Documentation File TT10DUSR.DOC

PLEASE NOTE:

This is a WordPerfect®-for-Windows document file, NOT an ASCII text file. It will NOT display or print correctly if you try to view or print it as ASCII text, because it contains many non-standard (non-ASCII) characters, as well as embedded graphics and complex formatting and font-attribute codes.

This document was created in WPWIN 5.2 using MS Windows' "Times New Roman" and "Courier New" TrueType fonts (only). This file should print correctly on any laser or dot-matrix printer if these two conditions are met:

▸ You must print it from WPWIN, *and you must use a Windows printer driver* (not a WordPerfect PRS printer driver, which will not include the TrueType fonts); and

▸ The "Times New Roman" and "Courier New" TrueType fonts must be installed on your system. (These fonts were included in your Windows 3.1 product package, and most users will find them already installed. If they are not already installed on your system, use the Fonts utility in Windows' Control Panel to install them.)

The original TT10DUSR.DOC file was write-protected at exactly 203,164 bytes, with a file date of 12:01 am 07/08/1994.

Presents

Version 1.0-D (Demo Edition)

Supplemental Features for Sci/Tech/Med Typing in WordPerfect® for Windows 5.2

COPYRIGHT NOTICE

TRADEMARK NOTICES

System Requirements

All TechType features are designed to run within WordPerfect for Windows version 5.2 (the "host" application), running under Microsoft Windows version 3.1 and any version of MS-DOS that supports the host application. ***A VGA or higher graphics adapter is required.*** All other hardware requirements and recommendations for TechType are the same as those for its host application.

How To Reach Us— For Tech Support, or Whatever

For technical support, application assistance, or any other questions or comments about TechType 1.0-D, please call or write to us at:

Badger SoftWerks
Phone 608/437-3348
1096 Perry Center Road
Mt. Horeb, WI 53572

TechType 1.0-D User Guide Table of Contents

Front Matter:

[more]

TABLE OF CONTENTS (CONTINUED):

TechType Version 1.0-D (Demo Edition) License Agreement

This License Agreement is by and between you, the trial user of this TechType 1.0-D software, and Badger SoftWerks ("BSW," of Mt. Horeb, WI, USA), its author, manufacturer and owner.

The TechType 1.0-D software package consists of the contents of all ten computer files included in this demonstration-version release, and all copies of those files in your possession or under your control, whether or not the original files or any copies have been modified or renamed. The original file names of these ten files are:

TT10D-IN.WCM
TT10D-01.WCM
TT10D-02.WCM
TT10D-99.WCM
TECHTYPE.INI
TT10DCUA.WWK
TT10DDOS.WWK
TT10DUSR.DOC
TT10DQWK.REF
README.NOW

Your possession or use of this software implies your acceptance of this License Agreement. If you cannot accept or choose to not accept any of these terms and conditions, you must immediately destroy all copies of the software in your possession or under your control.

This software remains and will always be the copyrighted property of BSW. Upon your acceptance of this License Agreement, BSW grants you qualified permission to use the software free of charge for a period of 30 days, beginning on the date you install the software, subject to these terms and conditions:

◆ You may not decompile, disassemble, reverse-engineer, create a derivative work, or otherwise modify the software except automatically, by using a TechType feature that automatically updates or alters the contents of TECHTYPE.INI, or as specifically noted in the TechType 1.0-D User Guide, or at the direct suggestion of BSW Support staff. Note that this prohibition against modifications applies to all ten files listed above, including the User Guide and Quick-Reference Card.

◆ You may not copy this software, or any portion thereof, except:

> ◆ You may use the TechType Installation macro to automatically install (copy) the software to your hard disk from its source diskette or directory.
>
> ◆ You may copy the software for FREE distribution to others, but only in its complete and original form. Note that the Installation macro initializes (modifies) the TECHTYPE.INI file that it copies to your hard disk, so you may not duplicate the INSTALLED copy of TECHTYPE.INI for

any reason.

If you obtained your copy of the software with all ten files bundled into a single compressed file (e.g., downloaded from a BBS), then you may copy and distribute only the same, original, single, unaltered compressed file. If you obtained your copy on a diskette, then any copy you distribute must be created from the original, unaltered files on the original diskette, and must include all ten files on the original diskette.

◆ You acknowledge and agree that:

◆ BSW provides the software as is, without warranty as to performance or fitness for any particular application or purpose.

◆ You and you alone assume the entire risk for all results and outcomes of this product's use. BSW specifically denies any and all liability for any incidental or consequential damages resulting from the use of this software.

◆ BSW also specifically denies any and all liability for any technical or editorial errors, omissions or ambiguities in its product documentation, and for any incidental or consequential damages resulting from the use of its product documentation.

◆ Under no circumstances shall any BSW liability associated with this product extend to any party other than a licensed user.

If any provision of this Agreement is found to be unlawful, void or unenforceable, then that provision shall be severed from this Agreement and will not affect the validity and enforceability of any of the remaining provisions.

This Agreement shall be governed by the laws of the State of Wisconsin.

SPECIAL NOTATIONS USED IN THIS GUIDE

Keyboard Keys: Within this User Guide, the names of most of the keys on your keyboard appear in Initial Caps, like these: Enter, Esc, Ctrl, Shft, PgUp, Del, Ins, F1, F2, F3, etc. For clarity, keys that are known by symbols rather than words appear within square brackets, like these: [/] , [.] , and [,] .

