WordPerfect Terminal Definition Program

WordPerfect Terminal Definition Program

Table of Contents

Introduction

This WordPerfect Terminal Definition Program guide is designed to help you edit and create terminal definitions using wpterm (the WordPerfect Terminal Definition program).

WordPerfect Corporation developers use wpterm to produce the collection of ready-made terminal definitions included with your WordPerfect package. We have furnished a copy of wpterm to make it possible for you to modify these terminal definitions or create definitions of your own. Using wpterm, you can tailor definitions to the exact needs of nearly all terminals on your system (you may not be able to create terminal definitions for some little-known or older terminals).

Instructions in this guide are written using the same conventions followed in the *WordPerfect Reference Manual*. To become familiar with these conventions, read *Using the Manual* in the *Getting Started* section of the reference manual.

WARNING: The WordPerfect Terminal Program (wpterm) is not for the casual user. People intending to edit or create definitions with wpterm should be comfortable with advanced terminal concepts. We suggest that you make a copy of the definition you intend to edit, and then make your changes to the copy. This will preserve the integrity of the original definition.

Whether you are working on an existing definition or creating a new one, the steps for editing a definition are the same. The wpterm program makes it possible for you to do the following:

- Define terminal screen attributes (e.g., underline, bold, blink, and reverse video).
- Map WordPerfect functions to the function keys of a terminal.
- Define the character sets and extended characters displayed by the terminal and map these to the character sets used by WordPerfect.
- Define terminal-specific hardware specifications (e.g., the size of the screen, cursor motion on the screen, text insertion and deletion, and scrolling).
- Define values to initialize the terminal for WordPerfect and to restore the terminal to its original state after exiting WordPerfect.

Starting wpterm

Use wpterm (the terminal definition program) as you would use any utility. Before beginning, make sure that the following conditions exist:

- wpterm is in your path (you'll find it in the WordPerfect shared executable directory).
- wpterm has the appropriate read and execute privileges.
- The terminal file (also called a .trs file) that you want to edit is in your current directory.

NOTE: wpterm is not for the casual user. If you intend to edit or create definitions with wpterm, you should be comfortable with advanced terminal concepts. We suggest that you make a copy of the definition you intend to edit, and then make your changes to the copy. This will preserve the integrity of the original definition.

To start wpterm from the shell prompt,

1 Enter **wpterm**.

A list of currently defined terminal definitions in the current working directory are displayed (if any). From this main menu, you can add/create, delete, rename, convert, mark/unmark, search for and look at/edit terminal definitions.

2 Use the arrow keys to highlight the desired definition name.

3 Select Look/Edit (7) to edit the definition.

The Look/Edit menu appears.

The Look/Edit Menu
From the Look/Edit menu you will work with eight areas of the terminal definition program listed below:

1 **E**xtended Characters
2 **A**ttributes
3 **G**raphics
4 Terminal **C**ontrol Strings
5 **S**creen Controls
6 **F**ormat Codes
7 **T**emplate Flag
8 **W**ordPerfect Functions

A chapter is devoted in this booklet to each of the eight areas listed above. As you are working through a terminal definition, you should use the chapters as resources to guide you through the menus that appear when you select any of the eight Look/Edit menu items.

Exiting wpterm

To exit wpterm, press **Exit**, answer the prompt **y** if you have made any changes and would like to save the current definition, and then press **y** to return to the system prompt.

or

Press **n** to exit wpterm without saving the current changes.

Extended Characters

Within the Extended Characters menu, you will be performing three simple tasks:

- Setting the special conditions (called "parameters") by which your terminal will move in and out of the available character sets,

- Filling in the enter/exit sequences your terminal needs to enter and exit the available character sets, and

- Determining which of the character tables available for use in WordPerfect will actually be used by your terminal.

You are not limited to only those characters that your terminal can display. If you wish, you may map characters not supported by your terminal for use with WordPerfect. You can embed a WordPerfect document with characters that cannot be displayed on the current terminal. This might be useful if you were writing a WordPerfect document with the intent of eventually retrieving it on a terminal that could display characters that may not be displayable on the current terminal.

You may also be limited by the capabilities of your printer as to what kinds of characters you can print.

When you select Extended Characters from the Look/Edit menu, three new menu items appear at the bottom of your screen: Parameters (1), Enter/Exit Sequences (2), and Character Tables (3). These three menu items are explored further below.

Parameters (1)
When you select Parameters, a list of questions appears on-screen. Each question provides a default response, which you may either accept or change. Enter your responses by typing the number of the question you wish to answer, and then typing **y** or **n** to indicate Yes or No. You should answer each question with the response that best describes your terminal's requirements.

The questions and a short description for each follow:

Return to the Default Character Set before changing character sets?
The answer to this question should probably be No. Most terminals do not require that you return to the default character set before changing into another set. The majority of terminals let you move directly from one character set to another.

You should answer Yes to this question if your terminal *does* require you to return to the default setting while switching from one character set to another. For example, if your terminal displays odd characters, or it remains in alternate character sets instead of returning to the default, you should answer yes to this question.

Use the exit sequence for character set (a) to return to the default character set?
Answering Yes to this question sets up a handy shortcut. It is likely that the exit sequence you will use to move out of character set (a) is the same exit sequence needed to move out of each of your other character sets. If you answer Yes to the above question, you need only provide the exit sequence for character set (a). WordPerfect will automatically plug in that exit sequence as the exit sequence for each of your other character sets as well.

Answer No if you prefer to provide a unique exit sequence for each of your character sets.

Remain in character set as long as possible?
Answering Yes to this question will tell WordPerfect to keep the terminal from moving between character sets when it is not necessary.

Take, for example, the Thesaurus feature (which appears on screen with an assortment of boxes drawn with line draw characters). The line draw characters needed to form the Thesaurus screen nearly always come from the same character set. If the above question is answered No, then WordPerfect will invoke the enter and then exit sequence of the character set for *each* of the line draw characters in the Thesaurus screen. Responding Yes to this question

significantly reduces the amount of work the terminal must do. In addition, the resultant screen is drawn much faster.

You would answer No to this question only if any of your character sets are mapped to be active for only one character at a time. Some terminals allow you to invoke a new character set with a "just one character" sequence (e.g., (27)H), in which only one character is output from the character set invoked, and then the terminal returns to the default set. If you intend to use such a "just one character" sequence, you should answer No to the above question.

Does switching between Reg/Long row mode affect internal character sets?
Presently, this flag is not used. In future updates of WordPerfect 5.0, however, this flag will be used to tell WordPerfect that when the switch from regular row mode (e.g., 24 lines) to long row mode (e.g., 42 lines) is made, the internal character sets may be affected. WordPerfect would then perform certain compensating operations. For now, this option should be left at its default value, which is No.

Enter/Exit Sequences (2)
The Enter/Exit Sequences screen is where you should fill in or edit the enter and exit sequences for any character sets (a through z) that the terminal will be using. You will find the enter sequences that correspond to the character sets listed in your terminal user's manual. (Exit sequences will usually not be found in the manual because it is generally understood that the exit sequence needed to move out of any character set is the same as the enter sequence needed to move back into the default character set.)

NOTE: The default character set is *not* character set (a). It is generally understood that character set (a) is the first of the alternative character sets that are available in addition to the default set. Character set (b) is the second alternative set, (c) is the third, and so on. The default character set is defined in the string to exit character set (a).

NOTE: Not all terminals have character sets in addition to the standard ASCII set. Some have only a rudimentary graphics set (with single-line drawing characters), while

others allow downloadable character sets. WordPerfect will compensate if all the enter and exit sequence entries are left blank. However, if your terminal has character set capabilities, you should use them to get a better graphics display.

