (bip) “une” (bip) “quatre”
- (tono) “seis” (tono) “uno” (tono) “cuatro”

If callers need to enter longer numbers, application designers must make sure that they write concise prompts which will not overload the caller with information and desired actions.

 **NOTE:**

For packages which support connected-digit recognition: Application developers can collect input by specifying a variable or fixed-length string within a determined range of up to 24 connected digits. Recognition, however, is most accurate when the feature is asked to recognize the provided 1-10 digit fixed-length strings. Recognition accuracy is poorer for variable-length strings than it is for fixed-length strings.

WholeWord Bilingual Speech Recognition Performance

The following sections contain information concerning the performance of the Speech Recognition feature on the VIS. Included are helpful hints and tips for using the feature and enhancing accuracy rates.

Recognition Accuracy Rate Factors

Accuracy is affected by both recognizer performance and caller performance. (See “Misrecognition vs. No Recognition.”) If callers input unrecognizable information the system can not be accurate. For example:

Callers can say something that the system doesn't recognize:

- “I don't know.”
- “Je ne sais pas”.
- “No se”.

⇒ NOTE:

For packages which support connected-digit recognition: Callers can interject nonvocabulary words into the string:

- “three-four-uh-four-two-um-five”
- “trois-quatre-euh-quatre-deux-euh-cinq”
- “tres-cuatro-eee-cuatro-dos-emmm-cinco”

Callers can say a natural number instead of the requested digits:

Incorrect	Correct
"thirty-five"	"three-five"
"trente-cinq"	"trois-cinq"
"treinta y cinco"	"tres-cinco"

For variable-length strings, such as a billing account number that is 4 to 5 digits long, the accuracy will be slightly less than that of a 5 to 6 digit string. Shorter variable-length digit strings tend to be more accurately recognized. In addition, any words the caller adds before or after saying a variable-length string will challenge the feature's word spotting capabilities and decrease the accuracy rate.

The accuracy of the speech recognition feature is highest for single or isolated words is high. For example, US English speakers located throughout the calling population who use the phone in a quiet environment can expect a per-word/digit recognition accuracy rate of 97%. Recognition accuracy for different languages will vary. Any particular individual or group of individuals within the calling population may see slightly better or worse results.

Extraneous speech, slurring of words, speaking too quickly and speaking too slowly all contribute to reduced recognition accuracy rates. The length of the string to be recognized also can greatly influence the accuracy of the recognition. Each additional word in a string causes the overall accuracy to fall slightly. The longer the string, the greater the chance the caller will say something other than the valid words, and, consequently, the greater the chance that the caller's words may not be recognized correctly. The recognition accuracy rate of US English digit strings can be calculated by using .97 to the nth power, where n is the number of digits. As a rule, you can deduct 3 percentage points from the accuracy rate for each digit in the string. For example, the accuracy rate of a 2-digit string is about 94%; a three-digits string, 91%; a four-digit string, 88%; a five-digit string, 85%; a six-digit string, 83%; a seven-digits string, 80%.

⇒ NOTE:

A 5-digit accuracy rate of 85% means that 85% of the 5-digit utterances made to the script are recognized correctly.

How to Improve Recognition Accuracy Rates

Given the variance in speech input and the statistical properties of recognition algorithms, the speech recognizer may make an occasional error. However, speech recognition accuracy depends not only on the recognition algorithms, but also on the models, grammars, DIPs, prompts, calling population, and the data to be recognized. Each of these underlying factors can influence recognition accuracy.

Positive Influences on Recognition Accuracy

The following items factor into positive recognition accuracy:

- **Isolated word recognition.** Accuracy for recognizing isolated words is very high.
- **Validation of data.** If possible, it is always a good idea to verify the recognized result against a database or host field. This will help improve the overall accuracy of your application.
- **Experienced calling populations.** Recognition improves for applications in which the calling populations are closed and the callers are experienced and/or trained.
- **Custom grammars and DIPs.** Custom grammars improve the signal processor's (SP) ability to "score" the candidate, since these grammars selectively limit recognition possibilities. Scoring the candidate is the SP's process of assigning a probability rating to the recognized input and sending the highest ranking result back to the script. Custom DIPs help further process the recognition result with information unavailable to the SP card. However, not all applications can take advantage of custom work.
- **Using "Recognize During Prompt."** When this option is used, the prompt stops as soon as the SP recognizes sufficient vocabulary words. The Recognize During Prompt or barge-in results in a potentially higher degree of accuracy.
- **Confirm.** If the accuracy measurement is based on an application with confirmation and reprompt steps, overall recognition accuracy will increase. It is always a good idea to check the recognized result for validity *before* proceeding with the interaction. For example:
 - "You said, 'one, two, three.' Is this correct? <pause> Please say 'yes' or 'no.'"
 - Vous avez dit "un, deux, trois". Est-ce correct? <pause> S'il vous plait répondez "oui" ou "non".

- “Usted dijo ‘uno, dos, tres.’ ¿Es esto correcto? Por favor diga ‘sí’ o ‘no’ “.

Thus, if the caller responded with nonvocabulary words, the validation check will allow the caller another chance to say acceptable words. For example, if the caller says “no” or <<non>>, your application could say:

- “Sorry, please say your account number again.”
- “Désolé, veuillez répéter votre numéro de compte”.
- “Los siento, por favor dígame su número de cuenta otra vez”.

If the caller confirms the input, she/he also affirms that the recognition was correct. verifies her/his response.

- **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 acceptable. One solution may be to provide more informative prompts for first-time callers only, otherwise design short or moderate length prompts. However, for applications with infrequent users, lengthy prompts may be more acceptable and help improve the accuracy.
- **Prompt structure.** Prompts offered in calm, clear voices greatly affect recognition accuracy as do the specific structures of the prompts. Prompts which are set up to help callers barge-in before the prompt is finished or to listen to the whole prompt before responding (as is the case when the barge-in is disabled), increase recognition accuracy.
 - 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 ‘one.’
To hear your savings account balance, say ‘two.’”
 - —Pour connaître la position de votre compte courant, dites “un”.
Pour connaître la position de votre compte d’épargne, dites “deux”.
 - “Para oír el saldo de su cuenta de cheques diga ‘uno’.
Para oír el saldo de su cuenta de ahorros diga ‘dos’ “.

