

Speech Recognition Operating Specification

Recognition and Training

Recognizer is a training-based system. That is, the user must speak each command word to it one or more times for Recognizer to be able to recognize and act on the voice command. There are two ways to train words for Recognizer: using the Edit Vocabulary command and training on-the-fly.

Vocabulary files can be opened and saved via the File menu. The files normally have a .SPV extension. These files in general will be specific to the user, computer, and sound card used to create them. A vocabulary file created under one configuration may not work well under another.

Multiple vocabulary files can be maintained, if for instance the computer is used by different people. Recognizer can be started by "running" a .SPV file, much like a .WRI file launches Write.

There is a limit of about 100 words per vocabulary, although this depends somewhat on the length of the words stored.

Editing the Vocabulary

The edit vocabulary command displays the following dialog box:

[Vocabulary dialog box]

The user can delete words from the vocabulary by selecting them from the list and pressing the Delete button.

To add a new word, it must first be recorded. Press record, speak the word into the microphone, and press Stop when done. Play will play back the current recording.

The Add to vocabulary... button will be enabled after a word is recorded. Pressing it will add the word to the vocabulary list. A dialog will appear with a list of words already known. One of these can be chosen, or if the word is new, it should be typed in the dialog. Recognizer will highlight the word it thinks was recorded:

[Add Vocabulary dialog box]

Press OK to add the word to the vocabulary.

For best results, you should re-record and Add... each word three times. Each word can have up to three trainings, which are all used when attempting to recognize the word.

Training-On-The-Fly

To train Recognizer on the fly, simply press the right-mouse button, speak the word, and release the mouse button, just as voice commands are normally executed. If recognizer doesn't recognize the word, it will put up a dialog asking the user to identify it:

[I'm not sure what you said dialog...]

The user can choose from the list of known words to enter a new word. In either case, pressing OK will add the word to the vocabulary. If the word was already known, OK will save another

training of the word, up to three per word. Pressing cancel will ignore the voice command.

Activating Voice Commands

Because of limitations in most PC sound boards, Recognizer uses the right mouse button as a substitute for a button or VOX-like circuitry in the microphone. To perform a voice command, the user must hold down the right mouse button, speak the command word into the microphone, and release the right mouse button. Recognition begins as soon as the button is released.

Note that Recognizer's icon will change to reflect whether Recognizer is idle, recording, or recognizing.

If Recognizer determines which word was spoken with sufficient confidence (see Preferences), it will attempt to execute the command associated with the word. Local, then global, commands are considered. As discussed in the Commands section, commands which Recognizer knows are invalid in a given context are not considered.

If Recognizer cannot determine which word was spoken, it goes into Training-On-The-Fly mode. This is described in the Recognition and Training section.

Correcting Recognitions

If the Recognizer incorrectly recognizes a word, the user can train it using the Edit Correct Last Recognition. This command will not undo the command which may have incorrectly been recognized, but it will allow the user to improve Recognizer through training. This command will show a dialog box which lists all known words, and highlights the one Recognizer thought was correct. The user can instead choose the word which was really said, or type in a new word if it isn't in the list. The previously-recognized phrase will be added to the vocabulary under this word.

Commands

Overview

Commands determine what Recognizer does when the user speaks a word and it is successfully recognized. All commands are stored in RECOG.INI, with the following form:

```
<word>=<command>
```

where <word> is the word as it was saved when training the vocabulary and <command> is a command, as described below.

There are two classes of commands: Global commands and local commands. Global commands affect all running Windows applications, while local commands affect only the currently running application. The type of command is determined by its section in RECOG.INI. Global commands are in the section [Global commands], while local commands are in a section which specifies the application or module .EXE or .DLL, such as [NOTEPAD.EXE].

When determining which command to execute, Recognizer first looks for any appropriate local command. If it can't be found, or is the G command, the global command for the same word is executed instead.

Recognizer currently supports several types of commands. A command's type is specified by the first letter of <command>; additional arguments follow (without an intervening space).

Button commands

Form: *Bbuttonlabel*

Button commands simulate the pressing of a pushbutton. For example, Bok would press the OK button of the currently active dialog. The buttonlabel need not include any underlines (&'s) or periods (...), as these are removed from any button before it is compared with the command.

Menu Commands

Form: *Mmenu1\menu2\menu3*

Menu commands simulate the selection of a menu item. Only command items (that is, those which do not open another popup menu) may be specified. The argument to the menu command specifies the "path" of the menu, starting with the top-level menu. For example, File\Print\All would simulate selecting the File menu, then the Print option, then the All item from a cascading menu. As with button commands, the menu of the active window is considered, and underlines and ellipses are ignored. In addition, accelerator keys which appear in the menu should not be included in the command. For example, if a menu has an &Open...\tCtrl-O identifier, Open will suffice in the menu command. If the top-level identifier is a hyphen, the the system menu will be accessed.

Launch Commands

Form: *Lpathname*

Launch commands launch an application. The full path name of the application's .EXE file should be given. If the application is in the path, only the file name need be specified.