Character Tables (3)
Filling in the character tables is the most time-consuming terminal definition task. While WordPerfect 4.2 had only the IBM PC (ASCII) character set available, WordPerfect 5.0 has expanded the available character sets to 12 proprietary sets that feature symbols as diverse as the Cyrillic letters and Hebrew characters.

Whether these 12 character sets can be displayed on your terminal depends upon its capabilities. Thus, the first step to filling in the Character Tables is to find out exactly what characters your terminal is able to display.

1 Refer to your terminal user's manual to find and review the tables of character sets that your terminal is able to display.

NOTE: A specific terminal may come in several different models with varying capabilities. Your user's manual will often contain information about the "deluxe" version and will mention only in passing that certain features may not be available in a given hardware configuration or revision. Character sets can also fall into the category of "options." You must be aware that just because a manual shows a series of character set tables, this does *not* ensure the availability of these sets. You must ultimately test the information contained in your manual to be sure.

Once you are familiar with what characters your terminal can display, your next step is to find out exactly what characters WordPerfect has available.

2 Refer to *WordPerfect Characters* in the *Appendix* of your *WordPerfect Reference* manual. Review the tables.

Now that you know what characters WordPerfect and your terminal can perform, you need to determine which of the available characters you want to use.

Begin making your choices from your terminal's tables, since WordPerfect cannot display any character that is not supported by your terminal. When you find a character in your terminal's tables that you want to include, move to the WordPerfect tables to find out which WordPerfect character set the character falls within. (Most characters that your terminal supports can be found in one of the 12 WordPerfect character sets.)

NOTE: You will *not* need to define any of the ASCII characters. The ASCII character set is predefined within WordPerfect. You need only define those characters that fall in character sets other than ASCII.

For example, suppose you want to use the "happy face" character on your terminal. You know this character is supported by your terminal because you find the "happy face" in one of the character tables in your terminal's manual. Moving to the WordPerfect tables in the *Appendix* of the WordPerfect reference manual, you look for the "happy face" character and find it under Character Set 5, the Iconic Symbols. You are now sure that you can display this character on your terminal, since you've found it both in your terminal's tables and in the WordPerfect tables. You are ready to define the character in the Character Tables screen in your terminal definition.

From within the Extended Characters screen,

3 Type **5** to select Character Set 5, the Iconic Symbols set.

Once inside a character set list, the cursor keys allow you to scroll through the entries. You can also go directly to a certain character by using the Number Search feature and typing the appropriate decimal number corresponding to the character. For example, to move immediately to the "happy face" entry,

4 Select **N**umber Search (1) and enter **007** (007 is the decimal value for the "happy face" character).

Once the cursor is resting on the character you wish to edit,

5 Select **E**dit To Sequence (2) to move to the slot where

you can edit or fill in the string sent to the terminal to display the character.

The cursor rests on the first character of the string, if the string exists.

6 Enter or edit the string or escape sequence that will tell the terminal to display the character.

For example, to display the "happy face" (which might have a decimal value of (33) and corresponds to the !), we would enter (b)! as the string to send to the terminal.

Remember to begin each string with the bracketed variable which corresponds to the character set to which the character belongs. In the above example, the character set is (b).

Keep the following rules in mind when entering escape sequences in any of the screens in wpterm:

- Any value equal to or lower than decimal 32 (non-printable characters) should be phrased in decimal form in brackets (e.g., (23), (07), etc.).

- Any value 33 through 126 (printable characters) can be entered as ASCII characters from the keyboard. For example, the decimal value (109) corresponds to the "m" (lowercase M) on the keyboard, so when entering a sequence that calls for decimal value (109), you may simply type the "m" from the keyboard (e.g., (27)m). In fact, if you type in (109) (i.e., (27)(109)), wpterm will automatically translate it to its printable character and display (27)m.

- Any value equal to or higher than 127 (non-printable characters) should be phrased in decimal form in brackets (e.g., (129), etc.).

You should now be familiar with mapping character sets with WordPerfect. Follow the steps numbered above for each of the characters beyond the ASCII set that you wish to use with your terminal. But first read the paragraphs below, which briefly list a few characters which are

especially important for an attractive WordPerfect appearance.

Characters You Should Map

The following characters help to give WordPerfect its on-screen look; therefore, we suggest that you map them.

Dark Triangle Pointing Up (▲)
Character Set 6, Decimal 029

Used in the split screen. Will use the "^" if this entry is left blank.

Dark Triangle Pointing Down (▼)
Character Set 6, Decimal 030

Used in the split screen. Will use the "v" if this entry is left blank.

Filled Degree Sign (•)
Character Set 6, Decimal 031

Used in Thesaurus to mark headwords. Will use "*" if this entry is left blank.

Line Draw Characters ┐, └, ┴, ┬, etc.
Character Set 3, Decimal 008-047

Used in Thesaurus, Graphics, and Line Draw. Will use +, |, and - if these entries are left blank.

Attributes

Attributes refer to the characteristics a terminal can assign to a character on-screen. In addition to normal display, a standard terminal might be capable of displaying four attributes: Bold, Underline, Blinking, and Reverse Video. Some terminals can display more attributes than the standard four (e.g., Shadow Characters, Double Underlined Characters, etc.), while some terminals can display fewer.

Within the Attributes screen, you will be performing two simple tasks:

- Answering questions which help WordPerfect understand how your terminal works with attributes.

- Filling in the attribute sequences that signal attributes to your terminal (i.e., the escape sequence that turns Bold on, the sequence that turns Underline off, etc.).

NOTE: If you enter unnecessary strings into any part of wpterm, you may needlessly delay WordPerfect processes.

When you select Attributes from the Look/Edit menu, two new menu items appear at the bottom of the screen: Parameters (1) and Attribute Sequences (2). These two menu items are explored further below.

Parameters (1)

The Parameters screen asks seven questions, with the first five to be answered Yes or No. Your responses will help WordPerfect identify how your terminal handles attributes. To answer these questions you must know a few things about the type of terminal you have. Each question provides a default response, which you may either accept or change. Enter your responses by typing the letter of the question you wish to answer, and then typing **y** or **n** to indicate Yes or No.

An explanation for each of the seven questions is found below:

Are the attributes additive?

A terminal has additive attributes if it is capable of adding attributes one at a time. This means that if, for example, you are typing with bold characters and then send the string to turn on underline, the terminal will keep the bold attribute and "add" the underline attribute. The terminal now displays underlined and bolded characters. If you then send the string to turn on blinking characters, the terminal "adds" the blinking attribute and will display underlined, bolded, and blinking characters. With each added attribute, the attributes that have previously been set remain in affect. If this describes your terminal, answer Yes to this question in wpterm.

A non-additive terminal does not allow you to add attributes in the way described above. With a non-additive terminal, if you are typing with bold characters and then send the string to turn on underline, the bold attribute will be turned off and replaced with the underline attribute. The subsequent characters typed will be underlined until another attribute is assigned, at which point underlining is turned off and replaced with the new attribute, and so on. If this describes your terminal, answer No to this question in wpterm.

Are the attributes subtractive?
A terminal has subtractive attributes if it is capable of subtracting attributes one at a time. This means that if, for example, you are typing with bolded, underlined, and blinking characters and then send the string to turn off underline, the terminal will keep the bold and blinking attributes and "subtract" the underline attribute. The terminal now displays only bolded and blinking characters. With each subtracted attribute, the attributes that have already been set remain in affect. If this describes your terminal, answer Yes to this question in wpterm.