By placing the action required at the end of the prompt, the caller is able to remember what exact action is required of her or him. Often prompts are designed with the action offered before the description:

“Say ‘one’ for a description of the up-coming gallery events.” This prompt structure encourages the caller to forget the specific <action required>, since the last thing related is the desired result and not the action. In addition, if you wish to encourage your callers to barge-in when they hear their desired option, you can add a small pause after the <action required> phrase.

- Yes/no prompts. A yes/no prompt should be structured as a yes or no question, that is, in the same way that a human would ask a question. For example:
 - “Would you like to hear your order again?”
 - “Voudriez vous entendre votre demande encore une fois?”
 - “¿Le gustaría oír su orden otra vez?”

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

- “Would you like to hear your order again? Please say ‘yes’ or ‘no.’”
- —Voudriez vous entendre votre demande encore une fois? Veuillez répondre “oui” ou “non”.
- “¿Le gustaría oír su orden otra vez? Por favor conteste ‘sí’ o ‘no’”.

The previous examples are worded in a way that is *more* natural than the following response:

- “If you would like to hear your order again, say ‘yes.’ Otherwise say ‘no.’”
- — Si vous voulez entendre votre demande encore une fois, dites “oui”, sinon dites “non”.
- “Si a usted le gustaría oír su orden otra vez, diga ‘sí’. En caso contrario diga ‘no’ “.

As previously stated, if you wish to encourage the use of barge-in, add a small pause (about 1-1.5 seconds) following the question.

For example:

- “Would you like to hear your order again? <pause> Please say yes or no.”
- —Voudriez vous entendre votre demande encore une fois? [pause] Veuillez répondre “oui” ou “non”.
- ¿ ”Le gustaría oír su orden otra vez ? <pausa> Por favor conteste ‘sí’ o ‘no’ “.

 **NOTE:**

For packages which support connected-digit recognition:

Fixed-length digit string. Accuracy is better for fixed-length strings (min_digits = max_digits) than for variable-length strings. If possible, avoid trying to recognize variable-length strings.

Negative Influences on Recognition Accuracy

The following items have a negative influence on recognition accuracy.

- **Information type.** Asking for information which is not usually related in digits, such as dates and dollar amounts, may adversely affect recognition accuracy. For example, to input the date December 15th, callers may be required to say the digits, "1-2-1-5". To input the dollar amount \$25.00, callers may be required to say, "2-5-0-0" or (beep) "two" (beep) "five" (beep) "zero" (beep) "zero." In addition, if callers wish to input natural numbers, the numbers need to be stated in digits. The feature will recognize "three-two," but it won't recognize "thirty-two." Training callers to input data in recognizable digits may help to increase recognition accuracy. (For information on techniques callers should use when inputting bilingual data, see "Number Entry" in the "WholeWord Bilingual Speech Recognition Features" section of this chapter.)
- **Callers respond without key words.** If callers respond to a prompt without using any key words (indicated by the Recognition Type selected in the Prompt and Collect statement), the VIS will try to find a result that matches the recognition type supplied. For example, let's say the Recognition Type "US_DIG" is set with both the *min_digits* and *max_digits* equal to 3. The caller is prompted with "Please say your 3-digit location code." If the caller responds, "I don't know what number you're looking for," the VIS will compare the caller's response to the 3-digit string prompt with 3 digits in the range of 0-9 (or will piece together the tone paced input from the words found in the caller's response). Application designs must be robust enough to handle to callers who might respond to prompts in such a manner. If there is a possibility that the caller does not have the requested information, for instance, an account number on the caller's bill, your script could screen this information by first asking, "Do you have your bill available?" The caller is more likely to answer the yes/no question correctly, and, with this yes/no prompt, the script can respond more intelligently to the caller. Remember that word spotting can't recognize keywords which have been omitted by the caller.
- **Environment.** Noisy environment, such as an airport or a train yard, may contribute to recognition accuracy problems. In certain cases, custom speech data can be collected and word models can be built in a way which will compensate for the noise level of the environment, thereby increasing recognition accuracy.

⇒ NOTE:

For packages which support connected-digit recognition:

- **Connected-digit string length.** Recognition works best for shorter digit strings, those comprised of less than 5 digits. Each digit added causes the overall accuracy rate to fall slightly. Custom grammars or DIPs should improve accuracy for longer digit strings. (For more information on calculating accuracy rates for digit strings, see "Recognition Accuracy Rate Factors" in the "WholeWord Bilingual Speech Recognition Performance" section of this chapter.
- **Word spotting with variable-length recognition type.** For fixed-length recognition types (`min_digits = max_digits`), the word spotting feature recognizes the desired vocabulary word in the caller's response and filters out all other nonvocabulary words. However, word spotting *does not* accurately filter out nonvocabulary words in variable-length recognition types (`min_digits < max_digits`). Accuracy substantially decreases if callers speak extraneous, non-vocabulary words while the system is trying to recognize a variable-length string.

Misrecognition vs. Recognition

"Misrecognition" occurs when the VIS incorrectly recognizes spoken input as something other than what the caller says. The chance of misrecognition is greater with strings of digits than it is with single words. The longer the string, the greater the chance of misrecognition.

"No recognition" occurs when the system does not recognize the spoken input. It may detect input, but it is unable to make a reasonable guess as to the exact spoken word(s).

⇒ NOTE:

For packages which support connected-digit recognition: If there is a check digit in the string, misrecognition can be identified in the DIP and a corrective action can be taken. For example, if this check digit is not correct, the DIP may tell the VIS to reprompt the caller for the string. In the case of no recognition, the VIS will follow the instructions for the "Not on List" category of the Prompt and Collect checklist. The default setting for "Not on List" reprompts the caller; however, the application developer can choose what action the VIS should take under this circumstance. When no recognition occurs, the value of `$CI_VALUE` is set to "?".

