WORDSTAR

MACROS

Combining relevant sections from
the WordStar 7 manuals
USING WORDSTAR
and REFERENCE

CHAPTER 8

USING MACROS TO EDIT

In this chapter you'll learn about

- ❑ Recording a Macro
- ❑ Playing a Macro
- ❑ Editing a Macro
- ❑ Creating a Macro

Macros help you automate word processing tasks. They can be used to type frequently used text or to execute a sequence of commands. You create a macro by recording and saving your keystrokes. When you play the macro, the keystrokes that you recorded are carried out. You can edit the macros to correct any errors or customize them.

RECORDING A MACRO

When you record a macro, WordStar captures and saves the keystrokes in a macro file. When you press the hot key you assign to the macro, the macro executes the keystrokes that you saved.

Follow the steps below to record a macro that saves the current file, opens a window on another file, and sizes the window. You'll call the macro "Window."

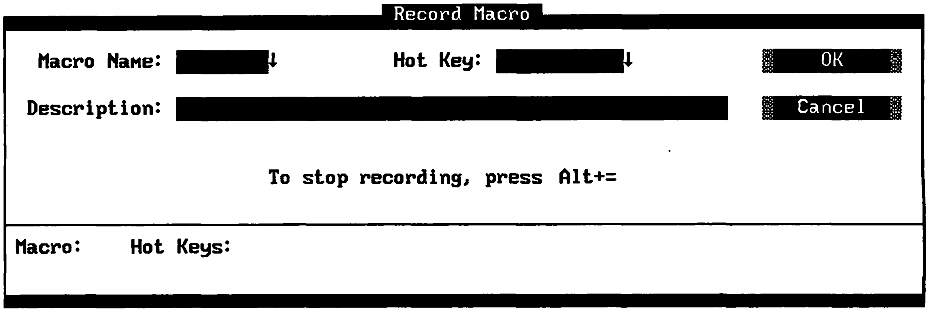
1 Choose File→Open Document (F,D) and name the file MACRO.DOC.

2 Type the sample text below:

```
WordStar stores each macro as a file, with the name you
gave it while recording or editing and the .WSM extension.
```

3 Choose Utilities→Macros→Record (Alt+U,M,R) to record a macro.

The Record Macro dialog box appears.



4 In the Macro Name box, type **window**.

Note: When you name a macro, use a name that will help you identify it. You can use up to eight characters. WordStar assigns the .WSM extension to the filename.

5 Press the Tab key.

The cursor moves to the Hot Key box. Now you'll assign the single key or combination of keys for running the macro. For more information about hot key assignments, see the *Reference* manual.

6 Press the Down Arrow to choose a hot key from the list of available hot keys.

7 In the Hot Key list, move the cursor to Alt+W and press Enter.

Alt+W appears in the Hot Key box and the cursor moves to the Description box.

8 In the Description box, type a description of the macro. Type **Save file-Open window-Size window.**

You can use as many as 48 characters, including spaces, for a macro description.

Now that you've named and described the macro, you're ready to record the keystrokes.

9 Press F10 or choose OK to begin recording keystrokes.

(Ignore the *End Recording* message at the top of the screen.)

Now you'll record the following sequence of keystrokes for the macro:

1 Press Alt+F,S to save the file MACRO.DOC.

2 Press Alt+F,W to open a window on a second file.

The Open Document dialog box appears.

3 Press Alt+Hyphen (hold down the Alt key while you press the hyphen key) to insert a pause for input command into the macro. When you play back the macro later, the macro will pause while you enter a filename.

4 Type **REVIEW.DOC** for the second filename.

5 Press Alt+hyphen to end the pause for input.

The macro resumes recording.

6 Press F10 or choose OK to open REVIEW.DOC.

7 Press Alt+V,W to size the window.

8 In the Change Window Size dialog box, press F to make each window use the full screen. Press F10 or choose OK to close the dialog box.

Each window now uses the full screen.

9 Press Alt+Equal (hold down the Alt key while you press the Equal key) to stop recording.

10 Close and abandon REVIEW.DOC. Then close and abandon MACRO.DOC.

You now have a macro that saves a file, opens a second window, and sizes the window to fill the entire screen.

PLAYING A MACRO

When you *play* a macro, the commands recorded in the macro are carried out. You can play a macro by using a menu command or the assigned hot key. In the following section, you'll play the WINDOW macro by using the hot key you assigned.

1 Open the MACRO.DOC file.

2 Press Alt+W to play the WINDOW macro. When the Open Document dialog box appears, the macro pauses for input.

3 Type a filename for the second window and press Alt+Hyphen to end the pause for input. The second window is opened and sized.

4 Close and abandon both files.

EDITING A MACRO

After you record a macro, you may want to change it. WordStar has a macro language that you can use to edit a macro.

Follow these steps to edit a macro:

1 At the Opening screen, choose Utilities→Macros→Edit/Create (U,M,E).

The Edit/Create Macro dialog box appears.

2 In the Macro Name box, type **window**.

3 Press F10 or choose OK to edit the macro.

The WINDOW macro (WINDOW.WSM) is opened.

In every macro file, you'll see the command "Sub MAIN" which is a subroutine. A subroutine is a collection of statements that are invoked by the Sub command. Each macro must contain the Sub MAIN subroutine.

Subroutines usually contain key functions. A key function tells the macro which keys to play back. The format for a key function is:

```
Key("key stroke string")
```

In the Window macro, you'll see these key functions near the end of the file.

```
KEY ("{F10}{Alt+V}w")
KEY ("f")
```

The first key function above displays the Change Window Size dialog box. The second key function changes the window size to full screen.

In the following steps, you'll put a comment at the beginning of the WINDOW.WSM file to provide descriptive information about the macro. A comment is a statement for you to read but is not a command for the macro to execute.

1 At the beginning of line 1, type this text:

```
rem  Saves current file, opens and sizes second window.
```

2 Press Enter.

In the following steps, you'll change a key function so that the window takes up only five lines instead of the full screen.

The key function looks like this:

```
KEY ("f")
```

1 Move the cursor to the "f" in the key function.

2 Delete the "f" and type **L5**.

3 Choose File→Save and Close (Alt+F,L) to save the macro.

4 Open a file and the WINDOW macro to see the changes you just made. (If you need help, see "Playing a Macro" in this chapter.)

If WordStar finds errors, it returns you to the macro file to correct them.

CREATING A MACRO WITH THE MACRO LANGUAGE

In this chapter, you've created a macro by recording your keystrokes. Now you'll create a macro using the macro language.

In the following steps, you'll create a macro that runs a spelling check, saves your document, and then runs Preview.

1 At the Opening screen, choose Utilities→Macros→Edit/Create (U,M,E).

2 Assign a macro name, for example, SPELL. WordStar gives the file a .WSM extension.

3 Assign a hot key such as Alt+3 and type a description.

4 Press F10 or choose OK to open the macro.

WordStar opens the new macro.

5 On line 1, type this text:

```
rem  Runs spelling check, saves file, runs Preview.
```

6 Press Enter.

7 Type **Sub MAIN** and press Enter. This marks the beginning of the main subroutine.

8 On the next two lines, type the sample statements shown below. (You can type these statements in all lowercase; the case used below is for readability only.)

Note: "AutoRestore(ON)" is an example of a statement. When there is more than one statement on a line, the statements must be separated by colons. When you type statements, separate them with a space, a colon, and another space.

```
AutoRestore(ON) : SetHelpLevel(4) : Insert(ON)
CmdTags(OFF) : HideDots(ON) : ColMode(OFF) :
ColReplace(OFF)
```

9 Press Enter.

The statements establish these settings:

AutoRestore(ON) Restores original settings after the macro plays.

SetHelpLevel(4) Sets the help level at 4.

Insert(ON) Allows characters to be inserted into existing text instead of overwriting text.

CmdTags(OFF) Turns off the display of command tags.

HideDots(ON) Turns off the display of dot commands.

ColMode(OFF) Turns column mode off. This is the default for marking blocks.

ColReplace(OFF) Turns off column replace. This is the default for moving and copying blocks.

In the following steps, you'll type two key functions and some exception conditions:

The first key function that you'll type is:

```
Key("{Alt+U}S")
```

1 Type **Key("** to begin the Key function. Every Key function begins with **("** and ends with **")**.

2 Type **{Alt+U}S** to move the cursor to the beginning of the file and start the spelling check.

3 Type **")** to end the function. Press Enter to move to the next line. Press the Tab key to indent the line.

4 Type **IfException** and press Enter.

When a spelling check is complete, WordStar displays a message giving the word count. This is an exception condition. When a macro encounters an exception condition, you can have the macro pause while you provide required input or you can tell the macro to provide the input. To have the macro provide input, you add an IfException clause to your macro statement. In the following steps, you tell the macro to issue an Escape when the word count message is displayed.