A non-subtractive terminal does not allow you to subtract attributes in the way described above. With a non-subtractive terminal, if you are typing with bolded and underlined characters and then send the string to turn off underline, the bold attribute will also be turned off along with the underline attribute. If this describes your terminal, answer No to this question in wpterm.

You should know that if your terminal is not additive, you

can be sure it is also not subtractive. (If you cannot "add" an attribute, it is impossible to "subtract" it.)

Your terminal manual may provide clues that will help you decide whether your terminal is subtractive. Review the list of sequences in the part of the terminal manual that lists on and off escape sequences for attributes. If you see escape sequences that turn off or cancel each attribute (i.e., Cancel Bold, Underlining Off), you can be almost sure that your terminal is subtractive.

If your terminal is not additive or subtractive, WordPerfect will perform certain compensating functions to simulate an additive terminal for smoother WordPerfect performance.

Is this a line (field) attribute terminal?
With a line (field) attribute terminal, once you set an attribute it is put into effect to the end of the line. With a line attribute terminal, when you turn on an attribute (e.g., underline) the attribute extends to the far right margin, as shown below:

Thank you for choosing our Equity Stock Fund.

The attribute is automatically turned off with the Carriage Return at the end of the line.

If this describes your terminal, you have a line attribute terminal.

If you answer Yes, WordPerfect will turn on all applicable attributes before the character is output to the screen and then turn off all attributes after the character is output. At this time, the effect (especially in the case of Underline) is for the attribute to extend to the right margin of the screen after every character. While this is messy and somewhat slow, it is workable (and the only way to maintain the integrity of the document on the screen).

Is this a stored (embedded) attribute terminal?
Sometimes called an "embedded" or "non-hidden" attribute terminal, a stored attribute terminal is one where the attribute code invisibly takes up a space in the line of text on-screen. For instance, a line of text on a stored attribute

terminal might look something like this:

You'll be **surprised** at how fast your money compounds.

There appears to be an extra space before "surprised," but this is actually the space occupied by the bold attribute that falls before the word "surprised."

If this describes your terminal, you have a stored attribute terminal and should answer Yes to this question. However, because WordPerfect Corporation will not support stored attribute terminals at this time, answering Yes to this question will tell WordPerfect to block the use of any and all attributes while WordPerfect is running.

Is this a single attribute terminal?
The feature that would make it possible to run WordPerfect on a single attribute terminal has not yet been implemented. You should leave this entry at its default response, which is No.

Default Foreground Color Number (0 based)
This feature allows you to set the default screen foreground color for all the WordPerfect users on the system.

The number you enter should correspond to the foreground/background color strings (1-16) in the "Attribute Sequences" portion of the attribute menu. For example, if you enter an escape sequence in the "Background Color 9" string to turn on a blue background (and you want blue to be your default background), you would enter (8) (one less than the actual string number).

Default Background Color Number (0 based)
This feature allows you to set the default screen background color for all the WordPerfect users on the system. (Refer to the previous paragraph for an explanation of which string to enter for a specific color.)

Attribute Sequences (2)
When you select Sequences, a screen appears which contains entry slots for the sequences (both On and Off) of all the single attributes and all their possible combinations. It is likely that you will not need to enter a sequence for every

entry in the Sequences screen, but will need to enter only a few, as will be explained.

Determining which entries to fill in depends upon how you answered the questions in the Parameters screen, explained at the beginning of this chapter. Based on how you answered those questions, you can probably put your terminal into one of three categories:

- Additive/Subtractive
- Additive/Not Subtractive
- Not Additive/Not Subtractive

Remember, there is no such thing as a terminal that is Not Additive/Subtractive.

If you have a line attribute terminal (whether you answered Yes or No in the Parameters screen, explained above), your terminal belongs in the Not Additive/Not Subtractive category.

If you have a stored attribute terminal, disregard the rest of this chapter. WordPerfect will block all attribute sequences automatically.

Once you have determined which category describes your terminal, skip to the applicable subheading below and read about the attribute sequences needed for your terminal.

Additive/Subtractive
This type of terminal is the easiest to map. Since it is capable of both adding and subtracting single attributes in any combination, you must provide escape sequences only for the following:

- Single attributes On
- Single attributes Off

Example: If your terminal is capable of 3 attributes (bold, underline, and reverse video), you must fill in only the sequences that turn each single attribute on and off.

Thus, for this example, you would provide strings for the following entries:

Bold On
Bold Off
Underline On
Underline Off
Reverse On
Reverse Off

It is not necessary to fill in sequences for any combination of your terminal's attributes (i.e., Bold and Underline On, Bold, Underline, and Reverse Off, etc.) In fact, if you enter unnecessary strings into any part of wpterm, you may needlessly delay WordPerfect processes.

Additive/Not Subtractive
This terminal can add attributes singly, but it cannot subtract them singly. On such a terminal, each time you turn off an attribute, all attributes that are also on at the time will be turned off as well (e.g., if, while typing with bolded, underlined characters you were to turn off bold, underlining would be turned off as well).

WordPerfect, however, will simulate subtractive functions by doing the following: when it receives the string to turn off any one attribute, it will turn off all current attributes and then turn back on, or reconstruct, the attributes that are still current.

For an Additive/Not Subtractive terminal, you must only provide strings for the following:

- Single attributes On

- An all Off value.

Example: If your terminal is capable of three attributes (bold, underline, and blinking), you must fill in only the sequences that turn on each single attribute (e.g., Bold On, Underline On, and Blinking On).

Because this is not a subtractive terminal, you must also have an All Off string. WordPerfect is set up to look to the

Bold Off entry for the string that will turn off all attributes. Therefore, you must be sure that the string to turn off all attributes is plugged into the Bold Off entry.

Thus, for this example, you would provide strings for the following entries:

Bold On
Bold Off (filled in with the All Off string)
Underline On
Blink On

It is not necessary to fill in sequences for any combination of your terminal's attributes (i.e., Bold and Underline On, Bold, Underline, and Reverse Off, etc.) In fact, if you enter unnecessary strings into any part of wpterm, you may needlessly delay WordPerfect processes.

Not Additive/Not Subtractive
Because it cannot add or subtract single attributes, this type of terminal is the most time-consuming to map. However, WordPerfect can simulate additive and subtractive functions by doing the following: when it receives the string to turn on an attribute (e.g., bold), it looks for the unique string that turns on the attribute. If you then send the string to turn on another attribute (e.g., underline), WordPerfect looks for the unique string that turns on both attributes (e.g., Bold and Underline On). If you then add another attribute, WordPerfect looks for the unique string that turns on all three attributes, and so on.

WordPerfect "subtracts" attributes in a similar way. If it receives the string to turn off any one attribute, it will turn off all current attributes and then look for the string that will turn back on, or reconstruct, the attributes that are still current.

Since this type of terminal cannot add or subtract single attributes, you must provide a unique string for every possible combination of attributes, both on and off that you wish to display. Specifically, you must provide strings for the following:

- All single attributes On,
- A Normal On value

- All attribute combinations On
- All attribute combinations Off

For example, if your terminal is capable of three attributes (bold, underline, and blinking), you must fill in the sequences that turn on each single attribute (Bold On, Underline On, Blinking On).

Because this is not a subtractive terminal, you do not need to enter off values for any of the single attributes. In fact, if you enter unnecessary strings into any part of wpterm, you may needlessly delay WordPerfect processes.