Installing WholeWord Bilingual Speech Recognition Hardware and Software

2

Introduction

This chapter contains information on installing the WholeWord Bilingual Speech Recognition 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 Speech Recognition Hardware

For information on installing the companion card (CMP), refer to Chapter 7, "Installing CONVERSANT Circuit Cards," of *Voice Processing Hardware Installation and Upgrade* 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.

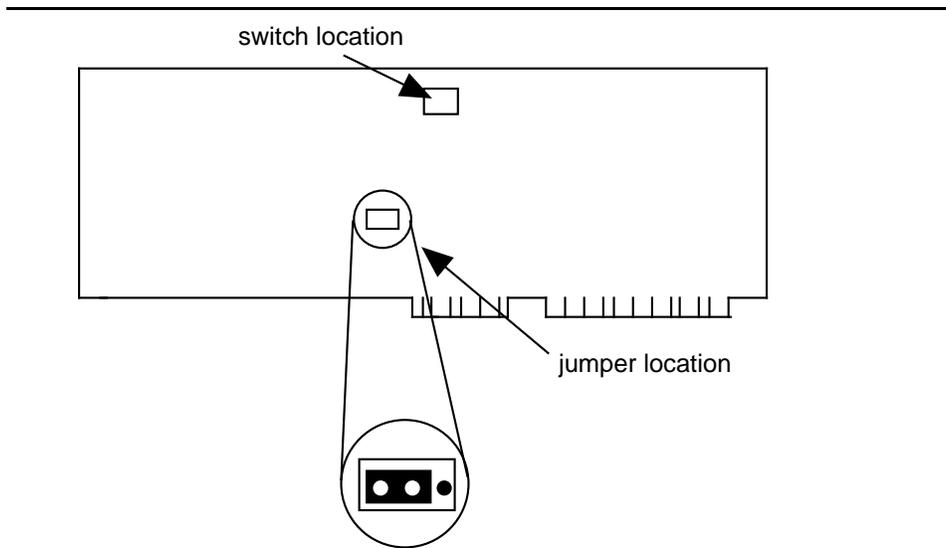


Figure 2-1. Companion Board Jumper and Switch Locations

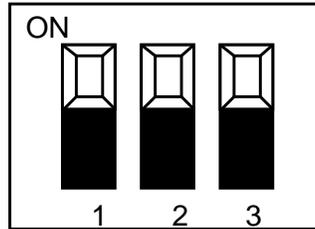


Figure 2-2. Companion Board 0 Switch Setting

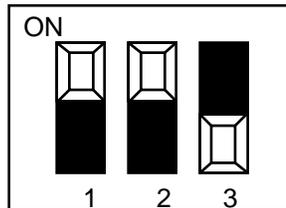


Figure 2-3. Companion Board 1 Switch Settings

Installing Speech Recognition Software

If you are installing the speech recognition 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*, 585-350-111 and *Software Upgrade*, 585-350-110.

If you are installing the speech recognition 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 One Speech Recognition Language

1. Insert the disk of the speech recognition feature package set into the floppy disk drive.
2. Enter **installpkg**
3. The instructions displayed on the screen will tell you when to insert the subsequent disks.
4. When the installation is complete, you may either restart the VIS, or continue with other tasks.

Loading Software for a Second Speech Recognition Language

1. Insert the disk of the second speech recognition package set into the floppy disk drive.
2. Enter **installpkg**
3. The instructions displayed on the screen will tell you when to insert the subsequent disks.
4. When the installation is complete, you may either restart the VIS, or continue with other tasks.

Removal of Any Speech Recognition Software

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

1. Stop the VIS using the procedure 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 language of the speech recognition software package that you wish to remove.

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



NOTE:

Again, make sure you remove only the language you want to remove, since there might be two languages on the system.

Assigning Speech Recognition Functionality to an SP Card

In order for the speech recognition feature to operate properly, you must assign speech recognition functionality to the SP card(s).

⇒ NOTE:

If you have previously completed this step while installing your first supported language, then you do not need to repeat this step when installing the second language.

⇒ 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.
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).

Assign Functions to SP Cards	
Function :	<u>WW_RECOG</u>
Card No(s) :	<u>1</u>

Select the desired function from CHOICES key.

HELP	CHOICES	SAVE	PREV-FRM	NEXT-FRM	CANCEL	CMD-MENU	CHG-KEYS
------	---------	------	----------	----------	--------	----------	----------

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.

- In the Function field, enter **WW_RECOG** or **WW_RECOG+VOICE** to assign speech recognition functionality to an SP. WW_RECOG allocates that SP for recognition only. WW_RECOG+VOICE allocates the SP for recognition and speech playback.

For an enabled barge-in feature to work correctly (SR_Prompt set to “yes” or sr_talkoff[im.0]), the SP must have WW_RECOG+VOICE functionality.

- In the Card No(s) field, specify the card that you wish to assign WW_RECOG or WW_RECOG+VOICE functionality.

Figure 2-4 has WW_RECOG assigned to card “1.”

- Press **(SAVE)** to save the entered values.
- Press **(CANCEL)** once to return to the Voice Equipment screen.
- 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 INSERT.
- Press **(CHGSTATE)** to display the Change State of Voice Equipment screen (See Figure 2-5).

Change State of Voice Equipment	
New State:	inserv _____
Equipment:	card _____
Equipment Number:	1 _____
Change Immediately:	yes _____

Enter inserv or manooos (manual out of service).

HELP	CHOICES	SAVE	PREV-FRM	NEXT-FRM	CANCEL	CMD-MENU	CHG-KEYS
						FRM-MGMT	CHG-KEYS

Figure 2-5. Change State Voice Equipment

- In the New State field, enter **inserv**
- In the Equipment field, enter **card**
- In the Equipment Number field, enter the number of the SP card.