5 On the next line under "IfException" type **ACK: Key("{Esc}")** and press Enter.

The ACK clause issues an Escape key command when the spelling check message appears.

6 To end the IfException clause type **End IfException** and press Enter.

7 Type **Key("{Alt+F}S")** and press Enter. This key function saves the file.

8 Type **Key("{Alt+V}P")** and press Enter. This key function runs Preview.

9 Type **End Sub** to end the subroutine and press Enter.

When your macro is finished, it will look like this:

```
rem  Runs spelling check, saves file, runs Preview
Sub MAIN
AutoRestore(ON) : SetHelpLevel(4) : Insert(ON) :
CmdTags(OFF)
HideDots(ON) : ColMode(OFF) : ColReplace(OFF)
Key(″{Alt+U}S″)
           IfException
           ACK: Key(″{Esc}″)
End IfException
Key(″{Alt+F}S″)
Key(″{Alt+V}P″)
End Sub
```

10 Save the macro and then play it in a document.

WORDSTAR

MACROS

From the WordStar 7

REFERENCE Manual

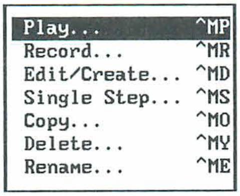
MACROS

***Pull-down:* ALT + U, M**
***Classic:* ^M**

Records, creates, edits, or runs a macro that enters a longer series of text and commands.

The WordStar macro capabilities allow you to use a few keystrokes to enter a series of keystrokes and commands. For example, you can create a macro so that each time you type Alt+W, the word "WordStar" is inserted in your text. Or a macro can change the order of two words (mark the first word as a block, move the cursor to the end of the next word, and move the block).

When you choose Macros, the following menu appears.



You can create a macro in either of two ways:

- Use Record, or the ^MR classic command, to record keystrokes. This is the easier way to create a macro. WordStar records commands and keystrokes as you enter them. When you are finished, WordStar compiles them into a macro file that contains all the appropriate macro commands.

- Use Edit/Create, or the ^MD classic command, to enter macro commands into a file that you edit like a nondocument. To use this command, you need to know the WordStar macro language. This language allows you to create macros that are too complex to record with the Record command. For example, you can have a macro repeat a certain action as many times as needed each time you run the macro. The example in "The Macro Language" in this section enters a bullet character and waits for you to type text, then repeats that sequence after the end of each paragraph; the macro stops and returns you to editing when you finish your bulleted list.

If you need to write a macro that is too complex to record, you can first record a simpler macro that contains commands and keystrokes that can be recorded, then use Edit/Create to add the macro language for the rest of the macro. You can also copy a macro and edit it to create a similar one, and you can copy blocks of macro commands from one macro file to another.

After you create a macro, you can use Single Step, or the ^MS classic command, to play it back step by step to identify any problems.

To execute a macro, you can either use a hot key, or use Play, or the ^MP classic command. You can stop the macro by pressing Esc or ^Break. (When a macro is paused for input, you must use ^Break.) You can change the Esc keystroke or add another key to stop a macro in WSCHANGE (DCDC and DCDD).

You can also copy, rename, and delete macros using commands on the Macros menu.

TIP *You can set up a macro to run a series of commands that you frequently use when you start WordStar. Then you can run the macro immediately when you start WordStar by typing* **ws /m*macroname***. *For example, you can set up a macro named LETTER to change to the C:\CUSTOMER directory, open a new file, and read in a file of boilerplate text and formatting. Then, any time you want to start WordStar and start a new letter, type* **ws /mletter**. *If you want to start WordStar, open a file, and then run a macro, type the filename before the* **/m** *option.*

WORKING WITH MACROS

MACRO HOT KEYS

You can execute a macro by pressing a single key or key combination called a hot key. You can assign a hot key to a macro when you create it. Eighty-two hot keys are available:

- The function keys F2 through F9, and F11 and F12. (F1 and F10 are reserved for help and closing dialog boxes.)
- The function keys F1 through F12 combined with the Shift key.
- The function keys F1 through F12 combined with the Ctrl key.

- The function keys F1 through F12 combined with the Alt key.
- The Alt key combined with the letters A-Z and the digits 0-9.

Keep these points in mind about macro hot keys:

- The Alt key combined with the letters F, E, V, I, S, L, U, and H access the WordStar pull-down menus. If you use any of these keystrokes as hot keys, you can't use them for those menus. (You can still display the menus using a mouse or using the Right and Left Arrow keys in the menu bar.)
- You don't have to assign a hot key to each macro. If a macro is not assigned a hot key, you can run the macro using Macros→Play (or the ^MP classic command).
- If you want to assign a hot key to a macro that doesn't have one, or change the hot key assigned to a macro, use Macros→Rename, or ^ME. Don't use Macros→Record (^MR); it will erase the macro and create a new one.
- WordStar comes with predefined macros for the function key hot keys and several other hot keys. These macros are listed in Appendix D, "Function Keys and Other Macros." You can define new macros for these keys if you want, using Record (^MR) or Edit/Create (^MD).

MACRO FILES AND DIRECTORIES

Each macro is stored as a file, with the name you gave it while recording or editing and the .WSM extension.

CAUTION *When you record, create, or edit a macro and save it, WordStar compiles the macro, adding executable code. Macro files appear to be nondocuments (ASCII files) when you edit them, but they contain other information. You must edit them with Macros→Create/Edit (or ^MD) to get usable macros.*

On most systems, all the macro files are stored in one directory. The default macro directory name is WS\MACROS; you can change it with WSCHANGE (CD2G).

PLAY

***Pull-down:* ALT + U, M, P**
***Classic:* ^MP**

Executes a macro.

You can execute a macro either using a hot key or using this command. For an explanation on using hot keys, see "Macro Hot Keys" in this section.

Follow these steps to play a macro:

1 Choose Utilities→Macros→Play, or use the ^MP classic command.

2 In the Play Macro dialog box, specify the macro to play. You can choose the macro either by name or hot key. When you press the Down Arrow key and scroll through the list of macros, the description for the highlighted macro appears in the dialog box.

3 Press F10 or choose OK.

WordStar executes all the commands and enters all the text specified in the macro. Generally, menus and dialog boxes are not displayed unless you need to enter input.

You can stop the macro by pressing Esc or ^Break. (When a macro is paused for input, you must use ^Break.)

RECORD

***Pull-down:* ALT + U, M, R**
***Classic:* ^MR**

Records a series of keystrokes and stores them as a macro that you can execute.

You can create a macro by choosing Record and then pressing all the keystrokes that you want in the macro. Follow these steps:

1 Choose Utilities→Macros→Record, or use the ^MR classic command.

2 In the Record Macro dialog box, type the name you want to give the macro. If the macro name is already in use, confirm that you want to redefine the macro or give it another name.

3 Specify the hot key that will execute the macro. You can press the Down Arrow key and scroll through the list to see which hot keys are available. If the hot key you specify is already in use, confirm that you want to reassign the hot key or specify another hot key. If you don't want to assign a hot key to the macro, you can skip this step.

4 Type a description of the macro to remind you what it does and help you find it in the list. The description can be up to 48 characters long. If you don't want to give a description for the macro, you can skip this step.

5 Press F10 or choose OK.

6 Enter all the keystrokes you want in the macro. Enter them in the same way you want them played back when you run the macro.

You may need the macro to pause when you run it so that you can enter information such as a filename. At any place you want the macro to pause, press Alt+Hyphen. Then type any number of keystrokes as an example of the input. Press Alt+Hyphen again to end the pause, and continue entering the macro keystrokes. When you execute the macro, it pauses at that point for input and continues after you press Alt+Hyphen. (You can change the Alt+Hyphen keystroke to another keystroke using WSCHANGE (DCDB).)

Note: Mouse actions and clicks are not recorded in macros and do not work while you are recording a macro; you must use the keyboard.

7 When you finish entering the keystrokes, press Alt+= (hold down the Alt key and press =). (You can change the Alt+= keystroke to another keystroke using WSCHANGE (DCDA).)

WordStar records and compiles the macro. If you want to make adjustments to the macro, you can either record it again or edit it using Macros→Edit/Create, or the ^MD classic command.

EDIT/CREATE

Pull-down: ALT + U, M, E
Classic: ^MD

Creates and modifies a macro that contains keystrokes and commands that you enter.

Each macro consists of a series of WordStar commands and functions. You edit a macro using the same commands you use for editing a nondocument file. Follow these steps:

Note: You cannot edit a macro if you already have two editing windows open.