Instead, you will provide a Normal On value (any single attribute turned off is really just normal mode turned on) and plug it into the Bold Off entry. WordPerfect is set up to look to the Bold Off entry for the string that turns Normal On.

You must also provide on and off strings for every combination of attributes.

So, for this example of a terminal with bold, underline, and blinking attributes, we must provide strings for the following entries:

Bold On
Bold Off (filled in with the Normal On string)
Underline On
Blink On

Bold and Underline On
Bold and Underline Off
Bold and Blink On
Bold and Blink Off
Underline and Blink On
Underline and Blink Off

Bold, Underline, and Blink On
Bold, Underline, and Blink Off

You should now have a good understanding of how to fill in the attribute sequences for your terminal. Keep the following in mind as you do so.

- Always begin each escape sequence with the escape value in decimal form (enclosed in brackets) to signal to WordPerfect that the following is an escape sequence (i.e., (27)(22m, (27)(7m, etc.).

- Any value equal to or lower than decimal 32 (non-printable characters) should be phrased in decimal form (in brackets, e.g., (23), (07), etc.).

- Any value 33 through 126 (printable characters) can be entered as ASCII characters from the keyboard (without brackets). For example, the decimal value (109) corresponds to the "m" (lowercase M) on the keyboard, so when entering a sequence that calls for decimal value (109) you may simply type the "m" from the keyboard (e.g., (27)m). In fact, if you type in (109) (i.e., (27)(109)), wpterm will automatically translate it to its printable character and display (27)m.

- Any value equal to or higher than 127 (non-printable characters) should be phrased in decimal form in brackets (e.g., (129), etc.).

When you select Graphics (3) from the Look/Edit menu of wpterm, a list of graphics resource files (.grs files) appears on-screen. These files are the graphics drivers shipped with WordPerfect to help your terminal display graphical representations of WordPerfect documents.

Using the options at the bottom of the screen (**I**nclude, **E**xclude, Select **D**efault, **N**ame Search), you can select from this list of graphics drivers those drivers to be used on your terminal. You can also choose or change the driver that will be the default for the WordPerfect users on your system. Please note, however, that the capabilities of your terminal (or terminal emulator) will determine which graphics types, if any, can be selected and used in WordPerfect.

Text

You will notice that the "text" graphics driver has already been selected for you, indicated by either an asterisk (*) or a "D" for default. This is the basic graphics driver that is supported on all terminals. If your terminal could support no other graphics driver, WordPerfect would still be able to adequately approximate the WordPerfect screens with "text" characters.

Linedraw

A large number of terminals also support linedraw characters. If your terminal supports them, and if you intend to map (or have already mapped) the linedraw characters in the Extended Characters screen of wpterm (explained in *Extended Characters*, a section of this guide), you should select the Linedraw graphics driver from the graphics driver list to include it for use on your terminal.

Because linedraw is a character-based character set, the Linedraw graphics driver would be a good one to choose as the default driver (see *Choosing a Default* below).

Other Graphics Drivers

Some terminals feature downloadable graphics capabilities, mosaic character sets, and other graphics capacities. In order to be able to use these capabilities within

WordPerfect, you must select the corresponding graphics driver from the graphics driver list. (If you are working with a terminal definition written by WordPerfect Corporation, it is likely that the correct drivers for your terminal have already been selected.)

When trying to decide which drivers apply to your terminal, keep in mind that if your terminal's name is not a prefix of the graphic driver's name, it probably isn't a valid selection (except "text" or "linedraw").

WARNING: It is important to not select graphics drivers that do not apply to the capabilities of your terminal. If you do, unpredictable results may occur.

To select a driver,

1 Move the cursor to the driver name you wish to select.

2 Select **I**nclude (1) to include the driver.

An asterisk appears next to your selection.

NOTE: Although there are entries for cga, ega, and vga, these drivers were created specifically for 386 console definitions (this does not include 386 consoles using PC terminal emulators). They should not be selected unless you are actually using a 386 console.

Choosing a Default

The default graphics driver that you select will determine the graphics mode for WordPerfect when it is first started by the users on your system.

We suggest that you choose Linedraw as your default graphics driver, if your terminal supports it.

To specify a default driver,

1 Move the cursor to the driver name that will be your default.

2 Choose Select **D**efault (3).

A "D" for default appears next to your selection. If another

driver had previously been the default, this driver now appears with an asterisk to indicate that it is selected, but is not the default.

Individual users may change the default graphics mode for their own use of WordPerfect within the WordPerfect Setup menu.

Terminal Control Strings

When you select **T**erminal Control Strings (4) from the Look/Edit menu, you see a list of entries which hold strings that control certain functions of the terminal. You may write or edit these strings to suit the needs of your terminal.

Hint

The series of sequences that you use as terminal control strings may become too long and cumbersome for convenient use within this screen. For example, the initialization string for a terminal might contain hundreds of bytes of escape sequences that initialize the terminal for WordPerfect.

Rather than write out the entire string in wpterm, you can take advantage of one or both of the shortcuts explained below. They will save byte space within wpterm, and will make editing the strings more convenient when editing is needed.

(E)*executable file*(E)
If you wish to have an executable file as a part of a terminal control string, you may choose to place the full pathname of the executable file between two "(E)" gates and embed it within your string. This will signal WordPerfect to execute the file when it comes across it within the terminal control string.

For example, this shortcut would be useful in the string to initialize the terminal in which you wish to execute a small program that maps the keyboard.

(X)*text file*(X)
If you wish to make a text file part of a terminal control string, you may do so by placing in the string the full pathname of the text file between two "(X)" gates. This will signal WordPerfect to write the text file to the terminal when it comes across it within the terminal control string.

For example, this might be useful in the string to create downloadable user character sets.

The Terminal

Control Strings

The following are brief explanations and suggestions regarding what kinds of escape sequences are needed in each of the terminal control screen entries.

String to Initialize the Terminal
This string will be executed at the very outset each time WordPerfect is begun. The escape sequences you include in this string depend upon what needs to be done to your terminal to get it ready to run WordPerfect.

As a general rule, you should include in this string the sequences that will set the terminal to its original defaults. Specifically, you should include the following sequences when writing or editing the string:

- The sequence that disables auto-wrap (sometimes called "auto-line wrap," "line truncate mode," or "auto new line").

- The sequence that sets all attributes to normal.

- The sequence that maps the keyboard keys back to their default state.

If you are editing a definition written by WordPerfect Corporation, you can be confident that all of the above bulleted items have been covered in the definition.

String to Reset the Terminal
You will want to include in this string the escape sequences that put the terminal back into its pre-WordPerfect initialization state. You may also want to have a sequence that sets the attributes back to normal (in the instance that a user exits WordPerfect with an attribute on), a sequence that clears the screen, and a sequence that replaces the cursor at the top left-hand corner of the screen.

String for Quick Terminal Initialization
This string is essentially the same as the string to initialize the terminal; however, it is used when exiting to shell. We suggest you duplicate the string to initialize the terminal.

String for Quick Terminal Reset
This string is similar to the string to reset the terminal. This

string is used to exit shell. We recommend that you duplicate the string to reset the terminal.

Pass-Through Printing On
This string allows a terminal to take data from the main port and instead of displaying it on the terminal, send it to the print port to enable pass-through printing.

This is necessary only if you intend to support pass-through printing.

Pass-Through Printing Off
This string turns off pass-through printing.

String Sent to Terminal to Place Into 80 Column Mode
This escape sequence places the terminal into 80 column mode. This is usually the default setting.

String Sent to Terminal to Place Into Wide Column Mode
The wide column mode string places your terminal into "compressed" or "wide column" mode. The usual display is 132.