15. In the Change Immediately field, enter **yes**
16. 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.
17. 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 speech recognition 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.

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

Diagnose Equipment	
Equipment to Diagnose:	<u>card</u>
Equipment Number:	<u>1</u>
Immediate Diagnose?	<u>yes</u>

Enter card.

HELP	CHOICES	SAVE	PREV-FRM	NEXT-FRM	CANCEL	CMD-MENU	CHG-KEYS
						FRM-MGMT	CHG-KEYS

Figure 2-6. Diagnose Equipment Screen

21. In the Equipment to Diagnose field, enter **c** to specify card.
22. In the Equipment Number field, enter the SP card number obtained from the Voice Equipment window. If speech recognition 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.

23. In the Diagnose Immediately field, enter **yes**
24. Press **SAVE** to save the values you entered and perform the diagnose.

Using WholeWord Bilingual Speech Recognition with Script Builder

3

Introduction

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

- Procedures for specifying speech recognition in a transaction
- A description of the external actions **SR_Prompt** and **SP_Allocate** used for speech recognition.



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 Speech Recognition

Specifying Input Mode

When the speech recognition 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 speech recognition grammar. Figure 3-1 shows this form with the valid speech recognition choices. To display these choices from the form, press **CHOICES**. This is a dynamic menu, which will display all recognition types available on your system.

The screenshot shows the AT&T CONVERSANT Script Builder interface. At the top, it displays 'AT&T CONVERSANT Script Builder Version 4.0 River_Bank'. The main window is titled 'Define Prompt and Collect Page 2 of 3'. A dropdown menu for 'RECOGN TYPE' is open, showing options: US_YN, US_1_3, US_1_3N, US_1_5, US_1_5N, and US_DIG. The form contains several fields for defining caller input parameters, including 'Caller Input Field', 'Tries Used Field', 'Digits Input Field', 'Mode', 'Min. No. Of Digits', 'Max. No. Of Digits', 'TT Terminator Code Active', 'TT Terminator Code Value', 'TT Repeat Code Active', 'TT Repeat Code Value', 'TT Erase Code Active', 'TT Erase Code Value', 'TT Cancel Code Active', 'TT Cancel Code Value', 'No. Of Tries To Get Input', 'Initial Timeout', and 'Interdigit Timeout'. Below the form, there is a section for '#Greet caller' with the text '11. Announce'. At the bottom, a message reads 'Define caller input parameters. Press CLOSE when the definition is complete.' and a row of buttons includes 'HELP', 'REDRAW', 'LIST', 'CANCEL', and 'EXIT'.

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

Script Builder validates the minimum and maximum number of digit fields for the speech recognition type selected and replaces invalid recognition types with values closest to the ones specified. The minimum and maximum number of digit fields in the form limit the amount of caller input (spoken or touch-tone) that can be recognized by the system. These numbers are set according to recognition type. The yes/no recognition type, for example, only accepts one character. Therefore, the minimum and maximum values for packages without connected digits are set to 1. The minimum and maximum number displayed tells you: "Minimum number of digits set to *x*; maximum number of digits set to *y*."

⇒ NOTE:

For packages which support connected-digit recognition: By setting the minimum and maximum number of digit fields to values greater than 1, you can collect numbers that are comprised of more than 1 digit. For example, to collect the digit 5, the min./max. would be set to 1. To collect the number 614, the min./max. would be set to 3, since there are three digits in this target number.

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 US English CONFIRM Prompt

Using page 3 of the Define Prompt and Collect form, you can direct US English speech recognition applications to prompt callers in order to have them confirm that the VIS recognized what was spoken. Figure 3-2 shows the Define Prompt and Collect form, page 3, with the action CHOICES menu displayed.

AT&T CONVERSANT Script Builder Version 4.0 River_Bank

Define Transaction

Define Prompt and Collect Page 3 of 3

CHECKLIST Do you want to use the Action Choices yes

Case	Voice Response	Action	Action Data
<u>Input OK</u>	_____	<u>Continue</u>	_____
<u>Initial Timeout</u>	_____	<u>Reprompt</u>	_____
<u>Too Few Digits</u>	_____	<u>Reprompt</u>	_____
<u>No More Tries</u>	_____	<u>Quit</u>	_____

9. Quit
HOST_UP_HOURS_IN:
10. Answer Phone
#Greet caller
11. Announce

Define caller input check list. Press EXPAND to add voice response.

EXPAND PREVPAGE NEXTPAGE CHG-KEYS

Figure 3-2. Define Prompt and Collect Page 3—Choices for Action Field

To display these choices, move the cursor to a line in the Action field, then press **(CHOICES)**.

The confirm action makes a single touch-tone digit an optional argument, and then enters the digit into the Action Data field. In addition, the confirm action has an associated voice response window that is spoken before the confirmation. This window is used to play back caller input and to ask the caller to confirm her/his

input by saying “yes” or by entering a touch-tone digit. To open the voice response field, position the cursor on that field, then press **(EXPAND)**.

If a spoken “yes” or a touch-tone digit that matches the optional argument is received, the script continues to the next action step in the defined transaction. If a spoken “no” or a touch-tone digit that does not match the optional argument is received, the script reprompts the caller for the input. If an optional argument is not given, then when the caller enters a touch-tone the script treats the confirm as a “continue”. Thus, the script only requests confirmation with callers’ spoken input, not callers’ touch-tone input.

The CONFIRM action, which has an optional touch-tone confirmation digit associated with it, can be used in one of two ways:

- If you want to require the caller to confirm the input, whether it was entered with touch-tones or spoken, then a touch-tone digit should be specified for the confirmation digit. The input will be accepted if the user says “yes” or presses the specified digit on the touch-tone pad.
- If you want to require the caller to confirm the input only if it was spoken (since touch-tone input is highly reliable, barring user error), leave the confirmation digit blank. In this case, the prompt and the CONFIRM are only done if the caller speaks the input. The CONFIRM requires a spoken “yes” before the input is accepted.