Combination Keystrokes: Most TechType features require that you press two or three keys simultaneously— that is, you press and hold one or two keys, then press and release another key, and then release the first key(s). These "combination keystrokes" appear as strings of two or three individual key names joined by plus symbols ("+"), as in: Ctrl+[/], Alt+[/], Ctrl+ Shft+ O, Ctrl+5, Alt+Shft+F1, etc.

1. Introduction

echType is a collection of six features designed to simplify and streamline the insertion of non-standard characters ("WP Characters") and special notations common to scientific, medical and other technical typing. TechType runs entirely and exclusively within WordPerfect for Windows. Versions 1.0 and 1.0-D (Demo Edition) will run only in WPWIN 5.2, but TechType 1.1 and 1.1-D will support both WPWIN 5.2 and 6.0.

The six features of TechType are:

- *Keyboard-Direct Characters:* Single-keystroke access to 28 common but non-standard characters: an assortment of 16 popular bullet characters, plus §, ¶, †, º, ©, ®, ™, ≤, ≥, — (em dash), – (en dash), and … (ellipsis).

- *TechType Characters:* Quick two-step keyboard shortcuts to 104 characters commonly used in scientific, math, medical, engineering and other technical text: fractions, arrows, Greek chars, math functions/relations, and med-notation symbols.

- *Custom Fractions:* Automates the process of setting pseudo-typographic fractions that are not available as "WP Characters"— e.g., $^3/_{16}$, $^9/_{32}$, etc. (WP-Character fractions consist only of ½, thirds, fourths and eighths.)

- *Super/Subscripter:* Assigns WPWIN's superscript and subscript features to the Alt+Shft+[↑] and Alt+Shft+[↓] keys. Nothing fancy here, but more accessible and much handier than via WPWIN's Font menu.

- *Super/Sub Stacker™:* Automates the creation of "stacked" super/ subscripts (e.g., Σ^{super}_{sub}).

- *ChemType™:* Automatically converts plain-text entries to true chem notations: h2so4 = H_2SO_4, na+cl– = Na^+Cl^-, etc. Interprets sequences of alpha characters to determine upper/lowercase (for element notations), and sets all digits and valences (+/–) to appropriate super- and subscripts.

1.1 How Does TechType Work?

TechType is *not* a separate program (from WPWIN), but rather a combination of an enhanced WPWIN "soft" keyboard definition (a *.WWK file) with two powerful WPWIN macros that function seamlessly within WPWIN, just like any other WPWIN macro.

The information below is provided only for the curious— FYI. You need not read and understand it to be able to use TechType features. (You may skip ahead to section 1.2 if you're not in the mood.)

THE TECHTYPE KEYBOARD

The heart of TechType is the keyboard-definition file (*.WWK) that contains the key assignments for activating TechType features and inserting the Keyboard-Direct Characters. WPWIN permits the creation and editing of customized "soft" keyboards, so that you can assign specific WPWIN features and macros to certain keys and key combinations.

TechType provides two such keyboard-definition files, one cloned from the CUA keyboard (the WPWIN default), and the other from WPWIN's "WP-DOS-compatible" keyboard. Both TechType Keyboards contain the same features, assigned to the same keys, so you may choose whichever layout is more convenient for you— or more familiar to you. Almost all of the new key assignments are simply *added to* the original keyboard layouts, so you don't need to un-learn the default keystrokes if you are already familiar with them. (Section 2.7 below notes three exceptions.) The TechType Quick-Reference Card shows you at a glance which features are assigned to which keys.

Both TechType keyboards are fully customizeable, just like any other WPWIN keyboard definition. You may use WPWIN's Keyboard Editor to move WPWIN and/or TechType features to alternate key combinations if you like, or to add your own macros. (Section 3.1 of this User Guide provides more information about how to tweak a TechType keyboard definition. Section 3.2 explains how to edit a non-TechType keyboard definition so that it will run the TechType features— an alternative approach that may be more convenient for you if you are already using a heavily customized "soft" keyboard.)

THE TECHTYPE MACROS

Four of TechType's six features are performed by WPWIN macros. A macro is a set of instructions— or commands— to a "host" program (in this case, WPWIN) that use the host program's features and capabilities to automate one or more tasks. That is, rather than manually perform a long series of tasks within WPWIN to accomplish some end result, you can use a macro to perform the steps automatically, and much more quickly than you could perform them yourself. For example, if you often use Initial Caps (words whose first letters are capitalized, but whose other letters are lowercased), you could create a macro that would automatically seek out the first letter of the current word, select it, apply the case change to uppercase, unselect the first letter, select the rest of the word, and finally apply a case change to lowercase. In this example, the macro would perform, in less than a second and at a single keystroke, all of the steps that would otherwise require several seconds and several keystrokes.

Although WPCorp provides several handy macros with its WPWIN product, most macros are created by end users to simplify or automate a variety of complex and/or repetitive tasks. The example in the above paragraph is actually a relatively simple macro. Other, more complex macros may contain hundreds or even thousands of steps, with many conditional forks and loops, and with one or more prompts to the user to specify what the macro should do next, or to specify variable information (file names, directory names, etc.).

The two TechType macros are examples of complex macros that can branch in different directions to perform different tasks depending on what you have asked the macros to do. Both TechType macros also "read" the context of the insertion point to automatically determine what they should do. All four of the features performed by these macros are essentially conversion functions: they convert plain-text "expressions" that you have typed on the screen into special, non-standard characters (TechType Characters), or into special notations with superscripts, subscripts, upper/lowercase changes, font-attribute changes, and/or formatting codes (Custom Fractions, Super/Sub Stacker, and ChemType). The action of either TechType macro, then, will depend on the content of the "expression" you have typed.