1. Choose Utilities→Macros→Edit/Create, or use the ^MD classic command.

2. In the Edit/Create Macro dialog box, specify the macro name.

   When you press the Down Arrow key and scroll through the list of macros, the description for the highlighted macro appears in the dialog box.

3. If you're creating a new macro, specify the hot key that will execute the macro. You can press the Down Arrow key and scroll through the list to see which hot keys are available. If the hot key you specify is already in use, confirm that you want to reassign the hot key or specify another hot key. If you don't want to assign a hot key to the macro, you can skip this step.

4. Type a description of the macro to remind you of what it does and to help you find it in the list. The description can be up to 48 characters long. If you don't want to give a description for the macro, you can skip this step.

5. Press F10 or choose OK. WordStar opens a full-screen editing window for the macro.

6. Type the text of the macro or make any editing changes. For information about what you can enter, see "The Macro Language" in this section. When you edit a macro, you can use the commands used for editing nondocuments.

   TIP *You may want to periodically use File→Save (^KS) to save your changes while editing.*

7 When you finish, save the macro using File→Save and Close (^KD).

WordStar compiles the macro. If it finds any syntax errors or other errors, they are entered into the macro file. You must edit the macro again to correct the errors.

Errors in a macro file are noted by a message describing the error, followed by the line of the macro that contains the error, with a caret (^) below the line pointing to the location of the error.

TIP The Alt+D hot key is preassigned to a macro that goes to the next error message in a macro file. This is helpful when you edit macros to fix errors.

If you want to copy a block of text from one macro into another, start from the Opening screen rather than the Editing screen. Choose Macros→Edit/Create (or ^MD), and then choose File→Open/Switch (or ^OK). Then copy the block between the windows in the same way you would while editing a document.

SINGLE STEP

***Pull-down:* ALT + U, M, S**
***Classic:* ^MS**

Plays back a macro one step at a time to locate errors.

After you create or edit a macro, you can play it back step by step to identify any problems. This command is helpful when a macro doesn't work exactly as you expect it to. Follow these steps:

1 Choose Utilities→Macros→Single Step, or use the ^MS classic command.

2 In the Single Step dialog box, specify the macro to play. You can specify the macro either by name or by hot key. When you press the Down Arrow key and scroll through the list of macros, the description for the highlighted macro appears in the dialog box.

3 Press F10 or choose OK.

WordStar executes the first command or keystroke in the macro, and pauses.

4 Press the Spacebar to continue to the next command, function, or keystroke. Continue in this manner until the macro finishes. You can stop running the macro by pressing Esc or ^Break. (When a macro is paused for input, you must use ^Break.)

Copy

***Pull-down:* ALT + U, M, C**
***Classic:* ^MO**

Copies a macro.

If you want to create a macro similar to another one, you can copy the first macro and edit the copy. You can also make copies of macros so that several people can use them on their computers. Follow these steps:

1 Choose Utilities→Macros→Copy, or use the ^MO classic command.

2 In the Copy Macro dialog box, specify the macro to copy.

You can specify the macro either by name or by hot key. When you press the Down Arrow key and scroll through the list of macros, the description for the highlighted macro appears in the dialog box.

3 Specify the macro name for the new copy. Type the path with the name if necessary. If the name is already in use, confirm that you want to overwrite the current macro.

4 Press F10 or choose OK to copy the macro.

The copied macro has the same description as the original, but is not assigned a hot key. You can assign a hot key and description when you edit the macro, or you can use Rename Macro (or the ^ME classic command).

DELETE

***Pull-down:* ALT + U, M, D**
***Classic:* ^MY**

Deletes a macro.

Follow these steps to delete a macro:

1 Choose Utilities→Macros→Delete, or use the ^MY classic command.

2 In the Delete Macro dialog box, specify the macro to delete. You can specify the macro either by name or by hot key. When you press the Down Arrow key and scroll through the list of macros, the description for the highlighted macro appears in the dialog box.

3 Press F10 or choose OK to delete the macro, and confirm the deletion.

RENAME

***Pull-down:* ALT + U, M, N**
***Classic:* ^ME**

Renames a macro.

You can use this command to rename a macro, assign a new hot key, or add or edit a description. Follow these steps:

1 Choose Utilities→Macros→Rename, or use the ^ME classic command.

2 In the Rename Macro dialog box, specify the macro to rename. You can specify the macro either by name or by hot key. When you press the Down Arrow key and scroll through the list of macros, the description for the highlighted macro appears in the dialog box.

3 Specify the new macro name, hot key, and/or description. If the name or hot key is already in use, confirm that you want to overwrite the current assignment.

4 Press F10 or choose OK to rename the macro.

THE MACRO LANGUAGE

WordStar macros are written and recorded in a BASIC-like language.

Each macro consists of one or more subroutines. A subroutine is a series of statements. The MAIN subroutine is required, and you can use as many other subroutines as you like.

Each statement is a command or function and, if required, its arguments. A statement can have a modifying clause. Each statement can have a label to identify it.

Commands generally control the flow of execution, by specifying which statement to execute next. Some commands are Call, GoTo, Return, Stop, and PlayMacro.

Functions perform specific tasks. Some functions are Key, SetHelpLevel, Insert, and PauseForInput. The Key function is the most important function because it supplies keystrokes to WordStar as if they were typed at the keyboard.

The macro language is described in detail in the following sections. To see more examples, you can look at the macro files that came with WordStar. Unless you've changed directory names, they are in the \WS\MACROS directory. These macros are for the function keys and several other hot keys, and are listed in Appendix D.

MACRO STRUCTURE

A very simple macro is shown below.

```
Sub MAIN
Key("Good morning!{Enter}")
End Sub
```

This macro types **Good morning!** into a document and starts a new line. The macro has three lines. The first line says that the subroutine MAIN begins here, and the last line ends the subroutine.

The second line in the macro is a statement that consists of a single Key( ) function and its argument, the keystroke string **Good morning!{Enter}**. Arguments for functions are always enclosed in parentheses. Keystroke strings are always enclosed in quotation marks. {Enter} signifies the Enter key.

The following example shows a macro for writing a bulleted list. A complete explanation follows the macro. After the explanation of the example, all the macro functions, commands, and other elements are described in detail.

```
REM     This macro is for creating bulleted lists.

Sub MAIN

AutoRestore(ON) : Insert(ON) : CmdTags(OFF)
SetHelpLevel(4) : HideDots(ON)

Key ("{Ctrl+O}FSHanging indent 1{Enter}")
Call Times12
ListLoop: Key ("   •{Tab}")
          PauseForInput("{Enter}", 'Enter the text.')
            IfException
            NCM: Goto ListDone  'User entered nothing
            End IfException
          Key ("{Enter}")
          GoTo ListLoop
ListDone: Key ("{Ctrl+Q}{Del}")
          Key ("{Ctrl+O}FP")
Stop
End Sub

Sub Times12
Key ("{Ctrl+P}=Times 12{Enter}")
Return
End Sub
```

This macro is composed of two subroutines, MAIN and Times12. MAIN has the instructions for entering the bulleted list, and Times12 changes fonts.

The macro has two comments to remind the user what the macro is doing. They are the REM line at the beginning of the macro and the comment "User entered nothing," which describes the situation in which that line takes effect.

At the beginning of the MAIN subroutine are statements that use the AutoRestore, Insert, CmdTags, SetHelpLevel, and HideDots functions to set those five editor states.

The first Key function selects the Hanging Indent 1 paragraph style. Notice that it contains all the keystrokes to choose this style, using the ^OFS classic command.

Next, the Call command runs the Times12 subroutine, which uses a Key statement to change to the Times 12 font. After the font change, the Return command returns to MAIN. Instead of having this subroutine, you could put the Key statement directly in the MAIN subroutine. However, you might want to use this subroutine often in a macro, and it is convenient to be able to call it at several places.

ListLoop is a label. A label marks a location so that you can go to it from elsewhere in the subroutine to run the subsequent instructions. In this case, the label allows a macro to return to the same location a number of times, creating a "loop." The Key statement types two spaces and then the bullet character and a tab. The PauseForInput function causes the macro to wait while the user enters the text that goes next to the bullet, displaying the message *Enter the text.* After entering the text, the user presses Enter, the input termination character. If the user enters text and presses Enter, the IfException is ignored, and the next Key statement presses Enter to go to the next line. Then the GoTo command moves back to the ListLoop label to repeat the procedure for entering another bullet and text.