Return Values

\$CI_MODE

The \$CI_MODE variable is defined for applications that use speech recognition. 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 digit or word recognized by the system. The recognized digit/word and the value in \$CI_VALUE are shown in Table B-1 of Appendix B. The value “?” can be returned as valid input in \$CI_VALUE within the Prompt and Collect Standard Checklist. This return value indicates that speech energy is present but cannot be interpreted. The recommended action is to check the validity of the \$CI_VALUE by using an Evaluate action step following the Prompt and Collect action step. If a “?” occurs as the return value, you can specify the appropriate action you wish the script to take, such as reprompting the caller for input.

⇒ NOTE:

In the Prompt and Collect Custom Checklist, the return code “?” is classified in the “Not on List” category. You can directly specify the action to take in this instance in the Custom Checklist.

SR_Prompt External Action

The SR_Prompt external action allows a caller to interrupt voice playback with speech input. This feature, also known as “Recognize During Prompt,” operates much like the talkoff feature for touch-tone input. When SR_Prompt is enabled, those prompts specifying “yes” for “Interrupt During Prompt?” for all Prompt and Collect actions are stopped as soon as the VIS recognizes enough valid speech input. Prompts that specify “no” for “Interrupt During Prompt?” are not affected by this action.

If you want to use barge-in, put the SR_Prompt action in your script right after “Answer Phone.” In addition, when using the Call Bridge action, the script should disable the Recognize During Prompt feature *before* the bridge and enable it once the bridge is established. This allows the speech recognizer to acclimate itself to the specific characteristics of the new connection.

⇒ NOTE:

SR_Prompt should only be turned off and on throughout the application when using a Call Bridge action. If SR_Prompt is repeatedly turned on and off, recognition accuracy is influenced.

Follow the procedure below to select the SR_Prompt external action:

1. In the Define Transaction window, with the cursor on the line above where you want to add SR_Prompt, press **(ADD)**.

System response:

The Action Choices menu is displayed.

2. Highlight SR_Prompt, then press **(ENTER)**.

System response:

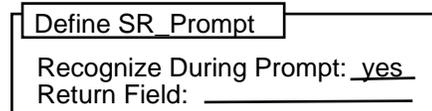
The external action is inserted in the transaction below the cursor.

3. Press **(CANCEL)** to close the Action Choices menu.

4. Highlight External Action: SR_Prompt, then press **DEFINE**

System response:

The following form is displayed. (See Figure 3-3.)



Define SR_Prompt

Recognize During Prompt: _yes_

Return Field: _____

Figure 3-3. Define SR_Prompt Form

5. To enable Recognize During Prompt, enter **yes** in the field, then press **ENTER**. The default value in the field when this form opens is “yes.”
To disable Recognize During Prompt, enter **no** in the field, then press **ENTER**.
6. Specify an optional return field in the Return Field: field then press **CLOSE**.
If the field name entered has not been defined previously, the Define Transaction Fields screen opens when **CLOSE** is pressed. Define in the Field Type of the return field as “num,” then press to exit the Define Transaction Fields screen.
The return field is set to one of the following values depending on the status of the SR_PROMPT: :

0	Success
-1	Failure
-2	System resources not available

7. Press **(CLOSE)** to exit the Define SR_Prompt form and return to the Define Transaction window.
 8. In the Define Transaction window, press **(SHOW)** to expand the External Action: SR_Prompt action step. (See Figure3-4.)
-

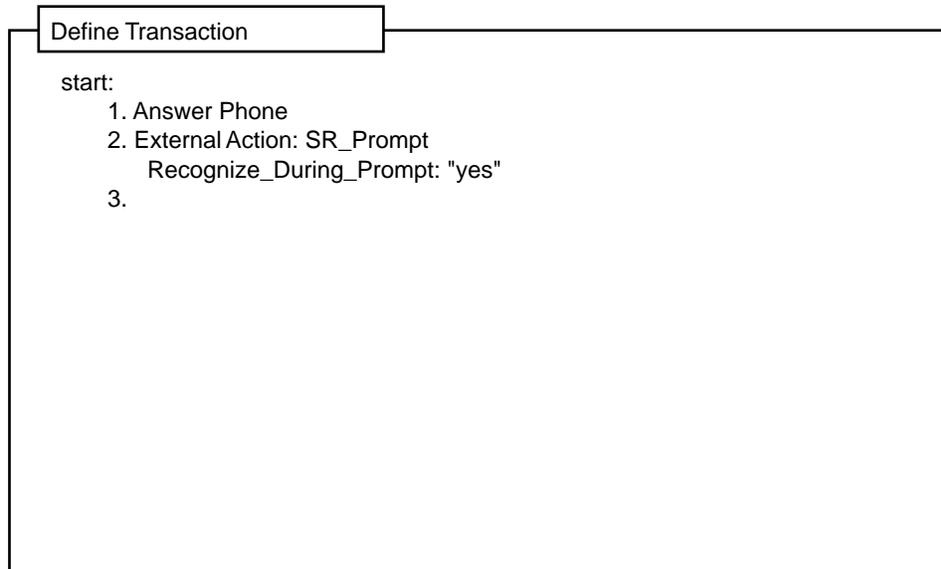


Figure 3-4. Expand External Action: SR_Prompt Action Step

SP_Allocate External Action

The script uses the SP_Allocate external action to help ensure that the 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 allocate explicitly the resource to the transaction until the transaction terminates or explicitly to deallocate the resource.

⇒ 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, “WholeWord Bilingual Speech Recognition”.

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. (See Figure 3-5.)

```
Define SP_Allocate
Speech Recognition Allocation: on
Speech Recognition Type: WW_RECOG
Return Field: 
```

Figure 3-5. Define SP_Allocate Form

5. To enable Speech Recog Allocation, enter **on** in the field, then press **(ENTER)**. To disable Speech Recog Allocation, enter **off** in the 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 3-6). This example assumes “RET_VALUE” was entered in the Return Field: field.