The main TechType macro (filename TT10D-01.WCM) is used for the TechType Characters, Custom Fractions, and Super/Sub Stacker features. The macro reads your input expression to determine which feature to apply to it. A "TechType expression" for any of these three features must begin with a "key character" that tells the macro which feature to apply. (You need not learn about key characters here, since chapter 2 below provides complete operating instructions for all features.)

The ChemType macro (filename TT10D-02.WCM) is used only for the ChemType feature. It reads the expression, interprets sequences of alpha characters to determine appropriate upper/lowercase (for element notations), and appropriately super/subscripts all digits and valences (+/–), depending on what characters they precede or follow.

1.2 Getting Started

A Very Important Note About TechType Files

Do not Open the TechType, ChemType or Trial Monitor macro files for any reason, ever! These files, named TT10D-01.WCM, TT10D-02.WCM and TT10D-99.WCM, will no longer function if you perform a Save, Save As, or any other operation on them that changes their original DOS file dates.

The macro code for these macros exists only in compiled form, and resides only in the file prefixes, since the original macro code has been stripped from the document portions of the files. WPWIN automatically attempts to recompile any macro if its DOS file date differs from the date it was last compiled, which is stored in the file prefix. WPWIN's attempt to recompile would delete the compiled macro code in the file prefix, but the recompilation would then fail since there are no macro commands to compile in the document portion of the file. *You will destroy the macro functionality of your TechType macro files if you Open them and then Save them,* ***even if you have not entered any changes to the document portion of the file!***

Special Notes About this Demo Edition

The 30-day limit of your free-trial period with this Demo Edition is enforced by a Trial Monitor macro that stores date-related information in your TECHTYPE.INI file, and regularly checks your system date to verify that your trial-period expiration date has not yet passed.

Our security strategies are somewhat complicated, but you don't need to understand them to be able to use TechType. The only thing you need to remember is this: *Leave your TECHTYPE.INI file alone! If you tamper with the Trial Monitor settings in TECHTYPE.INI, you will almost certainly disable the main TechType feature macros.* If, however, you are curious about how our Trial Monitor works, feel free to read on:

Your 30-day expiration date is set by the Installation macro, and is stored in TECHTYPE.INI along with a coded equivalent to that date called the "Run Key." To verify that the stored expiration date has not been changed (in TECHTYPE.INI), TechType daily decodes the Run Key and compares the decoded date to the stored expiration date. If they don't match, or if the expiration date has passed, neither of the two TechType feature macros (TT10D-01.WCM and TT10D-

02.WCM) will run. *TechType will know if you change the expiration date in TECHTYPE.INI, and will not run!*

The Installation and Trial Monitor macros also incorporate safeguards against unscrupulous users who might back-date their system clocks— or advance their clocks prior to installation— to try to defeat the 30-day limit.

We've designed our overall Trial Monitor system to check your expiration date only once a day, since the multi-step process of decoding the Run Key and evaluating the dates would otherwise slow the performance of TechType's primary features to a crawl. The first time in each calendar date that you use a TechType feature that calls either of the two main macros, the main macro will call the Trial Monitor macro, but thereafter (within the same day) the main macros will bypass the Trial Monitor and simply execute the features for which they were primarily designed. This once-a-day approach, however, requires that we also store the Trial Monitor's "Last Run Date" in TECHTYPE.INI, and encode it to thwart unscrupulous users.

Some users are likely to find the Trial Monitor annoying, even with this once-a-day approach, since it will interrupt and delay the performance of the first TechType macro feature you use each day. If you like, you may instead set up your WPWIN program to automatically run the Trial Monitor upon the start of WPWIN, and the TechType macros will thereafter bypass it. To automatically run the Trial Monitor upon WPWIN start-up:

1. In Windows' Program Manager, select your WPWIN program icon and press Alt+Enter to display the Program Item Properties dialog box.

2. In the Command Line box, add a space and this switch:

   ```
   /m-tt10d-99.wcm
   ```

   ... to the end of the existing command text. This switch tells WPWIN to automatically run the Trial Monitor macro (TT10D-99.WCM) immediately upon start-up.

3. Choose the OK button to save the change and clear the dialog box.

TECHTYPE 1.0-D INSTALLATION INSTRUCTIONS

TechType 1.0-D (Demo Edition) will function properly only if it is installed and initialized by the Installation macro included among the program files. This demo edition contains a Trial Monitor function that regularly checks your system date and compares it to your trial-period expiration date, stored in your TECHTYPE.INI file. The Trial Monitor function will not work if TECHTYPE.INI has not been properly initialized, and four of TechType's six features will not work unless the Trial Monitor is working.

For a more complete explanation of TechType's Trial Monitor, refer to the previous section of this User Guide, *Special Notes About this Demo Edition.*

In addition to copying the TechType 1.0-D program files and initializing your TECHTYPE.INI file, the Installation macro will also copy two WPWIN document files— TT10DUSR.DOC (the TechType User Guide), and TT10DQWK.REF (the TechType Quick-Reference Card)— to a WPWIN document directory of your choice. You may then open and print these documents on your own printer. Note that these files were created in WPWIN using only the Times New Roman and Courier New fonts that come with Windows 3.1, so they should print just fine on your own printer as long as: (1) these fonts are already installed in your system, and (2) *you use a Windows printer driver (not a WordPerfect PRS driver) to print them.* If you have problems printing either of these documents, please call us (Badger SoftWerks, phone 608/437-3348) and we'll be happy to help you. Trim the TechType Quick-Reference Card with a scissors, and tape it to the top or bottom edge of your monitor.