The three lines beginning with IfException are an IfException clause. An IfException clause tells the macro what to do if it does not receive the expected user input. The IfException signifies the beginning of the clause and the EndIfException signifies the end. NCM (No Cursor Movement) is the type of exception. The NCM line tells the macro that if the user doesn't enter anything before pressing Enter (after the bullet character is entered), the macro should go to the label InputDone. This clause provides the way to end the bulleted list after the user enters the text for the last bullet.

All the indentation in the macro is optional; it is used to make the macro easy to read. After the ListLoop label, the Key statement could be on the same line or the next line, and all the following lines could begin at the left margin.

At the ListDone label, there are two Key statements. The first deletes everything on the line to the left of the cursor using the ^QDel classic command. This removes the last bullet that was inserted, for which there was no text. The last statement reverts to the paragraph style that was in effect above the bulleted list, using the ^OFP classic command. At this point, the macro is finished, and the user returns to editing.

Note that this macro uses classic commands. You can instead use the pull-down menu Alt key combinations for the same commands.

Also note that you can put several statements on a single line by separating statements with a colon (:). You can put spaces or tabs before and after the colon, but they are not required. For example, the first two lines under Sub MAIN in the example above contain five statements.

Subroutines

The format of a subroutine is

```
Sub name
statements
End Sub
```

A subroutine name must begin with a letter and can contain up to 31 characters. The characters must be letters, digits, and the underline character, (_).

Each macro must include the MAIN subroutine, and it is always executed first. You can use additional subroutines to execute the same groups of statements in several locations within a macro; you can enter a subroutine once and call it each time it's needed.

Each statement in a macro must be within a subroutine. Subroutines are separate; a subroutine can call other subroutines, but one subroutine cannot be within another in the text of a macro. Only the MAIN subroutine is executed automatically; you must call any other subroutine to execute it.

Use the Call command to call the subroutine at each location you want to use it. See "Commands" in this section.

THE KEY( ) FUNCTION

The Key( ) function is the most important component of a macro. It specifies the keystrokes that the macro enters. It is in the following format:

```
Key ("keystroke string")
```

The keystroke string must be enclosed in quotation marks. The string can include text and special keys such as direction keys and WordStar commands. A simple Key command with text only is

```
Key ("Sincerely yours,")
```

This command types the phrase **Sincerely yours,**.

Special keys are generally enclosed in braces, for example, {Enter} and {PgDn}. A complete list of special keys is in Appendix C, "Macro Keystrokes." Braces are also used with key combinations, for example, {Alt+F1} and {Ctrl+P}. The following Key command opens a document named KNOX.DOC:

```
Key("{Alt+F}Dknox.doc{Enter}")
```

The following rules apply to special keys and key combinations:

- Upper- and lowercase letters are treated the same; for example, ALT is the same as Alt.
- The caret symbol (^) can be used for the Ctrl key. Braces are not required when this symbol is used. For example, ^C (not enclosed in braces) is equivalent to {Ctrl+C}.
- You can also enter control characters in a macro by pressing ^P and then the control character. For example, if you press ^P^C (hold down the control key, press P and then C, then release the control key), ^C appears in the macro.
- For key combinations, you can use the Plus (+) sign, a space, or a hyphen (–). For example, {Alt+F7}, {Alt F7}, and {Alt-F7} are equivalent.

- An extended ASCII character can be entered by holding down the Alt key and pressing the appropriate number keys on the numeric keypad. For example, pressing Alt+156 puts the £ symbol in the macro.

- To enter an actual ", { , }, or ^, double the character; type " ", { { , } }, or ^^.

For any dialog box option with a check box, you can enter the commands to move to that option, then enter a plus sign (+) for on, a minus sign (–) for off, or 0 for inherit (in paragraph styles). For example, the following Key( ) function goes to the Alignment and Spacing dialog box using the ^OS classic command and turns hyphenation on:

```
Key("^OSh+")
```

For more information about entering keystrokes, see Appendix C, "Macro Keystrokes."

OTHER FUNCTIONS

You can use several other functions besides the Key( ) function in WordStar macros.

The following two functions allow the macro to pause for user input.

PauseForInput(*"input termination key ", 'message text '*) Pauses macro execution so the user can enter text. The input termination key is the keystroke that causes macro execution to resume. It must be enclosed in quotation marks. The termination key is optional if you don't include a message; if you don't specify a keystroke, Alt+Hyphen is used.

The message text is the message you want displayed while paused for input. It must be enclosed in apostrophes. It is optional; if you don't specify a message, *Paused. Press {ALT+–} to continue* appears. To enter an apostrophe in the message text, double the character; type ' '. If you include a message, the input termination key is required in the PauseForInput statement. If you use a termination key other than the default, be sure to include a message that gives the name of the key; otherwise, the person running the macro won't know what to do.

When you record a macro, Alt+Hyphen is used as the input termination key, unless you change the default key using WSCHANGE (DCDB).

PauseForKey( ) Pauses macro execution so the user can enter a single keystroke (for example, Y or N). As soon as the user presses a key, macro execution resumes.

The remaining functions set editing conditions for playing back the macro. These editing conditions can affect how the commands in the macro work. When you create a macro, it's best to use these functions unless you're sure the editing conditions won't affect macro execution. For example, if a macro is written assuming that insert is on, and it is run in overtype mode, text could be lost. If a macro is run at a different help level than it is written for, different menus and dialog boxes may appear, and some of the commands in the macro may not work correctly.

SetHelpLevel(1/2/3/4) Sets the help level in use for macro execution.

Insert(ON/OFF) Sets Insert mode to on or off for macro execution.

CmdTags(ON/OFF) Sets Command Tags display (or ^OD) on or off for macro execution.

HideDots(ON/OFF) Sets display of dot commands on or off for macro execution.

ColMode(ON/OFF) Sets column block mode on or off for macro execution.

ColReplace(ON/OFF) Sets column replace mode on or off for macro execution.

AutoRestore(ON/OFF) After the macro finishes running, restores the editor conditions that were in effect when the macro started. This function is turned on whenever a macro is recorded.

SingleStep(ON/OFF) Allows you to Single Step through a selected portion of a macro.

When you record a macro, WordStar automatically inserts all these functions, with the current settings, into the macro file. For example, recording a simple macro to type **Good morning!** produces this macro file:

```
Sub MAIN
   SetHelpLevel(4)
   Insert(ON)
   CmdTags(OFF)
   HideDots(ON)
   ColMode(OFF)
   ColReplace(OFF)
   AutoRestore(ON)
   Key("Good morning!{Enter}")
End Sub
```

Commands

You can use the following commands to control the flow of the macro execution.

Call *subroutine* Executes the named subroutine in the current macro. When the subroutine is completed, macro execution returns to the statement after the Call statement, unless the subroutine includes the Stop command.

PlayMacro *'macro'* Executes another macro. The macro name must be enclosed in apostrophes. Include a path with the macro name if necessary. Do not include the .WSM extension. To enter an apostrophe in the macro name or path, double the character; type ' '. When the macro is completed, execution returns to the statement after the PlayMacro statement in the original macro, unless the other macro includes the Stop command.

ChainMacro *'macro'* Executes another macro. The macro name must be enclosed in apostrophes. Include a path with the macro name if necessary. Do not include the .WSM extension. To enter an apostrophe in the macro name or path, double the character; type ' '. When the macro is completed, execution does not return to the original macro unless the ChainMacro command was invoked from a subroutine other than Main. In this case, execution returns to the subroutine that called the subroutine containing the ChainMacro command.

Stop Terminates the macro.

Return Stops the current subroutine and returns to either the calling subroutine or the calling macro. If there is no calling subroutine or macro, the macro is terminated.

GoTo *label* Goes to the named label, executes it, and continues from that location. The label must be within the same subroutine as the GoTo command.

A label is a name that identifies a statement or location in a macro. You can use a label to go to a specific location in the macro, and to repeat a set of instructions in a loop. A label name must begin with a letter and can contain up to 31 characters, which must be letters, digits, and the underline character (_). It must end with a colon (:).

THE IFEXCEPTION CLAUSE

When you run the macro and it enters the keystrokes in the Key function, additional user input may be required before the macro can go on to the next function. For example, if you run a macro with the command to insert the file JOE.DOC, **Key("{Alt+I}Fjoe.doc{Enter}")**, and the JOE.DOC file doesn't exist, a message is displayed that requires user input. This is called an exception condition, and is handled using the IfException clause.

You can follow any Key command with an IfException clause. The IfException clause handles these three types of exception conditions:

- **User acknowledgement required (ACK)** This condition occurs when WordStar issues a message in response to a command issued by the macro, and the user needs to press a key to acknowledge it. For example, after a spelling check WordStar displays the number of words checked, and you must press F10 or choose OK to continue.