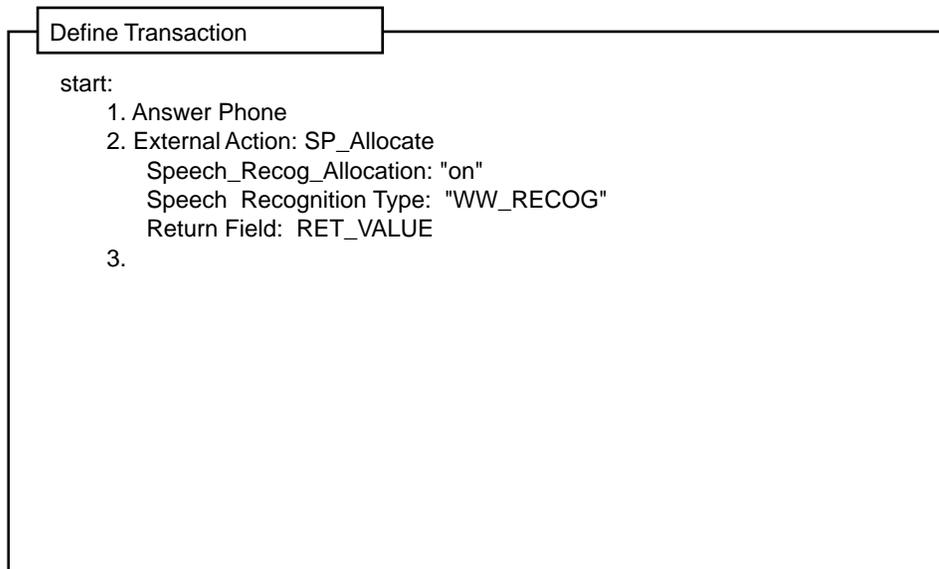


Figure 3-6. Expand External Actions: SP_Allocate Action Step

Using WholeWord Bilingual 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 Instruction for Speech Recognition

⇒ NOTE:

If the system handling incoming requests is overloaded with inefficient speech recognition resources, error message numbers SPIP001 and SPIP004 may appear. To prevent this condition, the script should check for an echo cancellation before proceeding with the recognition. If an echo cancellation is not used, the caller will not be able to barge-in to the script and speech breaks occur.

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 grammar. The choices available for *type* in this instruction can be found in the file `/att/include/sr_grammar.h`. The grammars and their specifying values are listed in Table B-2 of Appendix B.

⇒ NOTE:

For packages which support connected-digit recognition: For US English, you may use `US_0_9o` to recognize any variable-length string of 1-24 digits. If the string length is known in advance, however, superior recognition performance can be obtained by using one of the grammars with a fixed string length.

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 vocabulary provided. Possible characters are listed in Table B-3 of Appendix B.

⇒ NOTE:

For packages which support connected-digit recognition: This maximum value of the string length for speech input is 24.

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 listed in Table B-4 and Table B-5 in Appendix B.

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 resources can be added are listed in Table B-6 and Table B-7 of Appendix B.

The `sr_talkoff` Script Instruction for Speech Recognition

The script instruction **`sr_talkoff`** is used to enable/disable speech recognition during the prompt. The format for the instruction is as follows:

`sr_talkoff(im.1|im.0)`

If speech recognition during the prompt is enabled by using **`sr_talkoff(im.1)`**, the **`getdig`** instruction will begin to play any phrases in its queue and simultaneously turns on the recognizer. If the recognition during prompt is disabled by using **`sr_talkoff(im.0)`**, first, the **`getdig`** plays any phrases in its queue, and then it turns on the recognizer.

If recognition during prompt is enabled, the call can be received through a telephony interface card (T1, IVP4, IVP6, or VRS6) that is connected to the TDM bus with the “tdm” option set. Enabling **`sr_talkoff`** requires the use of the SP card to play the prompts. This is already set for T1, and VRS6 cards are set to “tdm.” However, for IVP4/6 cards set to “talk,” the system detects a “recognition during prompt” request in the script and automatically uses an SP to play the prompts. Settings and their results are as follows:

“talk” set + **`sr_talkoff`** “off” = the system plays all prompts with the IVP4/6 card

“talk” set + **`sr_talkoff`** “on” = the system plays all prompts with the SP card

Playing prompts uses resources on the SP card, so application developers who find that the SP resources are strained may want to consider configuring their IVP4/6 cards with “talk” and designing the application to use as few speech processor resources as possible.

Script register 0 (r.0) is set to the following values to indicate the status of **sr_talkoff** execution:

0	Success
-1	Failure
-2	System resource not available

Summary of WholeWord Bilingual Speech Recognition Commands

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
- sr_talkoff

⇒ 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 grammar. The choices available for *type* in this instruction can be found in the file **/att/include/sr_grammar.h**. The grammars and the values that specify them are contained in Table B-2 of Appendix B.

⇒ NOTE:

For packages which support connected-digit recognition: For US English, you may use **US_0_9o** to recognize any variable-length string 1-24 digits long. If the string length is known in advance, however, superior recognition performance can be obtained by using one of the fixed-string length.

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. (The maximum value is 24.) 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 vocabulary specified, and these null terminated characters are contained in Table B-3 of Appendix B.

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 and speech input are contained in Table B-4 and Table B-5 of Appendix B.

Example

`getdig (US_1_3n, ch.F__CI_VALUE, im.01,int.F__CI_MODE)`

In this example, the script accepts 1-3 spoken digits or the word “no” in a string no longer than the length defined in `im.01`. 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 B-6 and Table B-7 in Appendix B.

sr_talkoff

Synopsis

This script instruction enables/disables speech recognition during prompt.