This string is necessary only if your terminal provides the capibility to initiate wide column mode, and if you want to support it.

String Sent to Terminal to Place Into Regular Row Mode
This string sets the number of rows displayed on your terminal. The default setting is usually 24.

String Sent to Terminal to Place Into Long Row Mode
This string allows you to display more rows on your screen. It sets your terminal into long row mode if your terminal has the capability and you want to support this feature. For example, the Wyse 60 terminal has the ability to display 42 lines in "long row mode."

String Sent to Terminal to Place Into Keypad Application Mode
This string provides your terminal with the ability to display the corresponding escape sequence instead of the numeric display when a key on the keypad is pressed.

The default setting is nearly always keypad numeric mode.

String Sent to Terminal to Place Into Keypad Numeric Mode
This sequence returns the terminal to the default of displaying numbers when a numeric key is pressed.

NOTE: This string is *not* the same as Numlock On.

Is this a Windows Terminal Definition?
This field was created for inhouse use by WordPerfect Corporation. Answer **no** to this question unless you are modifying a WordPerfect Corporation windows definition.

Is this a Scan Code Terminal Definition?
This field was also created for inhouse use by WordPerfect Corporation. You should answer No to this question unless the terminal you are using has a PC mode or a scan-code mode (and your hardware is setup for 8-bit data transmission).

If you answer Yes to this question, you must then include the appropriate sequences to enter and exit PC mode in your terminal initialization/reset strings.

By doing this, additional keyboard functionality is gained. For example, your keystrokes to execute WordPerfect functions will be the same as if you were using a WordPerfect Corporation PC product (i.e., Alt-F4 for Block, Ctrl-Backspace for Delete Word, etc.).

NOTE: If you answer this question Yes, the PC-mode of a terminal may not use the same methods to accomplish these functions as the default "terminal" mode. Thus, you may be requried to edit certain control strings/attribute strings, etc.

Also, if you answer Yes, you must provide the name of the scan-code table file which will be used to interpret your keystrokes. This file should reside in the WordPerfect shared data directory. We suggest that you select wyscan.sys, a file already provided.

Name of External Scan Code Table File
This field was created for use at WordPerfect Corporation.

You should answer No to this question unless the terminal you are using has a PC mode or a scan-code mode (and your hardware is set up for 8-bit data transmission).

If you answer Yes, see the explanation for the above field for related information.

The **S**creen Controls (5) option lets you enter the necessary escape sequences which allow WordPerfect to execute screen control functions. As you work through each example you will note that some sequences are optional. If your terminal does not have wide column or long row mode, you will be directed to leave that entry blank.

It is important to note that the examples given for each category are for ANSI (American National Standards Institute) terminals. If your terminal does not conform to these standards, the escape sequence examples may not work. If this is the case, consult your terminal manual for the proper sequence.

Also, note that the escape sequence which contains variables in the cursor position sequence must follow WordPerfect conventions (e.g., row and column as (R);(C), and Top and Bottom as (T);(B)).

Number of Columns on the Terminal Screen
Enter the number of columns that you would like displayed on the terminal screen. The default value is 80 (select this value unless your terminal manual directs otherwise).

Number of Lines on the Terminal Screen
This entry establishes the number of rows your terminal can display. If your terminal can display 25 lines, enter that number. If not, choose the default of 24.

Number of Columns on the Terminal Screen (Wide Column Mode)
If your terminal has the capability of displaying in wide column mode, enter the wide column number here (i.e., 132). If not, duplicate the number in the number of columns on the terminal screen entry.

Number of Lines on the Terminal Screen (Long Row Mode)
If your terminal has the ability to display more than 24 lines (i.e., 42, 60), enter that number here. If not, duplicate the number in the number of lines on the terminal screen field.

String to Clear to the End of the Screen
Enter the escape sequence found in the terminal manual which clears from the cursor to the end of the screen. An ANSI example you might use is (27)(J.

String to Clear to the End of the Line
Enter the escape sequence found in the terminal manual which clears from the cursor to the end of the line. An ANSI example you might use is (27)(K.

String to Position the Cursor (80 Column Mode)
This string gives WordPerfect information to place the cursor in the correct screen position. An ANSI example you might use is (27)(Pl;Pch (where Pl is the row position and Pc is the column position). WordPerfect requires you to use (R) (C) conventions as variables for row and column. An example of this sequence is (27)((R);(C).

String to Position the Cursor (Wide Column Mode)
This string is optional and is only necessary if your terminal has wide column mode. If it does, duplicate the above entry unless your terminal manual calls for an alternate string for wide column mode.

String to Move the Cursor Up
This string moves the cursor up one row at a time. An ANSI example is (27)(A.

String to Move the Cursor Down
This string moves the cursor down one row at a time. An ANSI example is (27)(B.

String to Move the Cursor Right
This string moves the cursor right one column at a time. An ANSI example is (27)(C.

String to Move the Cursor Left
This string moves the cursor left one column at a time. An ANSI example is (27)(D.

String to Turn the Cursor On
This is an optional string which turns on the cursor after it has been turned off (see the following entry). An ANSI example is (27)(?25h.

String to Turn the Cursor Off
This optional string turns the cursor off. This provides for a cleaner output on some WordPerfect screens. An ANSI example is (27)(?25l.

String for a Carriage-Return
This string returns the carriage at the end of a line. This ANSI string is (13).

String for a Carriage-Return/Line-Feed
The string for a carriage-return/line-feed is entered in this field. The ANSI string is (13)(10).

String for a Hardware Insert
If your terminal has the insert text function, enter the terminal string here. An ANSI example is (27)(4H. If your terminal does not have this function, leave this field blank and WordPerfect will simulate this feature using other means.

String for a Hardware Overwrite
If your terminal has the text overwrite function, enter the terminal string here. An ANSI example is (27)(4l. If your terminal does not have this function, leave this field blank and WordPerfect will simulate this feature using other means.

String to Sound the Bell
This string allows users to select a "beep" to sound on their terminal during some situations while using WordPerfect. To provide this option, enter the appropriate ANSI string which is (7). If your terminal supports a visible bell, you could enter that sequence here as well.

String to Set the Scroll Region to a Window
If your terminal has the capability to set a scroll region, enter the string here. An ANSI example is (27)(Pt;Pbr (where Pt is the top row of the scrolling region and Pb is the bottom). WordPerfect requires you to use the (T) and (B) conventions as the variables for top and bottom. An example of WordPerfect convention for this sequence is (27)((T);(B)r. If your terminal does not have this function, leave this field blank.

String to Scroll Forward (Move Text Up)
If you entered a sequence in the previous field, enter the

string necessary to move the text up within the scroll region. An ANSI example is (27)D. If you did not define the previous string, leave this entry blank.

String to Scroll Backward (Move Text Down)
If you have information in the String to Set the Scroll Region to a Window field, enter the string necessary to move the text down within the scroll region. An ANSI example is (27)M. If you did not define the previous two strings, leave this entry blank.

String to Delete a Line (Move Text Together)
The information in this string is used by WordPerfect to emulate scrolling. For WordPerfect to perform scrolling, it is essential that you provide this string. An example is (27)(M.

String to Insert a Line (Move Text Apart)
The information in this string is used by WordPerfect to emulate scrolling. For WordPerfect to perform scrolling, it is essential that you provide this string. An example is (27)(L.

Can Autowrap be Disabled on this Terminal?
If your terminal has the capacity to disable autowrap using an escape sequence, enter **y** here. If so, you must include the sequence to disable autowrap in the String to Initalize the Terminal option in the Terminal Control Strings menu (see *Terminal Control Strings* in this guide).