- **User clarification required (QRY)** This condition occurs when WordStar displays a dialog box in response to a command issued by the macro, and the user needs to enter information or choose one of several options. For example, if the macro inserts a file and gives a filename that doesn't exist, you need to type the correct filename or close the dialog box. When you record a macro, ACK conditions are included when appropriate.

- **No cursor movement (NCM)** If the macro repeats a cursor movement command such as PgUp or PgDn indefinitely, this condition occurs when the cursor reaches the beginning or end of the file or line. This condition also occurs when no user input is received before the input termination key is pressed after a PauseForInput statement.

The IfException clause has the following format:

```
IfException
    [ACK:
    statements]
    [QRY:
    statements]
    [NCM:
    statements]
    [DEF:
    statements]
End IfException
```

The ACK, QRY, and NCM subclauses handle the three conditions described above, and the DEF statement gives the default exception processing for ACK and QRY conditions. At least one of these subclauses must be defined in each IfException clause. The IfException, End IfException, and each subclause must begin on a new line. You can use spaces or tabs to indent the subclauses if you want.

If one of the three exception conditions occurs when the macro is run, the IfException clause becomes active. If the subclause for the type of exception is defined, WordStar executes the statements in that subclause. Otherwise, if the exception is an ACK or QRY type and the DEF clause is defined, WordStar executes the statements in the DEF subclause.

If no IfException clause is present in an ACK or QRY condition, or if an appropriate subclause is not defined, the macro displays a message and waits for user input for the dialog box or message on the screen. When appropriate input is entered, the macro resumes. If no IfException clause is present in an NCM condition, the condition is ignored and the macro proceeds.

When you record a macro, ACK and QRY (but not NCM) IfException clauses are automatically inserted when exception conditions are encountered.

The following example shows the use of ACK:

```
Sub Main
Key("{Ctrl+Q}A")
Key("wonderful{Tab}"})
Key("excellent{Tab}")
Key("MG")
Key("{F10}")
    IfException
    ACK: Key("{Esc}")
    End IfException
End Sub
```

This macro uses the ^QA classic command. In the Find and Replace dialog box, it instructs WordStar to replace "wonderful" with "excellent" using the G (Entire File/Global) and M (Maintain Case) options. Then it issues the F10 keystroke to close the dialog box and start the Find and Replace operation. During this operation, the macro pauses at each instance of "wonderful" for the user to press Y or N for the replacement. When the operation is complete, the *All replacements complete* message triggers the ACK IfException, which supplies the Esc keystroke to clear the message from the screen and resume editing.

The following example shows the use of QRY:

```
Sub Main
Key("{Ctrl+P}=")
Key("Times 10{Enter}")
    IfException
    QRY: Key("{Esc}")
         Key("Courier 12{Enter}")
    End IfException
End Sub
```

This macro uses the ^P= classic command to choose a font. In the Font dialog box, it chooses Times 10. If the Times 10 font isn't available, the error message triggers the QRY IfException, which supplies the Esc keystroke to remove the message from the screen, then chooses the Courier 12 font.

For an example of using NCM, see "Macro Structure."

TIP *When a macro changes options in a dialog box, use separate Key statements for each option. That way, you can use IfException clauses to* *correct problems with each option (for example, if the macro chooses a font or paragraph style that isn't available in the document).*

COMMENTS

You can enter comments in a macro file in several ways:

- Use REM at the beginning of a line. All text following the REM on that line is comment. The REM can be preceded by tabs and spaces.
- Use an apostrophe (') at the end of a statement line. All text following the apostrophe on that line is comment.
- Start a comment anywhere in the macro with a slash and an asterisk (/*), and end it anywhere with an asterisk and a slash (*/). This type of comment can be as many lines long as you like.

MERGE PRINT COMMANDS

***Pull-down:* ALT + U, P**

Enters commands and variable information in order to print personalized form letters and other documents.

Merge printing is the process of inserting information into a document while it's being printed. You can use merge printing for form letters, mailing labels, invoices, contracts, reports, and any other standard document that requires you to "fill in the blanks."

When you print a document with merge print commands, be sure that the print option Interpret Merge Variables is selected (the default). If you want to print the document with the variable names showing in the text, and ignore the merge print commands, clear this option.

WORKING WITH MERGE PRINTING

To merge print, you need a master document that contains both the text of the document and the instructions for inserting the merged information. The merged information can come from a data file such as a mailing list, or from the keyboard at print time, or you can specify it at the beginning of the document.

APPENDIX **C**

Macro Keystrokes

This appendix describes the keystroke conventions used for the WordStar macro system, and provides a table of all the keystrokes that the macro system recognizes. For more information about macros, see "Macros" in "Utilities Menu."

The Key( ) Function

The Key( ) macro function provides keystrokes to the application. The argument of Key( ) is a keystroke string. For example, "Good Morning!{Enter}" is a valid keystroke string, and to have it typed into WordStar you'd include the statement KEY("Good Morning!{Enter}") in a macro. For guidelines for entering keystrokes in the Key( ) function, see "The Macro Language" in "Macros" in "Editing Screen Utilities Menu."

Types of Keystrokes

A keystroke is defined as the act of either pressing a single key or pressing a number of keys at the same time. Keystrokes generate codes that are interpreted in one of two ways:

- A keystroke can be an alphanumeric character, a punctuation mark, or symbol such as **A**, **;**, or ‡. These are all called characters.

- A keystroke can be a command to perform an action such as delete a character, move the cursor up one line, or start a new line. These keystrokes are called application defined keystrokes.

The Key( ) function provides both types of keystrokes. The first type of keystroke is represented in the keystroke string by simply entering the character. The second type of keystroke is represented by a description as shown in the table. Generally, the keystroke description is the same as the one that appears on the key cap on the US keyboard. It is usually enclosed in braces ({ }). (To enter an actual { or }, you double the character, for example, { { .)

A keyboard may contain more than one key that has the same general meaning. To differentiate between keys with similar meanings, the following conventions are used:

- Characters and descriptions enclosed in square brackets represent keys on the numeric key pad, for example, [Enter]. Some of these keys are only available on enhanced keyboards.

- Descriptions enclosed in parentheses represent the cursor and edit pad keys on enhanced keyboards, for example, (UP) and (Del).

Combining these conventions leads to keystroke descriptions like {Enter}, {[Enter]}, and {(UP)}.

SHIFTED KEY DESCRIPTIONS

Keyboards have three types of keys that can be used with other keys, the Shift key, the Control key (Ctrl), and the Alt key. Some non-US keyboards also have a fourth key, the right Alt key (AltGr), which is used to generate additional alphanumeric symbols. These keys are represented in macros enclosed in braces with the accompanying key, for example, {Ctrl C}, {Alt A}, and {Shift F1}.

For alphabetic characters, punctuation marks, and other symbols, don't indicate the Shift key required to generate the keystroke; just enter the character as you normally would. For example, {Shift a} does not produce A. This is also true for the AltGr shift key, which is used to generate characters, not instructions. For example, on a German keyboard, don't represent / as {AltGr ß}.

KEYSTROKES IN RECORDED MACROS

When a recorded macro is translated into a macro source file, keystrokes from enhanced keyboards are translated into the equivalent keystrokes from unenhanced keyboards. For example, {(UP)} becomes {UP}. The same is true for numbers and symbols on the numeric keypad; for example, {[1]} becomes 1.

If a specific keystroke is required, you can edit the macro and enter the extended or numeric keypad description. When the macro is compiled, the indicated scan code and character combination are used.

Note: When you execute a macro, it provides keystrokes to WordStar in the same general way that the BIOS keyboard i/o service does. WordStar often performs additional processing on a keystroke prior to acting on it. For example, if you press ^K^Q while recording (without taking your finger off of the control key before pressing Q), the macro file shows {CTRL K}{CTRL Q}. During playback, the WordStar editor takes the same action as if the source file contained {CTRL K}Q.

KEYSTROKES NOT IN THE TABLE

The table in this appendix shows the standard keystrokes based on the US keyboard in code page 437. Other keyboards and code pages can produce different keystroke codes. When a recorded macro is translated into a macro source file, all recognizable codes are translated into US keyboard equivalents. These equivalents may be particular keys or extended characters.

To enter a specific keystroke not in the table, enter it in *{xxx,yyy}* format, where *xxx* is the high byte and *yyy* is the low byte of the decimal value of the code.

Any unrecognized keystroke code is translated into this same format when a recording is translated into source code.

THE WORDSTAR MACRO US KEYSTROKE TABLE

The following table shows the valid keystrokes for WordStar macros.