Command Format

sr_talkoff(im.1|im.0)

Description

A new script instruction for speech recognition, **sr_talkoff**, is used to enable/disable speech recognition during the prompt. When speech recognition during prompt is enabled using **sr_talkoff(im.1)**, the **getdig** instruction starts playing any phrases in its queue and simultaneously turns on the recognizer. When speech recognition during prompt is disabled with **sr_talkoff(im.0)**, **getdig** plays any phrases in its queue, then turns on the recognizer. Caller speech or touch-tone input interrupts any of the phrases that were started with the **getdig**. If a **tflush** instruction is used to initiate phrases immediately before the **getdig**, the recognize during prompt does not apply to those phrases (because the recognizer is not on).

IF	THAN
speech recog. is enabled with sr_talkoff(im.1) ,	getdig plays phrases and <i>simultaneously</i> turns on the recognizer.
speech recog. is disabled with sr_talkoff(im.0) ,	getdig plays phrases <i>then</i> turns on the recognizer.
tflush starts playing phrases before getdig ,	Recognize during prompt does not apply. (Recognizer is not on.)

If recognition during prompt is enabled, the call can be received through a network interface card (T1, IVP4, IVP6, or VRS6) that is connected to the TDM bus with the “tdm” option set. Enabling **sr_talkoff** requires that the SP card play the prompts. The T1 and VRS cards are already set to “tdm.” However, for IVP4/6 cards set to “talk,” the system detects a “recognition during prompt” request in the script and automatically uses an SP to play the prompts. If an IVP4/6 card has the “talk” option set and **sr_talkoff** is off, the system will play all prompts with the

IVP4/6 card. If an IVP4/6 card has the “talk” option set and **sr_talkoff** is on, the system will play all prompts with the SP card.

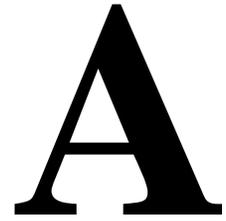
Because playing prompts uses resources on the SP card, application designers who find that the SP resources are strained may want to consider configuring their IVP4/6 cards with “talk” and designing the application to use as few speech processor resources as possible.

Example

sr_talkoff(im.1)

In this example, speech recognition is enabled during the prompt.

WholeWord Bilingual Speech Recognition



Supported Recognition Types

Supported recognition types are listed in Table A-1.

Table A-1. Supported Recognition Types

WholeWord Recognition				
Digit Lengths	US English	UK English	Canadian French	Mexican Spanish
"Yes" or "No"	US_YN	UK_YN	CF_YN	MS_YN
One character 1-3	US_1_3	UK_1_3	CF_1_3	MS_1_3
One character 1-3 or "no"	US_1_3N	UK_1_3N	CF_1_3N	MS_1_3N
One character 1-5	US_1_5	UK_1_5	CF_1_5	MS_1_5
One character 1-5 or "no"	US_1_5N	UK_1_5N	CF_1_5N	MS_1_5N
0-9	US_DIG	UK_DIG	CF_DIG	MS_DIG

⇒ NOTE:

Intended for **PACKAGES WHICH SUPPORT CONNECTED-DIGIT RECOGNITION**

Connected-digit recognition uses the same recognition types as digit string lengths 0-9. Check min/max values to determine if a string length is fixed or variable. If min=max, string length is fixed. If min<max, string length is variable. US English, for example, uses the following type:

US English Connected-Digit Recognition

Recognition Type	Length
US_DIG	Variable (1-24) 0-9 and "oh"

Vocabularies of Supported Languages

The vocabularies for all supported languages are as listed in Table-A2. The US English speech recognition feature supports spoken English from any US region. The UK English recognition feature supports England, Scotland, Wales, and Northern Ireland. Canadian French is designed to support French spoken in the province of Quebec. Mexican Spanish is designed to support spoken Spanish input collected throughout the country of Mexico. The vocabularies for these languages consist of the following words:

Table A-2. Vocabularies of Supported Languages

US English	UK English	Canadian French	Mexican Spanish
zero	zero	zéro	cero
Oh (= zero)	Oh (= zero)	—	—
one	one	un, une	uno
two	two	deux	dos
three	three	trois	tres
four	four	quatre	cuatro
five	five	cinq	cinco
six	six	six	seis
seven	seven	sept	siete
eight	eight	huit	ocho
nine	nine	neuf	nueve
yes	yes	oui	sí
no	no	non	no

**WholeWord Bilingual Speech
Recognition**

B

SCI_VALUES

The following table contains \$SCI_VALUE variables:

Table B-1. :\$SCI_VALUES for Recognize and Return

Supported Language	Recognize	Return
All supported languages	0	"0"
All supported languages	1	"1"
All supported languages	2	"2"
All supported languages	3	"3"
All supported languages	4	"4"
All supported languages	5	"5"
All supported languages	6	"6"
All supported languages	7	"7"
All supported languages	8	"8"
All supported languages	9	"9"
US English and UK English only	Oh	"0"
All supported languages	Yes	"Y"
All supported languages	No	"N"
For packages which support connected-digit recognition only	1234	"1234"
All supported languages	Not understood	"?"

'type' Field Values

The following table contains the grammars and specific values for all supported languages. (The shaded units represent values which are not supported.)

Table B-2. 'type' Field Values:

Range	US English	UK English	Canadian French	Mexican Spanish
Yes or No	US_YN	UK_YN	CF_YN	MS_YN
Digits 1-3	US_1_3	UK_1_3	CF_1_3	MS_1_3
Digits 1-3, no	US_1_3N	UK_1_3N	CF_1_3N	MS_1_3N
Digits 1-5,	US_1_5	UK_1_5	CF_1_5	MS_1_5
Digits 1-5, no	US_1_5N	UK_1_5N	CF_1_5N	MS_1_5N
One digit 0-9	US_DIG	UK_DIG	CF_DIG	MS_DIG
One digit 0-9 and "oh"	US_1dig	UK_1dig		
Two digits 0-9, oh	US_2dig			
Three digits 0-9, oh	US_3dig			
Four digits 0-9, oh	US_4dig			
Five digits 0-9, oh	US_5dig			
Six digits 0-9, oh	US_6dig			
Seven digits 0-9, oh	US_7dig			
Eight digits 0-9, oh	US_8dig			
Nine digits 0-9, oh	US_9dig			
Ten digits 0-9, oh	US_10dig			
One to twenty-four digits 0-9, oh	US_0_9o			

Null Terminated Characters

Possible null terminated characters are listed in the following table:

Table B-3. Null Terminated Characters

Supported Language	Recognize	Return
All supported languages	0	"0"
All supported languages	1	"1"
All supported languages	2	"2"
All supported languages	3	"3"
All supported languages	4	"4"
All supported languages	5	"5"
All supported languages	6	"6"
All supported languages	7	"7"
All supported languages	8	"8"
All supported languages	9	"9"
US English and UK English only	Oh	"0"
All supported languages	Yes	"Y"
All supported languages	No	"N"
For packages which support connected-digit recognition only	1234	"1234"
All supported languages	Not understood	"?"