WPCorp Strings

The following fields are intended for inhouse use only at WordPerfect Corporation. They must be left blank.

String to Enable Mouse Menus (Windows Version Specific)

String to Disable Mouse Menus (Windows Version Specific)

String to Turn Scroll Bar On (Windows Version Specific)

String to Turn Scroll Bar Off (Windows Version Specific)

String to Update Scroll Bar (Windows Version Specific)

String to Notify Tool of Status (Windows Version Specific)

The **F**ormat Codes (6) section allows you to determine the format of the data assigned to cursor position and scroll region variables when WordPerfect formats a document.

You must answer selections A, C, E, G, and I in this section with one of three choices: Binary, ASCII with no pad characters, or 2-Byte Data General addressing padded with zeros. You must answer selections B, D, F, H, and J with an offset value (explained under question B).

The choices for selections A, C, E, G, and I are defined as follows:

Options

A. Binary

Binary is used whenever the data which will appear is in the form of a single byte or character in decimal format (without a delimiter). Binary characters can have a value of 0-127; however, the first 31 characters are non-printable. This allows a choice from 32-127, or a limit of 95 characters (values from 0-95).

When you enter values, WordPerfect subtracts the appropriate offset which could be any number from 0 to ~40. For example, if your terminal uses a binary value for cursor positioning and requires an offset of 32 to be added, the sequence to position to the top left corner (0,0) will most likely be (27)=(32)(32). If the specified offset is 0, WordPerfect will send (27)=(0)(0) to the terminal.

To determine whether your terminal is binary, compare examples in your terminal manual with the following examples.

(27)0B(R)(C)
(27)=(R)(C)
(27)X(R)(C)

If (R) and (C) are not separated by a delimiter, then select Binary in questions A, C, E, G, and I.

B. ASCII with no pad characters

ASCII characters are represented with a graphical symbol. In the escape sequence, they are represented by one (as in row 1), two (as in column 15), or three bytes (as in column 132), with (R) and (C) always separated by a delimiter.

If the string examples in your manual are listed with a delimiter and share characteristics with any of the following examples, then choose this format.

(27)a(R)R(C)C
(27)(R);(C)H
(27)X(R)(27)Y(C)

C. 2-Byte Data General addressing padded with zeros
This format is unique to Data General terminals which use Write Screen Address. Do not select this option unless this description fits your terminal.

Selections

The following explanations should help you answer each category.

A. Row <R> format (Regular Mode) B
This is the data format of the (R) row value in the string to set regular 80 column mode. Select either A, B, or C depending on the requirements of your terminal. The default is set to B.

B. Offset to add to the Row (Regular Mode) (1)
This is a value which allows WordPerfect to calculate a coordinate for a screen setting.

Your selection here will depend on your how you answered question A. WordPerfect interprets the top row and left column as position 0,0, while most ANSI (American National Standards Institute) terminals refer to that screen position as 1,1, and most binary terminals refer to it as 32,32.

If you selected A (Binary) in selection A, you will most likely enter **32**. If you selected B (ASCII) in question A, enter **1**. If you selected C (DG) in question A, enter **1**. This will allow WordPerfect to deduct the correct spaces to set to the proper 0,0 coordinate.

The default for this entry is 1.

C. Column <C> format (Regular Mode) **B**
This selection refers to the format of the (C) column and should be answered the same as question A. The default is set to B.

D. Offset to add to the Column (Regular Mode) **(1)**
The input here should be the same as question B.

E. Row <R> format (Wide Screen Mode) **B**
This selection deals with your terminal's ability to support wide screen mode. This entry will most likely be the same as question A; however, it will be different if your terminal uses a separate format to position the cursor in wide screen mode. If your terminal does not support wide screen mode, leave this entry blank.

F. Offset to add to the Row (Wide Screen Mode) **(1)**
What you input here will normally be the same as question B; however, it will be different if your terminal uses a separate format to position the cursor in wide screen mode. If your terminal does not support wide screen mode, leave this entry blank.

G. Column <C> format (Wide Screen Mode) **B**
This selection deals with your terminal's ability to support wide screen mode. This entry will most likely be the same as question C; however, it will be different if your terminal uses a separate format to position the cursor in wide screen mode. If your terminal does not support wide screen mode, leave this entry blank.

H. Offset to add to the Column (Wide Screen Mode) **(1)**
The input here should be the same as the entry for question F.

I. Top <T> and Bottom <B> scroll row format **B**
This selection is normally set to an ASCII value. Therefore, if your terminal supports scrolling, you should select the default. Otherwise, leave this entry blank.

J. Offset to add to the scroll rows **(1)**
Select the default of 1 if your terminal supports scrolling; otherwise, leave this entry blank.

Template Flag

The **T**emplate Flag (7) option lets you select the style of the on-screen template which is accessed when you press the Help key twice while working in WordPerfect.

If you want an on-screen template, you should model your keyboard function layout after one of the available styles and then choose the desired option for the template type.

You must first determine which of the templates best fits the needs of your terminal, and then answer **y** to that question. If you answer Yes to any of the options (A-F), the others will automatically default to N.

Once you have selected the style of the template, you must move to **W**ordPerfect Functions (8) option and fill in the function and keystroke locations (see *WordPerfect Functions,* the next section of this guide).

To see an example of the terminal you have selected, return to WordPerfect and press the Help key twice.

Template Descriptions

A. PC-Style Function Layout ("Cancel" on F1, etc.)?
This template is styled after a PC style keyboard. There are 10 function keys with Cancel on F1.

B. Adjusted PC-Style Function Layout ("Escape" on F1, etc.)?
This selection mirrors some WordPerfect 4.2 definitions. You may want to choose this option if you do not want to convert any template functions from a 4.2 version of WordPerfect. This template (printed plastic) is no longer supported by WordPerfect for version 5.0.

C. WP 4.2 VT102-Style Application Mode Keypad Function Layout?
This template lists the function keys on the keypad. If you have a VT100 style keyboard and you are using the keypad for all WordPerfect functions (e.g., keypad 6 for Exit), you may want to type **y** on this option.

D. Horizontal Numeric Key Function Layout?
This template is designed to list 12 function keys instead of

10. This allows you to add Page Down, Page Up, Hard Page, and Go To to function 11, and Soft Hyphen, Macro Pause, and Typeover/Insert to function 12.

This template is intended for use on terminals which do not have many function keys. Therefore, you could use 1-0, -, and = as function keys. However, you could also use regular function keys.

E. WP 5.0 VT220-Style Function Layout ("Cancel" on F7, etc.)?
The design of this template features 10 function keys. It places two middle function keys (Help and Do) on F15 and F16. Also, F1 through F5 are functions which are local to the terminal (e.g., Break).

F. Potpourri-Style Function Layout (i.e., none of the above)?
This option will try to open a WordPerfect 4.2 style template file to see if drawings are available for that terminal. If such screens are available, they will be shown. This template file (wptpl.us) is no longer being updated with additional screens or corrections.

You may also want to choose this option if you do not intend to display a template in the Help screen.

WordPerfect Functions

The **W**ordPerfect Functions (8) option lets you assign the locations of the functions and keystrokes used by both the on screen template and the Help screens within WordPerfect. You can also input sequences sent by keys which relay information from the terminal to WordPerfect.

From the status line you can select Name Search (1), Edit Help Keystrokes (2), Edit Sequence From Terminal (3), and Exit.

Status Line

Name Search (1)
This option allows you to move directly to a desired listing in the information field.