- The first column of the table shows the character or description associated with each keystroke.
- The second column shows the hexadecimal value of the code returned by BIOS Interrupt 16H for each keystroke. This is called the keystroke code. The high byte is usually called the scan code, and the low byte is usually called the ASCII code. For text and punctuation keystrokes, the ASCII code (low byte) is the number associated with the character.
- The third column shows the decimal values of the high and low byte of the keystroke code.
- The fourth column indicates if the particular keystroke is available only on an enhanced keyboard. It also shows keystrokes that are specific to WordStar.

The last part of the table includes all the extended characters (those with decimal codes 1 to 31 and 128 to 255). The characters shown in the table are those displayed on the screen for the IBM symbol set for code page 437. (If you're using a different code page, different characters are displayed on the screen. The characters that are printed depend on the symbol set for the font you use for printing.) You can enter extended characters in several ways. For more information, see "Extended Character" in "Insert Menu."

KEYSTROKE			
DESCRIPTION	HEX	DECIMAL	NOTE
{BACKTAB}	0F 00	015 000	
{BKSP}	0E 08	014 008	
{DEL}	53 00	083 000	
{(DEL)}	53 E0	083 224	Enhanced keyboard
{DOWN}	50 00	080 000	
{(DOWN)}	50 E0	080 224	Enhanced keyboard
{END}	4F 00	079 000	
{(END)}	4F E0	079 224	Enhanced keyboard
{ENTER}	1C 0D	028 013	
{[ENTER]}	E0 0D	224 013	Enhanced keyboard
{ESC}	01 1B	001 027	
{HOME}	47 00	071 000	
{(HOME)}	47 E0	071 224	Enhanced keyboard
{INSERT}	52 00	082 000	
{(INSERT)}	52 E0	082 224	Enhanced keyboard
{LEFT}	4B 00	075 000	
{(LEFT)}	4B E0	075 224	Enhanced keyboard
{PGDN}	51 00	081 000	
{(PGDN)}	51 E0	081 224	Enhanced keyboard
{PGUP}	49 00	073 000	
{(PGUP)}	49 E0	073 224	Enhanced keyboard
{RIGHT}	4D 00	077 000	
{(RIGHT)}	4D E0	077 224	Enhanced keyboard
{TAB}	0F 09	015 009	
{([5])}	4C 00	076 000	Enhanced keyboard
{UP}	48 00	072 000	
{(UP)}	48 E0	072 224	Enhanced keyboard
space	39 20	057 032	
!	02 21	002 033	
“	28 22	040 034	
#	04 23	004 035	
$	05 24	005 036	
%	06 25	006 037	
&	08 26	008 038	
’	28 27	040 039	
(	0A 28	010 040	
)	0B 29	011 041	
*	09 2A	009 042	
{[*]}	37 2A	055 042	

KEYSTROKE DESCRIPTION	HEX	DECIMAL	NOTE
+	0D 2B	013 043	
{[+]}	4E 2B	078 043	
,	33 2C	051 044	
-	0C 2D	012 045	
{[-]}	4A 2D	074 045	
.	34 2E	052 046	
{[.]}	53 2E	083 046	
/	35 2F	053 047	
{[/]}	E0 2F	224 047	Enhanced keyboard
:	27 3A	039 058	
;	27 3B	039 059	
<	33 3C	051 060	
=	0D 3D	013 061	
>	34 3E	052 062	
?	35 3F	053 063	
@	03 40	003 064	
[	1A 5B	026 091	
\	2B 5C	043 092	
]	1B 5D	027 093	
^	07 5E	007 094	
_	0C 5F	012 095	
‘	29 60	041 096	
{	1A 7B	026 123	
|	2B 7C	043 124	
}	1B 7D	027 125	
~	29 7E	041 126	
n/a	00 7F	000 127	
0	0B 30	011 048	
{[0]}	52 30	082 048	
1	02 31	002 049	
{[1]}	4F 31	079 049	
2	03 32	003 050	
{[2]}	50 32	080 050	
3	04 33	004 051	
{[3]}	51 33	081 051	
4	05 34	005 052	
{[4]}	4B 34	075 052	
5	06 35	006 053	
{[5]}	4C 35	076 053	

KEYSTROKE			
DESCRIPTION	HEX	DECIMAL	NOTE
6	07 36	007 054	
{[6]}	4D 36	077 054	
7	08 37	008 055	
{[7]}	47 37	071 055	
8	09 38	009 056	
{[8]}	48 38	072 056	
9	0A 39	010 057	
{[9]}	49 39	073 057	
A	1E 41	030 065	
B	30 42	048 066	
C	2E 43	046 067	
D	20 44	032 068	
E	12 45	018 069	
F	21 46	033 070	
G	22 47	034 071	
H	23 48	035 072	
I	17 49	023 073	
J	24 4A	036 074	
K	25 4B	037 075	
L	26 4C	038 076	
M	32 4D	050 077	
N	31 4E	049 078	
O	18 4F	024 079	
P	19 50	025 080	
Q	10 51	016 081	
R	13 52	019 082	
S	1F 53	031 083	
T	14 54	020 084	
U	16 55	022 085	
V	2F 56	047 086	
W	11 57	017 087	
X	2D 58	045 088	
Y	15 59	021 089	
Z	2C 5A	044 090	
a	1E 61	030 097	
b	30 62	048 098	
c	2E 63	046 099	
d	20 64	032 100	
e	12 65	018 101	

KEYSTROKE DESCRIPTION	HEX	DECIMAL	NOTE
f	21 66	033 102	
g	22 67	034 103	
h	23 68	035 104	
i	17 69	023 105	
j	24 6A	036 106	
k	25 6B	037 107	
l	26 6C	038 108	
m	32 6D	050 109	
n	31 6E	049 110	
o	18 6F	024 111	
p	19 70	025 112	
q	10 71	016 113	
r	13 72	019 114	
s	1F 73	031 115	
t	14 74	020 116	
u	16 75	022 117	
v	2F 76	047 118	
w	11 77	017 119	
x	2D 78	045 120	
y	15 79	021 121	
z	2C 7A	044 122	
{CTRL+BKSP}	0E 7F	014 127	
{CTRL+BREAK}	00 00	000 000	
{CTRL+DEL}	93 00	147 000	Enhanced keyboard
{CTRL+(DEL)}	93 E0	147 224	Enhanced keyboard
{CTRL+DOWN}	91 00	145 000	Enhanced keyboard
{CTRL+(DOWN)}	91 E0	145 224	Enhanced keyboard
{CTRL+END}	75 00	117 000	
{CTRL+(END)}	75 E0	117 224	Enhanced keyboard
{CTRL+ENTER}	1C 0A	028 010	
{CTRL+[ENTER]}	E0 0A	224 010	Enhanced keyboard
{CTRL+HOME}	77 00	119 000	
{CTRL+(HOME)}	77 E0	119 224	Enhanced keyboard
{CTRL+INSERT}	92 00	146 000	Enhanced keyboard
{CTRL+(INSERT)}	92 E0	146 224	Enhanced keyboard
{CTRL+LEFT}	73 00	115 000	
{CTRL+(LEFT)}	73 E0	115 224	Enhanced keyboard
{CTRL+PGDN}	76 00	118 000	
{CTRL+(PGDN)}	76 E0	118 224	Enhanced keyboard

KEYSTROKE			
DESCRIPTION	**HEX**	**DECIMAL**	**NOTE**
{CTRL+PGUP}	84 00	132 000	
{CTRL+(PGUP)}	84 E0	132 224	Enhanced keyboard
{CTRL+PRTSC}	72 00	114 000	
{CTRL+RIGHT}	74 00	116 000	
{CTRL+(RIGHT)}	74 E0	116 224	Enhanced keyboard
{CTRL+TAB}	94 00	148 000	Enhanced keyboard
{CTRL+UP}	8D 00	141 000	Enhanced keyboard
{CTRL+(UP)}	8D E0	141 224	Enhanced keyboard
{CTRL+’}	C1 00	193 000	WordStar keystroke
{CTRL+[*]}	96 00	150 000	Enhanced keyboard
{CTRL+[+]}	90 00	144 000	Enhanced keyboard
{CTRL+,}	C3 00	195 000	WordStar keystroke
{CTRL+-}	0C 1F	012 031	
{CTRL+[-]}	8E 00	142 000	Enhanced keyboard
{CTRL+.}	C4 00	196 000	WordStar keystroke
{CTRL+/}	C5 00	197 000	WordStar keystroke
{CTRL+[/]}	95 00	149 000	Enhanced keyboard
{CTRL+;}	C2 00	194 000	WordStar keystroke
{CTRL+[}	1A 1B	026 027	
{CTRL+\}	2B 1C	043 028	
{CTRL+]}	1B 1D	027 029	
{CTRL+‘}	C0 00	192 000	WordStar keystroke
{CTRL+0}	CD 00	205 000	WordStar keystroke
{CTRL+1}	C6 00	198 000	WordStar keystroke
{CTRL+2}	03 00	003 000	
{CTRL+3}	C7 00	199 000	WordStar keystroke
{CTRL+4}	C8 00	200 000	WordStar keystroke
{CTRL+5}	C9 00	201 000	WordStar keystroke
{CTRL+[5]}	8F 00	143 000	Enhanced keyboard
{CTRL+6}	07 1E	007 030	
{CTRL+7}	CA 00	202 000	WordStar keystroke
{CTRL+8}	CB 00	203 000	WordStar keystroke
{CTRL+9}	CC 00	204 000	WordStar keystroke
{CTRL+A}	1E 01	030 001	
{CTRL+B}	30 02	048 002	
{CTRL+C}	2E 03	046 003	
{CTRL+D}	20 04	032 004	
{CTRL+E}	12 05	018 005	
{CTRL+F}	21 06	033 006	