Notes for Network Installation: The TechType Installation macro will properly install TechType to a network as well as to a stand-alone PC, but installation for two or more networked users may require an extra step, depending on your network configuration.

Note in particular that, for any network installation, *each TechType user must be set up for both read and write access to his/her designated WPWIN default macro path!* (The Trial Monitor macro for this demo edition regularly reads and writes to the TECHTYPE.INI file, which resides in the user's WPWIN default macro path.)

If your networked WPWIN is configured with separate default macro paths for different users (the typical configuration), then each user may simply run the Installation macro separately to install TechType for his/her own personal use. If, however, your networked WPWIN is configured with a single common default macro path shared by all users, and you want more than one user to have access to TechType, then be sure to see the *Extra Step for Networked Users Sharing a Common Macro Path* at the end of these instructions.

★ *To install TechType 1.0-D on your computer or network:* Run TechType's Installation macro (TT10D-IN.WCM), from its source directory (you need not first copy the TT10D-IN.WCM macro file to your default macro path):

1. Start WPWIN (if it isn't already running), and press Alt+F10 to display WPWIN's Play Macro dialog box.

2. Use the Quick List (or Directories list) in the dialog box to find the TechType 1.0-D master program files. They will reside on drive A: or B: if you are installing from a floppy diskette, or in one of your hard-drive directories if you have downloaded TechType from a BBS.

3. Select the TT10D-IN.WCM file, and choose the Play button.

Once you have activated TechType's Installation macro, simply follow the macro's prompts and enter your appropriate choices until the macro has run its course.

▸ *Extra Step for Networked Users Sharing a Common Macro Path:*

If your networked WPWIN is configured with a single common default macro path shared by all users, and you want more than one user to have access to TechType, then *each individual user* must manually select (activate) a TechType keyboard after the Installation macro has copied the program files and initialized your single, common TECHTYPE.INI file. If necessary, see the topic *Keyboard Layout* in your WPWIN *Reference* for the procedure to select one of these TechType 1.0-D "soft" keyboards:

TT10DCUA.WWK:	CUA-compatible keyboard file
TT10DDOS.WWK:	WP-DOS-compatible keyboard file

The Installation macro can (optionally) automatically select a TechType keyboard for whichever user performs the installation, but all other users must then individually, separately, select a TechType keyboard for access to TechType features.

1.3 Technical Support for TechType 1.0-D

adger SoftWerks provides unlimited free technical support for all demo versions of all of its software. In fact, we encourage calls from trial users on any subject pertaining to the performance, useability or underlying philosophy of our software. Certainly call us if you experience any problems. And please call us if you have any questions, or any comments or suggestions for how we might improve the software's design, performance or documentation, or our business practices, or the pass-rush capabilities of the University of Wisconsin football team.

Badger SoftWerks: 608/437-3348

2. TechType Features: Operating Instructions

2.1 Keyboard-Direct Characters

echType's Keyboard-Direct Characters feature provides single-keystroke direct access to 28 common but non-standard characters: an assortment of 16 popular bullet characters, plus §, ¶, †, º, ©, ®, ™, ≤, ≥, — (em dash), – (en dash), and … (ellipsis). Most of these 28 keyboard-direct characters are of no particular value to scientific/math/med/engineering typing, but are included in TechType as an added convenience simply because they are so commonly used in *general* typing. Some of the characters, however, *will* be of particular interest to sci/tech typists, and some are *required* for the use of other TechType features.

Twenty of the 28— including all 16 bullets— are assigned to Ctrl and Ctrl+ Shft with the number keys 1 through 0. The others are assigned to intuitive combinations: Ctrl+Shft+S = §, Ctrl+Shft+O = º, etc. The lower-left corner of your TechType Quick-Reference Card identifies all 28 characters and the keystrokes to which they are assigned.

The keyboard-direct characters that will be of particular interest to sci/tech typists, and to users of other TechType features, are:

≤ = Ctrl+[,] ≥ = Ctrl+[.] º
= Ctrl+Shft+O

† (a "key" character in some TechType expressions) = Ctrl+[/]
§ ("key" character for Super/Sub Stacker™ feature) = Ctrl+Shft+S
— (em dash, also valid w/in ChemType expressions) = Ctrl+Shft+M
– (en dash, also commonly used as "minus" symbol)= Ctrl+Shft+N

2.2 TechType Characters

he TechType Characters feature provides quick two-step keyboard shortcuts to 104 characters commonly used in scientific, math, medical, engineering and other technical typing: fractions, arrows, Greek characters, math functions/relations, and med-notation symbols. TechType characters are inserted into a WPWIN document by typing a TechType "expression" (explained below) and then pressing Ctrl+Shft+[/] (the "slash" key). The Ctrl+ Shft+ [/] keystroke activates a macro that quickly reads the preceding expression and converts it to its corresponding TechType character.

★ *To insert a TechType character into a WPWIN document:*

Type the TechType expression that corresponds to the character you want to insert, and press Ctrl+Shft+[/].

WHAT IS A TECHTYPE "EXPRESSION"?