Edit Help Keystrokes (2)
With the **E**dit Help Keystrokes (2) option you can name the key on which a specific function resides. As you recall from working in the previous section of this guide, you selected the style of the template with the **T**emplate Flag (7) option. You will now finish providing the information needed to display the keystroke locations used for that template.

Note that this field accepts any character that is typed. Information is listed in Help screens and on the on-screen template exactly as it is input (watch for spelling errors and other mistakes).

Edit Sequence From Terminal (3)
Using this option, you will enter the sequence for WordPerfect features. If a character is unprintable, it must be entered in brackets.

If you do not know the value of the key to input the sequence, use wpkey to find the information (see the *Wpkey Utility* section in this guide).

Template Key Locations

You will complete three tasks while working in this option. First, you must assign the locations where function lead-in keys will reside on the on-screen template. In the previous section you selected a template style, and now you will

finish setting up the information needed to display keystrokes on the Help screen template.

The following review of WordPerfect conventions will help you understand the keystroke sequences and how you must enter them in this section.

- If only one keystroke is required to execute a function, enter the text for that keystroke (e.g., if you are entering F7 to invoke Exit, type F7).

- If a lead-in key is required to be pressed and held while selecting a WordPerfect function, then separate the key names with a hyphen (e.g., if you are entering Print as holding Shift and then pressing F10, you would enter Shift-F10).

- If a function requires pressing a series of individual keys, then the key names should be separated by commas (e.g., if you are entering Setup as pressing PF1, releasing it and then pressing F5, then you would enter PF1,F5).

To assign the location of the WordPerfect function lead-in keys for the on-screen template,

1 Move to the Level One Keystroke entry using the down arrow or the Name Search option.

2 Choose Edit Help Keystrokes (2) and name the lead-in key used on your terminal to access first level WordPerfect features (i.e., Cancel, →Search, Help, etc.).

Normally this entry would be left blank because on most terminals these features require no lead-in keys to be pressed. However, on number key layouts all level one functions (e.g., Cancel, →Search, Help, etc.) require a lead-in function key (i.e., F1, PF1 etc.). If this is the case, and your level one lead-in key was F1, you would enter **F1,** (notice the required comma indicating to WordPerfect that a keystroke will follow). If your level one lead-in key was PF1, you would enter **PF1,** (again, note the comma).

3 Press Exit.

Next you will edit the Level Two Keystroke entry.

4 Move to the Level Two Keystroke entry.

5 Choose Edit Help Keystrokes (2) and name the lead-in key used on your terminal to access second level WordPerfect features (i.e., Setup, ←Search, Switch, etc.).

This entry could be Shift, PF1, F14, or another second level lead-in key. On a standard PC layout, you would enter **Shift-** (you must include the dash to indicate to WordPerfect that a keystroke will follow). For a VT220, you might enter **PF1,** (comma included). And on number key layouts you might enter **PF2,**, or **F2,** to produce level two functions.

6 Press Exit.

Next you will edit the Level Three Keystroke entry.

7 Move to the Level Three Keystroke entry.

8 Choose Edit Help Keystrokes (2) and assign the lead-in key used on your terminal to access third level WordPerfect features (i.e., Thesaurus, Replace, Reveal Codes, etc.).

This entry could be **Ctrl-**, **PF2,**, **F15,**, or another third level lead-in key. On number key layouts, you might use **F3,** to invoke level three functions.

9 Press Exit.

Next you will edit the Level Four Keystroke entry.

10 Move to the Level Four Keystroke entry.

11 Choose Edit Help Keystrokes (2) and assign the lead-in key used on your terminal to access the fourth level WordPerfect features (i.e., Shell, Spell, Screen, etc.).

This entry could be **Alt-**, **PF3,**, **F16,** or another fourth level lead-in key. On number key layouts, you might use **PF4,** to invoke level four functions.

12 Press Exit.

Function Key Keystrokes

The second task you will complete is to move to the Function Key Keystroke entries (1-12) and assign where each WordPerfect function will reside.

If you select Function Key 1 Keystroke, this setting does not necessarily make the entry F1. It could be F5, or something entirely different (depending on your terminal).

For example, on a DEC VT220 terminal the F1 key is a local function and performs a hold screen. With the first 5 function keys local, the WordPerfect Function Key 1 Keystroke would appear on F6.

To complete the entries for each Function Key Keystroke,

1 Move to a Function Key Keystroke entry (1-12).

2 Choose Edit Help Keystrokes (2) and enter the text of the key used on your terminal for the appropriate block of functions (i.e., the block with Cancel, Setup, Thesaurus, and Shell could reside on the F1, F5, F6, 1, or another key, depending on your terminal). Thus, you would enter **F6** if you have Bold, Center, Tab Align, and Flush Right assigned to that block.

3 Press Exit.

Repeat steps 1-3 above by selecting Edit Help Keystrokes (2) and Edit Sequence From Terminal (3) for the remainder of the Function Key Keystroke entries. If you are using a 10 function keyboard layout, you could leave the entries for function key 11 and function key 12 blank.

Editing Terminal Sequences

The final task you will complete using the **W**ordPerfect Functions (8) option is to edit Help screen entries and provide sequences from the terminal for the remaining WordPerfect features.

By selecting Edit Help Keystrokes (2) you provided information that allows wpterm to label the on-screen template and name keystrokes for Help screens. Now you will continue to enter information in that field, and also input a terminal

sequence by selecting Edit Sequence From Terminal (3) from the status line.

NOTE: If you do not know the value of the key to input the sequence, use wpkey to find the information (see the *Wpkey Utility* section in this guide).

If you create a new terminal definition from scratch, each entry will be blank, with the exception of the Ctrl- letter substitute functions (Ctrl-A will have a (1), Ctrl-B will have a (2), etc. in the sequence entry). If you are editing an existing definition, you will need to update only those entries which have changed.

When entering keystroke sequence information, WordPerfect will not allow you to use the same keystroke for different functions. Also, WordPerfect will not accept a keystroke if it has already been used as a lead-in for another function.

Complete the following steps as necessary for the following features.

1 Move to a WordPerfect function.

2 Choose Edit Help Keystrokes (2) and enter the text you wish to be displayed in the WordPerfect help menus for that function (like Alt-F4 or PF1, or F12, etc.).

3 Press Exit or Return.

4 Select Edit Sequence From Terminal (3) and enter the sequence for the selected entry.

For example, if you are assigning the sequence for Block and wish to use PF3 followed by F10 as your keystrokes, you should enter here the sequences sent by the keys (in this case, (27)OR(27)(21~ where PF3 sends an (27)OR and F10 sends an (27)(21~).

5 Press Exit or Return.

Optional Functions/ Help Strings

The following features are optional, depending on your terminal. If your terminal does not have a particular feature, leave the entries blank.

Column Toggle (Reg/Wide)
If your terminal has the capability for wide column mode, enter the name of the keystroke that will function as a toggle between regular and wide column mode in the first field. Then in the second field, enter the keystroke sequence which performs that function.

Comments about the Terminal
This entry allows you to add comments and a description of the terminal in the Edit Help Keystrokes (2) field. This information is not displayed on the on-screen template or in the Help screens, but can be assessed by users if they select Screen, and then choose Terminal Help (3). You cannot select Edit Sequence From Terminal (3) for this entry.

Information you type here might include special adjustments you made to the keyboard or other information useful to users (WordPerfect Corporation developers use it to keep track of revisions).

Get WP Status (Windows Only)
This field is for use at WordPerfect Corporation only. Leave the two entries blank in this field.