KEYSTROKE DESCRIPTION	HEX	DECIMAL	NOTE
{CTRL+G}	22 07	034 007	
{CTRL+H}	23 08	035 008	
{CTRL+I}	17 09	023 009	
{CTRL+J}	24 0A	036 010	
{CTRL+K}	25 0B	037 011	
{CTRL+L}	26 0C	038 012	
{CTRL+M}	32 0D	050 013	
{CTRL+N}	31 0E	049 014	
{CTRL+O}	18 0F	024 015	
{CTRL+P}	19 10	025 016	
{CTRL+Q}	10 11	016 017	
{CTRL+R}	13 12	019 018	
{CTRL+S}	1F 13	031 019	
{CTRL+T}	14 14	020 020	
{CTRL+U}	16 15	022 021	
{CTRL+V}	2F 16	047 022	
{CTRL+W}	11 17	017 023	
{CTRL+X}	2D 18	045 024	
{CTRL+Y}	15 19	021 025	
{CTRL+Z}	2C 1A	044 026	
{ALT+BKSP}	0E 00	014 000	Enhanced keyboard
{ALT+(DEL)}	A3 00	163 000	Enhanced keyboard
{ALT+(DOWN)}	A0 00	160 000	Enhanced keyboard
{ALT+(END)}	9F 00	159 000	Enhanced keyboard
{ALT+ENTER}	1C 00	028 000	Enhanced keyboard
{ALT+[ENTER]}	A6 00	166 000	Enhanced keyboard
{ALT+ESC}	01 00	001 000	Enhanced keyboard
{ALT+(HOME)}	97 00	151 000	Enhanced keyboard
{ALT+(INSERT)}	A2 00	162 000	Enhanced keyboard
{ALT+(LEFT)}	9B 00	155 000	Enhanced keyboard
{ALT+(PGDN)}	A1 00	161 000	Enhanced keyboard
{ALT+(PGUP)}	99 00	153 000	Enhanced keyboard
{ALT+(RIGHT)}	9D 00	157 000	Enhanced keyboard
{ALT+TAB}	A5 00	165 000	Enhanced keyboard
{ALT+(UP)}	98 00	152 000	Enhanced keyboard
{ALT+’}	28 00	040 000	Enhanced keyboard
{ALT+[*]}	37 00	055 000	Enhanced keyboard
{ALT+[+]}	4E 00	078 000	Enhanced keyboard
{ALT+,}	33 00	051 000	Enhanced keyboard

KEYSTROKE			
DESCRIPTION	HEX	DECIMAL	NOTE
{ALT+-}	82 00	130 000	
{ALT+[-]}	4A 00	074 000	Enhanced keyboard
{ALT+.}	34 00	052 000	Enhanced keyboard
{ALT+/}	35 00	053 000	Enhanced keyboard
{ALT+[/]}	A4 00	164 000	Enhanced keyboard
{ALT+;}	27 00	039 000	Enhanced keyboard
{ALT+=}	83 00	131 000	
{ALT+[}	1A 00	026 000	Enhanced keyboard
{ALT+\}	2B 00	043 000	Enhanced keyboard
{ALT+]}	1B 00	027 000	Enhanced keyboard
{ALT+‘}	29 00	041 000	Enhanced keyboard
{ALT+0}	81 00	129 000	
{ALT+1}	78 00	120 000	
{ALT+2}	79 00	121 000	
{ALT+3}	7A 00	122 000	
{ALT+4}	7B 00	123 000	
{ALT+5}	7C 00	124 000	
{ALT+6}	7D 00	125 000	
{ALT+7}	7E 00	126 000	
{ALT+8}	7F 00	127 000	
{ALT+9}	80 00	128 000	
{ALT+A}	1E 00	030 000	
{ALT+B}	30 00	048 000	
{ALT+C}	2E 00	046 000	
{ALT+D}	20 00	032 000	
{ALT+E}	12 00	018 000	
{ALT+F}	21 00	033 000	
{ALT+G}	22 00	034 000	
{ALT+H}	23 00	035 000	
{ALT+I}	17 00	023 000	
{ALT+J}	24 00	036 000	
{ALT+K}	25 00	037 000	
{ALT+L}	26 00	038 000	
{ALT+M}	32 00	050 000	
{ALT+N}	31 00	049 000	
{ALT+O}	18 00	024 000	
{ALT+P}	19 00	025 000	
{ALT+Q}	10 00	016 000	
{ALT+R}	13 00	019 000	

KEYSTROKE DESCRIPTION	HEX	DECIMAL	NOTE
{ALT+S}	1F 00	031 000	
{ALT+T}	14 00	020 000	
{ALT+U}	16 00	022 000	
{ALT+V}	2F 00	047 000	
{ALT+W}	11 00	017 000	
{ALT+X}	2D 00	045 000	
{ALT+Y}	15 00	021 000	
{ALT+Z}	2C 00	044 000	
{F1}	3B 00	059 000	
{F2}	3C 00	060 000	
{F3}	3D 00	061 000	
{F4}	3E 00	062 000	
{F5}	3F 00	063 000	
{F6}	40 00	064 000	
{F7}	41 00	065 000	
{F8}	42 00	066 000	
{F9}	43 00	067 000	
{F10}	44 00	068 000	
{F11}	85 00	133 000	Enhanced keyboard
{F12}	86 00	134 000	Enhanced keyboard
{SHIFT+F1}	54 00	084 000	
{SHIFT+F2}	55 00	085 000	
{SHIFT+F3}	56 00	086 000	
{SHIFT+F4}	57 00	087 000	
{SHIFT+F5}	58 00	088 000	
{SHIFT+F6}	59 00	089 000	
{SHIFT+F7}	5A 00	090 000	
{SHIFT+F8}	5B 00	091 000	
{SHIFT+F9}	5C 00	092 000	
{SHIFT+F10}	5D 00	093 000	
{SHIFT+F11}	87 00	135 000	Enhanced keyboard
{SHIFT+F12}	88 00	136 000	Enhanced keyboard
{CTRL+F1}	5E 00	094 000	
{CTRL+F2}	5F 00	095 000	
{CTRL+F3}	60 00	096 000	
{CTRL+F4}	61 00	097 000	
{CTRL+F5}	62 00	098 000	
{CTRL+F6}	63 00	099 000	
{CTRL+F7}	64 00	100 000	

KEYSTROKE			
DESCRIPTION	HEX	DECIMAL	NOTE
{CTRL+F8}	65 00	101 000	
{CTRL+F9}	66 00	102 000	
{CTRL+F10}	67 00	103 000	
{CTRL+F11}	89 00	137 000	Enhanced keyboard
{CTRL+F12}	8A 00	138 000	Enhanced keyboard
{ALT+F1}	68 00	104 000	
{ALT+F2}	69 00	105 000	
{ALT+F3}	6A 00	106 000	
{ALT+F4}	6B 00	107 000	
{ALT+F5}	6C 00	108 000	
{ALT+F6}	6D 00	109 000	
[ALT+F7}	6E 00	110 000	
{ALT+F8}	6F 00	111 000	
{ALT+F9}	70 00	112 000	
{ALT+F10}	71 00	113 000	
{ALT+F11}	8B 00	139 000	Enhanced keyboard
{ALT+F12}	8C 00	140 000	Enhanced keyboard

EXTENDED CHARACTERS

CHARACTER	HEX	DEC
☺	00 01	000 001
☻	00 02	000 002
♥	00 03	000 003
♦	00 04	000 004
♣	00 05	000 005
♠	00 06	000 006
•	00 07	000 007
◘	00 08	000 008
○	00 09	000 009
◙	00 0A	000 010
♂	00 0B	000 011
♀	00 0C	000 012
♪	00 0D	000 013
♫	00 0E	000 014
☼	00 0F	000 015
►	00 10	000 016
◄	00 11	000 017
↕	00 12	000 018
‼	00 13	000 019