A TechType "expression" consists of two parts: a "key" character ("/" or "†") and a one- to three-character "abbreviation":

The key character identifies the beginning of a TechType expression, which might otherwise be ambiguous to the macro since TechType characters sometimes immediately follow other words or characters with no intervening spaces. Key characters also divide the 104 TechType characters into two groups: the "/" key character is used for all fractions and Greek characters, while the "†" key character is used for all other TechType characters. The "abbreviation" part of a TechType expression identifies the specific TechType character to be inserted.

(The "†" character is one of 28 common but non-standard characters assigned to various keystrokes in TechType's Keyboard-Direct Characters feature, described separately in section 2.1 above. With a TechType Keyboard selected, you may type the "†" character by simply pressing Ctrl+[/].)

Your TechType Quick-Reference Card contains a chart that will help you quickly determine the correct expression to type for each of the 104 TechType characters. The appropriate key character appears at the left of each line, and the abbreviations appear directly under the corresponding TechType characters. The TechType expression for "≠," for example, would be "†ne": the "†" key character plus the "ne" abbreviation for the "not-equal-to" character:

∼	≈	∝	≪	≫	≠	≮	≰	≯	≱	≃	≄
si	ap	pr	ml	mg	ne	nl	nle	ng	nge	se	ns

A Few Rules and Hints

▸ *Several common sci/tech characters are available as Keyboard-Direct Characters.* You won't find these characters in the TechType Characters chart, because they are available directly, via single keystrokes:

≤	= Ctrl+[,]	º	= Ctrl+Shft+O
≥	= Ctrl+[.]	–	= Ctrl+Shft+N (en dash, useful as a "minus")

▸ *No codes allowed!* The TechType macro reads the expression backwards, one character at a time, and will abort itself and report an "invalid TechType expression" if it encounters *any* WPWIN codes. You *may,* however, enclose the *entire* expression within attribute on/off codes, as in:

[Bold On]/t[Bold Off]

... but you must then execute the macro (press Ctrl+Shft+[/]) from *inside* the [Bold Off] code (between the "t" and the [Bold Off]).

Note: The only exception to this "no codes allowed" rule is WPWIN's standard hyphen character ("[-]," which WPWIN regards as a code), and then *only* when it is used in the TechType expression "†-" (for the "" TechType character).

▸ *Always run the TechType macro (Ctrl+Shft+[/]) from the first possible insert point following the TechType expression.* If you enter a space after the expression and then run the macro, TechType will read the space, abort itself, and report an "invalid TechType expression."

▸ *Uppercase and lowercase* ***are*** *important in TechType expressions.* For example, the expression "/D" converts to "Δ" (uppercase delta), but "/d" converts to "δ" (lowercase delta). Similarly, the expression "†r" converts to "→," but "†R" converts to "." If you enter the wrong upper/lowercase for any alpha character in a TechType expression, the macro will convert to the alternate (wrong) character if one exists, or will otherwise abort and report an "invalid TechType expression."

▸ *The TechType Characters feature* ***may*** *be used in conjunction with the Custom Fractions, Super/Sub Stacker and ChemType features.* These features are described separately in later sections of this chapter, which also offer special notes about including TechType characters within custom fractions, stacked super/subscripts, and ChemType expressions.

▸ *FYI: A few of the TechType characters are not really single characters, per se, but rather are composites created with WPWIN's Overstrike feature.* This fact has no effect on your procedures for inserting these "characters," since the macro will create the Overstrikes automatically, but we thought we should mention it. Most of the med-notation symbols are Overstrike composites ( ), as is the "x-bar" character ( ).

> Frankly, we ourselves are not satisfied with the appearance of the , , , and "characters," in which the circled characters are not quite centered (in most common fonts), and the on-screen displays look even worse than the printed characters. But these symbols seem to be in great demand, so we are providing them in these imperfect forms rather than not at all. If you know of a better way to produce these symbols (the method must yield uniformly better results across a variety of common fonts), please call us and we'll consider your alternatives for our next release.

2.3 Custom Fractions

echType's Custom Fractions feature automates the process of setting pseudo-typographic fractions that are not available as "WP Characters." WPWIN's WP Characters include true typographic fractions for ½, thirds, fourths and eighths, and these are all available via TechType Characters (see section 2.2 above). But many WPWIN users occasionally need to enter other fractions— "$^{9}/_{32}$," for example— that simply do not exist in WPWIN's character sets. It's no big trick to create a pseudo-typographic fraction— simply superscript the numerator and set the denominator to the Fine font size— but the procedure is cumbersome without this automatic feature.

To create a custom fraction, you enter a special type of TechType "expression," and press Ctrl+Shft+[/] (the "slash" key) to run the TechType macro. (If you are unfamiliar with TechType expressions, please see section 2.2 earlier in this chapter.) The macro then reads the expression and converts it to a pseudo-typographic fraction.

The TechType expression for a custom fraction uses a "/" for the key character, followed by a colon, the text of the numerator, another colon, and the text of the denominator. The expression for "$^{9}/_{32}$," for example, is "/:9:32."

/:n:d … converts to … $^{n}/_{d}$

★ *To create a custom fraction in a WPWIN document:*

Type a "slash" and a colon ("/:"), followed by the numerator value, another colon, and the denominator value— all with no intervening spaces— and then press Ctrl+Shft+[/].

A Few Rules and Hints

▸ *No codes allowed!* The TechType macro reads the expression backwards, one character at a time, and will abort itself and report an "invalid TechType expression" if it encounters *any* WPWIN codes. You *may,* however, enclose the *entire* expression within attribute on/off codes, as in:

[Bold On]/:3:16[Bold Off]

... but you must then execute the macro (press Ctrl+Shft+[/]) from *inside* the [Bold Off] code (between the "6" and the [Bold Off]).