Touch-Tone and Speech Input Values

Table B-4 contains the values for touch-tone input. Table B-5 contains the values for speech input.

Table B-4. Return Values for Touch-Tone Input

$N > 0$	If the <i>number</i> 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 B-5. Return Values for Speech Input

$N > 0$	Speech was heard and recognized. In this case, N represents the length of the string containing the word recognized. For connected digits only, this value can be greater than one.
$N = 0$	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 B-6. 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 B-7 for examples.

Table B-7.

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

A

Accuracy
 calculation of, 1-17
 factors influencing, 1-16
 for isolated words, 1-17
 for variable-length digit strings, 1-17
Action Data field, 3-4
Action required, 1-19

B

Barge in, 1-4, 1-9, 1-11, 1-21, 3-7
Bilingual, see Speech recognition

C

Call bridge action, 3-7
Channels, number of supported, 1-9
CMP cards
 channels supported, 1-9
 jumper settings, 2-2
 scoring word matches, 1-3
 switch settings, 2-2
Companion cards, see CMP cards
Confirming and reprompting, 1-18
Connected-digit recognition, 1-4
Custom grammars, 1-18

D

Data interface processes, see DIP
Define Prompt and Collect form, 3-2, 3-4
Desired result, 1-19
Digit strings
 connected-digit, 1-23
 fixed-length, 1-21
 variable-length, 1-21, 1-23
DIP, 1-7

E

Echo cancellation, 4-2
Environment, 1-22
Error messages
 SPIP001 and SPIP004, 4-2
Experienced calling populations, 1-18
Extraneous speech, 1-17

F

Fixed-length digit strings, see Digit strings, fixed length

G

getdig
 ch.F__CI_VALUE, 5-3
 command format, 5-2
 ctype.dst argument, 4-2
 ctype.mode argument, 4-2
 instruction, 1-10, 4-2, 4-3, 5-6
 int.F__CI_MODE, 5-3
 number argument, 4-2
 type argument, 4-2
Grammars, 1-3, 1-6

H

Hardware requirements, 1-9

I

Input mode, specifying of, 3-2
Inputting of information, 1-22
Interface cards, 4-5, 5-6
IVP6 cards, 1-13

K

Key words, 1-22

L

Languages, see Supported languages
Loading software, see Software installation

M

Max_digits, 1-21, 1-23
Messages, see Error messages
Min_digits, 1-21, 1-23
Minimum and maximum digits, setting of, 3-3
Misrecognition vs. recognition, 1-23
Models, 1-5

N

Natural numbers, problems with speaking, 1-17
No recognition, 1-23
Nonvocabulary words
 interjection of, 1-16
 word spotting of, 1-23
Not on List, default setting of, 1-23
Number entry, 1-14

P

Prompt and Collect checklist, 1-23, 3-6
Prompts
 follow-up, 1-20
 informative, 1-19
 menu, 1-19
 structure, 1-19
 yes/no, 1-20

R

Range of connected digits., 1-15
Recognition accuracy, see Accuracy
Recognition types
 definition of, 1-5
 supported, 1-8
Recognize During Prompt, 1-9, 1-11, 1-18, 3-7
Return field values, 3-8, 3-11

Return variables
 \$CI_MODE, 3-6
 \$CI_VALUE, 3-6

S

Script Builder, 3-3
Signal processing card, see SP cards
Software installation
 of one language, 2-4
 of second language, 2-5
 of speech recognition package, 2-4
Software removal of any language, 2-5
SP cards
 assigning functionality to, 2-6
 channels supported, 1-9
 description of, 1-3
 MANOOS state, 2-6
 resource values, 5-5
 supporting CMP cards, 2-2
SP resource
 adding values, B-7
 allocation, B-7
 deallocation, B-7
sp_alloc
 allocation values, 5-5
 command format, 5-4
 deallocation values, 5-5
 status values, 5-5
 type.onoff argument, 4-4
 type.resource argument, 4-4
 values, 4-4
SP_Allocate external action, 3-10
Speech recognition
 bilingual, 1-2, 1-4
 description of, 1-2
 resources, 1-9
 sub-word, 1-2
 whole-word, 1-2
sr_alloc replaced by sp_alloc, 4-4
SR_Prompt
 external action, 3-7
 set to no, 1-9
 set to yes, 1-9
 with sr_talkoff, 1-11
sr_talkoff
 command format, 5-6, 5-7
 instruction, 4-5
 script instruction, 1-11
 status values, 4-6
 when SR_Prompt is set to no, 1-9
 when SR_Prompt is set to yes, 1-9
Stopping the VIS, see VIS, stopping of

Supported languages, list of, 1-3

T

tdm function, setting of, 1-13

U

US English CONFIRM prompt, specifying of, 3-4

V

Validation of data, 1-18

Variable-length digit strings, see Digit strings variable length
and Accuracy

VIS, stopping of, 2-4

VRS6 cards, 1-13

W

Word spotting

 bilingual examples of, 1-14

 capabilities, 1-4

 filtering out nonvocabulary words, 1-23

 recognizing nonvocabulary words, 1-23

 variable-length recognition types, 1-23

Wordlist, 1-3