Key Commands

Form: *Kkeystrokes*

Key commands simulate the user typing characters into the window which has keyboard focus. In addition to ASCII characters, the keystrokes can include special keys, enclosed in {}'s. For example {Alt} simulates the Alt key and {Ctrl} the control key. A full list of these will appear elsewhere.

System Commands

Form *Sc*

System commands carry out special commands which affect Windows applications. *c* is a one-letter code which refers to the command to execute. Currently, these are:

X	Maximize active window
N	Minimize active window
R	Restore active window
T	Bring up task list

- Q Send Quit message to active window (exits most apps)
- C Close active window
- S Activate Windows 3.1 screensaver

Advanced Commands

Form: *Acommand1;command2;command3*

Advanced commands allow multiple commands to be executed by one word. The form of the sub-commands is identical to that of main commands. Note that semicolons cannot appear in the sub-commands. For Key commands, use {semi} for semicolon.

Do nothing

Form: X

This command does nothing. It can be used to prevent a global command from being executed in a local context. This is the default command for global commands as well.

Do Global Command

Form: G

This is valid only in local commands. This command will cause the global command for the same word to be executed instead. This is the default for local commands.

Command Pruning

To improve recognition, words are removed from consideration if it can be determined in advance that they are invalid in the given context. The following rules apply:

Button commands:

The button must exist in the active window and be enabled

Menu commands:

The menu command must exist in the active window's menu and be enabled

Key commands:

Some window must have keyboard focus

Launch commands:

Always valid

System commands:

N and X depend on the active window; all others are always valid.

Advanced commands:

Always valid (since the sub-commands could change activation, etc.)

Do nothing

Never valid

Do global command
Valid if global command is valid

User Editing of Commands

Adding and Editing Voice Commands

Recognizer allows commands to be added or edited without having to know the low-level command format. Before adding a command, Recognizer must be trained for the word that the command will be activated by.

Global commands can be added using the Edit Global Commands... menu item. Local commands can be edited by selecting Voice Commands... from the system menu of the application the local commands will apply to. The actual process of adding or editing commands is nearly identical in these two cases. A dialog similar to the following will appear with either of these commands:

[Voice commands dialog]

For local commands, the application name that the commands apply to will appear in the dialog's title bar.

The user first should choose the word that will activate the command from the Word list. The Command: area will show what the current command for this word is. By default, global commands are set to Do nothing, while local commands are set to Do global command. Recognizer comes pre-configured with several global and local commands for certain words, such as close, open, save, and so forth.

The Type list lets you choose which type of command the word will activate. It will initially reflect the type of the command currently associated with the selected word. The choices are:

Do nothing
Launch Application
Menu command
Pushbutton command
System command
Send keystrokes
Advanced command

For local commands, another option will be listed:

Do global command

Once the type is selected, the Edit command... button can be used to edit the specific parameters of the command (except for Do nothing and Do global commands, of course). The specifics for each type of command are described in the following sections.

Launch Application Commands

A standard Windows file dialog allows the user to choose the .EXE file of the application that the command will launch. In addition, any file which is associated with an .EXE file may be chosen instead. Optional command line parameters can be specified.

[Launch App dialog]

Menu Commands

The user can fill in the top-level and sub-menu labels to describe the menu item that the command will execute. The System (command) menu checkbox lets the user choose between the standard application menu bar and the system menu. If more than three levels of cascading are needed, they can be specified in the third menu box, using the "path" notation described under Commands.

[Menu dialog]

Pushbutton Commands

The user can specify which text appears on the pushbutton that will be pressed when this command is executed.

[Button command dialog]

Key Commands

The user can type the characters that will be sent to the application when this command is executed. In addition, codes for special keys can be inserted by choosing them from the list and pressing the Insert button.

[Key command dialog]

System Commands

The user can choose which system command the voice command should execute. The options are similar to those described in System Commands in the Commands section.

[System Commands]

Advanced Commands

The user can enter any command string in the low-level command format. It is not necessary to enter the A for the command.

[Advanced commands dialog]

Preferences

The Edit Preferences dialog allows the user to change certain features of the Recognizer.

[Preferences dialog]

Co-processor Mode

Recognizer will use a numeric co-processor (80x87, 486) on a machine which has one by default.

If a user plans on using the same speech vocabulary on different machines, some with a co-processor and some without, the user may want to choose the No coprocessor mode. Also, this mode is somewhat faster than the co-processor mode. On the other hand, the Use coprocessor mode is slightly more accurate at recognition. It is also possible to use the Use coprocessor mode on machines without a coprocessor, but the result will generally be very slow recognition.

Confidence Threshold

The confidence threshold determines how confident Recognizer must be that it correctly recognized a word before executing its command. Any recognitions less than this threshold will cause the "I don't know what you said" dialog to appear, allowing the user to choose a word from the list or train a new word. The confidence is never more than 95, so a 100 threshold will cause all recognitions to be doubted. The default for this value is 85

Board Zero-Level

Some low-end sound boards do not record sounds with a proper zero level. This value should be set to 128 for most boards. Small variations in a sound board's zero level will not adversely affect recognition. In most cases, this value should be left alone, or set using the Initialize Board function. This setting is mainly intended to compensate for irregular hardware.

[Other parameters will be added]