▸ *Always run the TechType macro (Ctrl+Shft+[/]) from the first possible insert point following the TechType expression.* If you enter a space after the expression and then run the maco, TechType will read the space, abort itself, and report an "invalid TechType expression."

▸ *The numerator and denominator values may contain any combination of alpha, numeric and/or symbol characters, including TechType characters, but may **not** include any spaces.* The expression "/:a+b:c+d," for example, converts to "$^{a+b}/_{c+d}$." Please note, however:

▸ *The prohibition against codes within a TechType expression means that WPWIN's standard hyphen character ("[-]"), which WPWIN regards as a code, may **not** be used in a Custom Fraction expression.* You may substitute either a "hard hyphen" character (press Ctrl+[-]), or an en-dash character (Ctrl+Shft+N on any TechType keyboard). Most users prefer the en dash for a "minus" symbol.

▸ *To include a TechType character within a custom fraction:* You must convert all TechType Character expressions to their corresponding TechType characters *before* you run the macro to convert the Custom Fraction expression to a custom fraction. For example, to create the custom fraction "$^{\alpha}/_{\beta}$," you would:

1. Type "/:/a" (the opening "/:" for the custom fraction, followed by "/a"— the TechType expression for "α"), and press Ctrl+Shft+[/].

   The TechType macro then reads back to the "/" in "/a," and converts that sub-expression to "α." What you see on your screen then is: "/:α."

2. Type ":/b" (the second colon, followed by "/b"— the TechType expression for "β"), and press Ctrl+Shft+[/].

   The TechType macro then converts the "/b" sub-expression, and your screen displays: "/:α:β."

3. Press Ctrl+Shft+[/] again, to convert the Custom Fraction expression "/:α:β" to the custom fraction "$^{\alpha}/_{\beta}$."

This rule is more easily understood when you remember that the TechType macro reads expressions *backwards* from the insertion point, and will begin to evaluate the expression at the first "key" character ("/," "†" or "§") it encounters. Any unconverted subexpressions within a Custom Fraction expression, therefore, will trigger the macro's evaluation function before the full Custom Fraction expression has been read.

- *You **may** include a "/," "†" or "§" character within a Custom Fraction expression, but **only** if you double it ("//," "††" or "§§").* When the macro encounters any of these key characters, it checks to see if the character is doubled. If the character *is* doubled, the macro will delete the duplicate character and resume reading the expression, rather than treating the character as a key character. For example: "/:1//2:3" converts to "$^{1/2}/_{3}$."

- *To create a custom fraction that includes spaces, first type and convert the expression with**out** spaces, and then go back into the converted fraction and insert the spaces.* (Remember that a space within a TechType expression will cause the macro to abort.) For example: You may type the expression "/:(a+b):(c+d)" and convert it to "$^{(a+b)}/_{(c+d)}$," and then edit the converted fraction to add spaces: "$^{(a + b)}/_{(c + d)}$,"

- *To create a custom fraction that includes font size/appearance attributes for only a portion of the fraction, first type and convert the expression with**out** attributes, and then go back into the converted fraction to apply the attributes.* (Remember that a code within a TechType expression will cause the macro to abort.) For example: You may type the expression "/:a:(1–a)" and convert it to "$^{a}/_{(1–a)}$," and then apply italics to only the denominator: "$^{\mathrm{a}}/_{\mathit{(1–a)}}$."

2.4 Super/Subscripter

ost scientific and other technical typing requires the frequent use of superscripts and subscripts— for exponential notations, chem notations (elemental proportions and valences), variable arrays (k_1, k_2, etc.), and a variety of other applications. WPWIN offers super/ subscripting capabilities, of course, but we've always felt that the path to them (via the Font menu) is awkward and less than intuitive. We have therefore assigned these functions to the Alt+ Shft+ [↑] and Alt+Shft+[↓] keys on the TechType keyboards. These are "toggle" functions; that is, the same keystroke turns the attribute on or off, depending on its status at the insertion point.

★ *To turn on or off WPWIN's superscript or subscript attribute:*

Press Alt+ Shft+ [↑] to toggle the superscript attribute on or off, or Alt+Shft+[↓] to toggle the subscript attribute on or off.

Other TechType features— ChemType, Custom Fractions and Super/Sub Stacker— automatically apply super- and subscripts to appropriate characters in the course of their work, and these features will eliminate the need for much manual super- and subscripting for many users. But super- and subscripts are so pervasive in sci/med/tech typing that TechType's Super/Subscripter should be a welcome addition to most users' arsenals of tech-typing tools.

2.5 Super/Sub Stacker™

"stacked" super/subscript is a pair of values, one superscripted and the other subscripted, and aligned so that the subscripted value appears directly under the superscripted value. The most common applications of stacked super/subscripts are found in mathematics and statistics (e.g., Σ^a_b and $\int^a_b$), but they occasionally sneak into other disciplines as well. The simplest way to create a stacked super/subscript in WPWIN is to use WPWIN's Advance feature between the two values, to shift the subscript back to the same horizontal position as the beginning of the superscript. TechType's Super/Sub Stacker feature automates that process, and adds this enhancement: If the subscript value is shorter (horizontally) than the superscript value, Super/ Sub Stacker will add another Advance code at the end, so that your next typed character will always begin at the end of the longer value.