Keypad Toggle (Num/Appl)
If your keyboard can move between number lock and application mode, enter the Help screen and terminal sequence information in the appropriate fields.

Mouse Cursor Positioning
This field is for use at WordPerfect Corporation only. Leave this entry blank.

Null Function
This is a sequence to assign no action to a key. It could be used to deactivate certain keys on the keyboard if you do not want them active. Otherwise, leave the two entries blank in this field.

Resize Window
This field is for use at WordPerfect Corporation only. Leave the two entries blank in this field.

Row Toggle (Reg/Long)

This field is for use at WordPerfect Corporation only. Leave the two entries blank in this field.

Scroll Bar Display Off
This field is for use at WordPerfect Corporation only. Leave the two entries blank in this field.

Scroll Bar Display On
This field is for use at WordPerfect Corporation only. Leave the two entries blank in this field.

Scroll Bar Function
This field is for use at WordPerfect Corporation only. Leave the two entries blank in this field.

Scroll Bar Display OffLine Down
This field is for use at WordPerfect Corporation only. Leave the two entries blank in this field.

Scroll Bar Display Off Line Up
This field is for use at WordPerfect Corporation only. Leave the two entries blank in this field.

Scroll Bar Move
This field is for use at WordPerfect Corporation only. Leave the two entries blank in this field.

Scroll Bar Restore
This field is for use at WordPerfect Corporation only. Leave the two entries blank in this field.

User Defined Key 1-10
These 10 entries allow you to define keys for any remaining keys on your keyboard. Users can select Keyboard Layout in the Setup screen to map where these keys can be used. Leave the two entries blank for each field you decide not to use.

Essential Functions

The remaining entries must be edited for WordPerfect to work correctly. Complete the following steps for the listed features.

1 Move to a WordPerfect function.

2 Choose Edit Help Keystrokes (2) and enter the text you want displayed in the WordPerfect help menus for that function (e.g., **Shift-F7**, **F11,Dash**, etc.).

3 Press Exit or Return.

3 Select Edit Sequence From Terminal (3) and assign the sequence for the selected entry.

4 Press Exit or Return.

If you choose not to list the necessary information in Help screens, you can leave the Edit Help Keystrokes (2) entry blank. However, you must input the correct sequence in the Edit Sequence From Terminal (3) field.

The following table lists the name of the WordPerfect feature, the level of the lead-in key where it normally resides, and the location of the keystroke which usually invokes that feature. This is the ideal keyboard setup, but actual keystrokes you establish may vary with the type of terminal you are using.

Function	**Level**	**Key**
Block	4	F4
Bold	1	F6
Cancel	1	F1
Center	2	F6
Clear Notification Banner	3	5
Common Interface Lead-In	3	8
Compose Key	3	2
Ctrl-A/Z Substitute	3	A-Z/a-z
Cursor Down	1	Down Arrow
Cursor Left	1	Left Arrow
Cursor Right	1	Right Arrow
Cursor Up	1	Up Arrow
Date/Outline	2	F5
Delete Left	1	Backspace
Delete Right	1	Delete
Delete to End of Line	3	End
Delete to End of Page	3	PageDown
Delete Word	3	Backspace
End of Line	1	End
Escape	1	Esc, Esc
Exit/Restart	1	F7

Flush Right	4	F6
Font	3	F8
Footnote	3	F7
Format	2	F8
Go To	3	Home
Graphics	4	F9
Hard Hyphen	1	Dash "-"
Hard Page	3	Return
Hard Return	1	Return
Help	1	F3
Home	1	Home
→Indent	1	F4
→Indent←	2	F4
List Files	1	F5
Macro Commands	3	PageUp
Macro Define	3	F10
Macro Perform	4	F10
Macro Variable 1-10	4	"1-0"
Margin Release	2	Tab
Mark Text	4	F5
Math/Columns	4	F7
Merge Codes	2	F9
Merge R	1	F9
Merge/Sort	3	F9
Move	3	F4
Page Down	1	Page Down
Page Up	1	Page Up
Print	2	F7
Quick Macro A-Z	4	"A-Z/a-z"
Replace	4	F2
Reset Keyboard Mapping	3	"6"
Retrieve	2	F10
Reveal Codes	4	F3
Save	1	F10
Screen	3	F3
Screen Down	3	DownArrow
Screen Up	3	UpArrow
→Search	1	F2
←Search	2	F2
Setup	2	F1
Shell	3	F1
Shell Macro Define	3	"7"
Soft Hyphen	3	Dash "-"
Spell	3	F2

Style	4	F8
Switch	2	F3
Tab Align	3	F6
Tab Function	1	Tab
Text In/Out	3	F5
Thesaurus	4	F1
Typeover/Insert	1	Insert
Underline	1	F8
Word Left	3	LeftArrow
Word Right	3	RightArrow

NOTE: The Ctrl-letter subsitiute functions are used like the Ctrl-letter macros for the PC product. However, in some cases, you cannot use the actual sequences of the Ctrl-letter sequence because of a "clash" with another key or communication limitations. For example, the Return key sends a (13) which is the same value sent by pressing the Ctrl-M key. In this case, you would need to use an alternate level-3 lead-in key for the Ctrl-M substitute function (e.g., **F11,m** might be used).

WordPerfect provides the wpkey utility to supplement the terminal program (wpterm). Wpkey serves two purposes: 1) to display sequence information about the various keys on a keyboard, and 2) to verify the accuracy of a terminal definition. The wpkey utility is also discussed in the *System Administrator's Guide*.

To run wpkey,

1 Enter **wpkey** at the shell prompt.

2 Press the key for which you want to receive information.

wpkey displays information about a key using a hexadecimal, a decimal, and a character (if the character is printable) representation. For example, if you press Return, wpkey displays the following:

val = Oxa (10) Ctrl-J

In this example, the Return key sends the value Ctrl-J (which can be interpreted as Oxa in hexidecimal or as (10) in decimal value). "Ox" is simply an indicator that the value is a hex value, and "()" is an indicator of the WordPerfect decimal notation. In the wpterm program, any keystroke sequence information requiring values which are less than 32 should be entered as a decimal value surrounded by "()" (for example, ^A (Ctrl-A) would be entered as (1)).

3 Once you have seen the displayed value(s), type the plus sign (+) three times in a row to exit the wpkey utility.

-n Option

The -n option which can follow the wpkey command (wpkey -n) provides information similar to the example above. In addition, this option causes the terminal to undergo the same initialization that takes place when you start WordPerfect.

To use the -n option, a valid .trs file must already exist in the WordPerfect data directory, and your WPTERM (or

WPTERM50) environment variable must be set to the corresponding definition name.

Then, if you press Return, wpkey with the -n option displays the following:

```
val = oxd          (13)          Ctrl-M
```

Notice that the Return key has been changed from (10) to (13). This is an effect of the initialization that the terminal undergoes when WordPerfect starts. For this reason, the value (13) is used instead of (10) in the "Hard Return" function in the terminal program. This -n option is especially useful for checking the values sent by any reprogrammed function keys.

-f Option

The -f option, which can follow the wpkey command (wpkey -f), is necessary if you want to verify the accuracy of a terminal definition you have created or edited. The terminal undergoes the same initialization that takes place when you start WordPerfect. This option displays the actual WordPerfect function name instead of a numeric representation of the value(s) of the key (if it exists). Not all keystrokes map into WordPerfect functions--for example, on the PC, Ctrl-Shift-F5 does not invoke a function.

If you press Return, wpkey with the -f option displays the following:

```
Hard Return
```