CHARACTER	HEX	DECIMAL	NOTE
¶	00 14	000 020	
§	00 15	000 021	
▬	00 16	000 022	
↨	00 17	000 023	
↑	00 18	000 024	
↓	00 19	000 025	
→	00 1A	000 026	
←	00 1B	000 027	
∟	00 1C	000 028	
↔	00 1D	000 029	
▲	00 1E	000 030	
▼	00 1F	000 031	
Ç	00 80	000 128	
ü	00 81	000 129	
é	00 82	000 130	
â	00 83	000 131	
ä	00 84	000 132	
à	00 85	000 133	
å	00 86	000 134	
ç	00 87	000 135	
ê	00 88	000 136	
ë	00 89	000 137	
è	00 8A	000 138	
ï	00 8B	000 139	
î	00 8C	000 140	
ì	00 8D	000 141	
Ä	00 8E	000 142	
Å	00 8F	000 143	
É	00 90	000 144	
æ	00 91	000 145	
Æ	00 92	000 146	
ô	00 93	000 147	
ö	00 94	000 148	
ò	00 95	000 149	
û	00 96	000 150	
ù	00 97	000 151	
ÿ	00 98	000 152	
Ö	00 99	000 153	
Ü	00 9A	000 154	

CHARACTER	HEX	DECIMAL	NOTE
¢	00 9B	000 155	
£	00 9C	000 156	
¥	00 9D	000 157	
₧	00 9E	000 158	
ƒ	00 9F	000 159	
á	00 A0	000 160	
í	00 A1	000 161	
ó	00 A2	000 162	
ú	00 A3	000 163	
ñ	00 A4	000 164	
Ñ	00 A5	000 165	
ª	00 A6	000 166	
º	00 A7	000 167	
¿	00 A8	000 168	
⌐	00 A9	000 169	
¬	00 AA	000 170	
½	00 AB	000 171	
¼	00 AC	000 172	
¡	00 AD	000 173	
«	00 AE	000 174	
»	00 AF	000 175	
	00 B0	000 176	
▒	00 B1	000 177	
▓	00 B2	000 178	
│	00 B3	000 179	
┤	00 B4	000 180	
╡	00 B5	000 181	
╢	00 B6	000 182	
╖	00 B7	000 183	
╕	00 B8	000 184	
╣	00 B9	000 185	
║	00 BA	000 186	
╗	00 BB	000 187	
╝	00 BC	000 188	
╜	00 BD	000 189	
╛	00 BE	000 190	
┐	00 BF	000 191	
└	00 C0	000 192	
┴	00 C1	000 193	

CHARACTER	HEX	DECIMAL	NOTE
┬	00 C2	000 194	
├	00 C3	000 195	
─	00 C4	000 196	
┼	00 C5	000 197	
╞	00 C6	000 198	
╟	00 C7	000 199	
╚	00 C8	000 200	
╔	00 C9	000 201	
╩	00 CA	000 202	
╦	00 CB	000 203	
╠	00 CC	000 204	
═	00 CD	000 205	
╬	00 CE	000 206	
╧	00 CF	000 207	
╨	00 D0	000 208	
╤	00 D1	000 209	
╥	00 D2	000 210	
╙	00 D3	000 211	
╘	00 D4	000 212	
╒	00 D5	000 213	
╓	00 D6	000 214	
╫	00 D7	000 215	
╪	00 D8	000 216	
┘	00 D9	000 217	
┌	00 DA	000 218	
█	00 DB	000 219	
▄	00 DC	000 220	
▌	00 DD	000 221	
▐	00 DE	000 222	
▀	00 DF	000 223	
α	00 E0	000 224	
β	00 E1	000 225	
Γ	00 E2	000 226	
π	00 E3	000 227	
Σ	00 E4	000 228	
σ	00 E5	000 229	
μ	00 E6	000 230	
τ	00 E7	000 231	
Φ	00 E8	000 232	

CHARACTER	HEX	DECIMAL	NOTE
Θ	00 E9	000 233	
Ω	00 EA	000 234	
δ	00 EB	000 235	
∞	00 EC	000 236	
φ	00 ED	000 237	
ε	00 EE	000 238	
∩	00 EF	000 239	
≡	00 F0	000 240	
±	00 F1	000 241	
≥	00 F2	000 242	
≤	00 F3	000 243	
⌠	00 F4	000 244	
⌡	00 F5	000 245	
÷	00 F6	000 246	
≈	00 F7	000 247	
°	00 F8	000 248	
•	00 F9	000 249	
·	00 FA	000 250	
√	00 FB	000 251	
η	00 FC	000 252	
^{2}	00 FD	000 253	
■	00 FE	000 254	
	00 FF	000 255	

APPENDIX **D**

FUNCTION KEYS AND OTHER MACROS

You can use the function keys, which are usually labeled **F1**, **F2**, **F3**, and so on, to execute the most frequently used WordStar commands with a single keystroke.

Each function key is set to issue four different commands, depending on whether you use the function key by itself or while holding down one of these keys: Shift, Ctrl, or Alt.

The commands issued by the function keys are preprogrammed macros created with the WordStar macro commands. You can change the function key assignments by choosing Utilities→Macros and then Record, Edit/Create, or Rename, or using the classic command ^MR, ^MD, or ^ME. For information about working with macros, see "Macros" in "Utilities."

All the function key assignments are shown in the following table.

In addition to the function keys, several other preprogrammed macros are supplied with WordStar. These keys are also listed in the table. Several other macros that are not included in the table are also preprogrammed as examples.

FUNCTION KEY	CLASSIC COMMAND	PULL-DOWN COMMAND OR FUNCTION
F1		
F1	F1	Help →Help (this key cannot be reprogrammed)
Shift+F1	^OD	View→Command Tags
Ctrl+F1	^QF	Edit→Find
Alt+F1	│	Inserts box-drawing character
F2		
F2	^U	Edit→Undo
Shift+F2	^OC	Layout→Center Line
Ctrl+F2	^QA	Edit→Find and Replace
Alt+F2	─	Inserts box-drawing character
F3		
F3	^PS	Style→Underline
Shift+F3	^QR^QL	Utilities→Spelling Check Global
Ctrl+F3	^L	Edit→Next Find
Alt+F3	┌	Inserts box-drawing character
F4		
F4	^PB	Style→Bold
Shift+F4	^QN	Utilities→Spelling Check Other→Word
Ctrl+F4	^QI	Edit→Go to Page
Alt+F4	┐	Inserts box-drawing character
F5		
F5	^Y	Edit→Delete→Line
Shift+F5	^KY	Edit→Delete→Block
Ctrl+F5	^QS.LM^N	Inserts **.lm** command on its own line
Alt+F5	└	Inserts box-drawing character
F6		
F6	^T	Edit→Delete→Word
Shift+F6	^KH	View→Block Highlighting
Ctrl+F6	^QS.RM^N	Inserts **.rm** command on its own line
Alt+F6	┘	Inserts box-drawing character
F7		
F7	^B^QP	Utilities→Reformat→Paragraph, then Edit →Go to Other→Previous Position
Shift+F7	^KV	Edit→Move→Block
Ctrl+F7	^QS.PM6↵	Inserts **.pm6** command and a carriage return
Alt+F7	┬	Inserts box-drawing character

FUNCTION KEY	**CLASSIC COMMAND**	**PULL-DOWN COMMAND OR FUNCTION**
F8		
F8	^QS^OO	Inserts the **.rr** command with current ruler line settings above the current line
Shift+F8	^KC	Edit→Copy→Block
Ctrl+F8	^QS.PA↵	Insert→Page Break
Alt+F8	┴	Inserts box-drawing character
F9		
F9	^KS	File→Save
Shift+F9	^KB	Edit→Mark Block Beginning
Ctrl+F9	^QS	Moves the cursor to beginning of the line
Alt+F9	├	Inserts box-drawing character
F10		
F10	^KD	File→Save and Close (this key cannot be reprogrammed)
Shift+F10	^KK	Edit→Mark Block End
Ctrl+F10	^QD	Moves the cursor to the end of the line
Alt+F10	┤	Inserts box-drawing character
Alt+0	^OFS	Style→Select Paragraph Style
Alt+1	^OP	View→Preview
Alt+D	n/a	Find next compiler error for editing macros
Alt+M	n/a	Enters a memo boilerplate
Alt+T	n/a	Enters a To Do list boilerplate
Alt+B	n/a	Bulleted List
Alt+N	n/a	Numbered list