To create a stacked super/subscript, you enter a special type of TechType "expression," and press Ctrl+Shft+[/] (the "slash" key) to run the TechType macro. (If you are unfamiliar with TechType expressions, please see section 2.2 earlier in this chapter.) The macro then reads the expression and converts it to a stacked super/subscript.

The TechType expression for a stacked super/subscript differs from the Custom Fraction expression (described in section 2.3 above) only in its key character: "§." The expression begins with the "§" key character and a colon, followed by the text of the superscript value, another colon, and the text of the subscript value:

§:super:sub ... converts to ... $^{\text{super}}{}_{\text{sub}}$

(Note that the "§" key character is one of TechType's Keyboard-Direct Characters, available by pressing Ctrl+Shft+S on any TechType keyboard.)

★ *To create a stacked super/subscript in a WPWIN document:*

Type the key character and a colon ("§:"), followed by the superscript value, another colon, and the subscript value— all with no intervening spaces— and then press Ctrl+Shft+[/].

A Few Rules and Hints

▸ *WPWIN's dysfunctional WYSIWYG often fails to display some portion of a stacked super/subscript and adjacent text, and occasionally wipes out the entire line. But the text and codes are really there, and should print correctly.* This may vary depending on your display adapter, and we hope you won't find it too annoying. We've verified that the same problem occurs when we create a stacked super/subscript manually, without the macro, so we feel safe in saying that this is WPWIN's problem— not ours.

▸ *No codes allowed!* The TechType macro reads the expression backwards, one character at a time, and will abort itself and report an "invalid TechType expression" if it encounters *any* WPWIN codes. You *may,* however, enclose the *entire* expression within attribute on/off codes, as in:

[Bold On]§:super:sub[Bold Off]

... but you must then execute the macro (press Ctrl+Shft+[/]) from *inside* the [Bold Off] code (between the "b" in "sub" and the [Bold Off]).

▸ *Always run the TechType macro (Ctrl+Shft+[/]) from the first possible insert point following the TechType expression.* If you enter a space after the expression and then run the maco, TechType will read the space, abort itself, and report an "invalid TechType expression."

▸ *Super/Sub Stacker must be run in left- or full-justified text (only), or the aligning Advance code(s) will be inaccurate.* The macro reads the horizontal position of the insert point before and after it has typed the superscript value, to determine how far it needs to Advance to the left before typing the subscript value. In other words, the macro assumes that the starting point of the superscript value will remain fixed as the value is typed, but this would be a false assumption in centered or right-aligned/justified text. You *may,* however, use centering or right-justification with no ill effects *after* the macro has converted an expression to a stacked super/subscript.

▸ *The superscript and subscript values may contain any combination of alpha, numeric and/or symbol characters, including TechType characters, but may **not** include any spaces.* The expression "§:k–1:x," for example, converts to "$^{k-1}{}_{x}$." Please note, however:

▸ *The prohibition against codes within a TechType expression means that WPWIN's standard hyphen character, which WPWIN regards as a code ("[-]"), may **not** be used in a Super/Sub Stacker expression.* You may substitute either a "hard hyphen" character (press Ctrl+[-]), or an en-dash character (Ctrl+Shft+N on any TechType keyboard). Most users prefer the en dash for a "minus" symbol.

▸ *To include a TechType character within a stacked super/subscript:* You must convert all TechType Character expressions to their corresponding TechType characters *before* you run the macro to convert the Super/Sub Stacker expression to a stacked super/subscript. This is the same rule that applies to the Custom Fractions feature (see section 2.3 above for an example). Remember that the TechType macro reads expressions *backwards* from the insertion point, and will begin to evaluate the expression at the first "key" character ("/," "†" or "§") it encounters. Any unconverted sub-expressions within a Super/Sub Stacker expression, therefore, will trigger the macro's evaluation function before the full expression has been read.

▸ *WPWIN does not always accurately read the widths of TechType characters and other "WP Characters," which can distort the macro's Advance-code alignment(s).* That is, the macro may shift the subscript value too far back, or not far enough. This problem seems to occur only with a few of the non-standard characters that WPWIN creates graphically (which WPWIN does automatically if the "true" character is not available in the selected font). The only solution we've found for this problem is to manually edit the Advance code, to force a true alignment.

▸ *You **may** include a "/," "†" or "§" character within a Super/Sub Stacker expression, but **only** if you double it ("//," "††" or "§§").* When the macro encounters any of these key characters, it checks to see if the character is doubled. If the character *is* doubled, the macro will delete the duplicate character and resume reading the expression, rather than treat the character as a key character. For example: "§:a//b:c" converts to "$^{a/b}_{c}$."

▸ *To create a stacked super/subscript that includes font size/appearance attributes for only a portion of the stack, first type and convert the expression with**out** attributes, and then go back into the converted stack to apply the attributes.* (Remember that a code within a TechType expression will cause the macro to abort.) For example: You may type the expression "§:k–1:x" and convert it to "$^{k–1}_{x}$," and then apply italics to only the superscript value: "$^{\mathit{k–1}}_{x}$."

▸ *Unlike custom fractions, a stacked super/subscript (after conversion) usually can**not** accommodate inserted spaces without a manual adjustment to the Advance code(s).* A space within a TechType expression will cause the macro to abort, but in a custom fraction you can go back and add any spaces you like *after* converting the expression. A stacked super/subscript, however, contains one or two Advance codes to align the subscript under the superscript, and to leave the cursor at the end of the longer element. The distance of either Advance is automatically calculated for the stack as it appeared *without* spaces, so the alignment will be lost if you insert spaces *after* the conversion.

2.6 ChemType™

hemical notations are typically full of subscripts and superscripts (e.g., H_2SO_4 and Na^+Cl^-) and upper/lowercase changes (for element notations), and are therefore difficult to type— in WPWIN or any word processor. Even with TechType's Super/Subscripter, it's irritating to have to manually apply and turn off a superscript or subscript every two or three characters, or every other character, and to shift into and out of uppercase at about the same frequency.

TechType therefore offers a ChemType feature that lets you "power-type" chemical notations in plain, lowercased text, and then quickly convert the plain-text entries to true chemical notations, with appropriate upper/lowercase alpha characters for element notations, and appropriate super/subscripts for digits and valences. To create the chemical notation "Na_2ZnO_2," for example, you can simply type "na2zno2" and then press Alt+[/] to activate the ChemType macro. The macro then reads the plain-text entry and converts it to the true chem notation.

ChemType is thus another conversion feature, similar to the TechType Characters, Custom Fractions and Super/Sub Stacker features described above, inasmuch as the macro reads, interprets and converts an "expression" that you have typed. For ChemType, the "expression" is your plain-text entry, and you will note that ChemType expressions do *not* include a "key" character, unlike the expressions for TechType's other conversion features. The next subsection below provides a more complete explanation of ChemType expressions.

For most users with 386-processor computers, ChemType will be a "hassle saver" more than a time saver. That is, the ChemType conversions may not save you any time compared to your creating the notations manually, and for some expressions may even take a second or two longer, but you *will* be spared the keystrokes and hassle of having to manually enter and release all of the super/subscript attributes and upper/lower case changes. (The macro algorithm that interprets sequences of alpha characters to determine upper/ lowercase is complex, and therefore time-consuming.) But users with 486-class and Pentium processors should realize satisfying time savings as well.

We've tried to design ChemType so that it will be useful to a broad range of users, from professional scientists, med/health practitioners and researchers, and science/ health educators, to the more casual chem-notation "name droppers" (those whose chemistry vocabularies do not extend far beyond O_2, H_2O, N_2, etc., but who are nevertheless fond of these shorthand notations). We've also built some versatility into ChemType, so that you may apply it both to single-"word" notations (e.g., H_2SO_4, $(NH_4)_2SO_4$, etc.), and to longer reaction notations that contain spaces (e.g., $Ba^{2+} + 2\ OH^- + CO_2 \rightarrow BaCO_3 + H_2O$).

Finally, ChemType offers a Preferences feature that lets you customize ChemType's operations for the sorts of notations that appear most commonly in your own work. ChemType Preferences and how to set/change them are explained in a separate subsection following the *Rules and Hints* below.

★ *To convert a single-"word" ChemType expression to chem notation:*

Leave the cursor (or reposition it) at the first possible insert point following the plain-text expression you want to convert (no intervening spaces), and press Alt+[/].

★ *To convert a multi-"word" ChemType expression to chem notation:*

Select the entire expression you want to convert, and press Alt+[/].

ChemType Expressions

ChemType expressions do *not* include "key" characters, unlike the expressions for the other TechType conversion features. With ChemType, you simply type the plain-text expression and activate the macro.

A ChemType expression may include any alpha or numeric character (except for "j" and "q," which do not exist in any element notations), but only a few other special characters— for valences, groupings and reactions— are valid. Most of the rules for valid vs. invalid characters in ChemType expressions are either self-evident or seldom applicable, so you need not spend much time studying these rules. (ChemType is much easier to use than the rules list here might suggest.) Instead, simply read through the rules once, and then just remember where they are in case you need to refer to them later.

A complete list of valid characters in ChemType expressions, with explanations of how the ChemType macro interprets them:

▸ *Alpha Characters (AaBbCc ... Zz, except for Jj and Qq):* Assumed to be element notations, or portions thereof. The macro reads and interprets sequences of alpha characters (e.g., "naoh," "zno," "feso," "kmno," etc.) to determine upper/lowercase for element and compound notations. Note:

▸ The macro ignores your input case, so that you can "power-type" the expression without regard for upper/lowercase. You don't gain any performance benefit by entering the correct case in the input expression, since the macro converts all alpha characters to lowercase before it begins its interpretations.

▸ If you enter an ambiguous alpha sequence (e.g., "co" could be "CO" or "Co"), the macro will prompt you to resolve it or, if your ChemType Preferences are set to "Express Mode" (explained below), may assume that certain common combinations are one way or the other.

- *Numeric Characters (1234567890):*
  - Digit(s) that follow an alpha character *or* that follow ")," "]" or "}":
    - If the digit(s) are *not* followed by "+," "-" or "–": Digit(s) are assumed to be proportion notations, and are *sub*scripted (as in "O_2").
    - If the digit(s) *are* followed by "+," "-" or "–" (e.g., "na2+"): Digit(s) are assumed to be part of the valence notation, and are therefore *super*scripted with the valence character (as in "Na^{2+}").
  - Digits that *precede* an alpha character or "(," "[" or "{" are superscripted (as in isotopic notations— e.g., "^{14}C"), or are left normal (as in balanced equations— e.g., "$2CO_2$"), depending on your ChemType Preferences setting, explained below.
- Other valid characters:
  - *+ - – :* Superscripted as a valence notation only if preceded by an alpha character (e.g., "Na^+"), *or* by digit(s) that follow an alpha character (e.g., "Fe^{2+}"). Otherwise the character is left normal (no superscript or subscript).

    Note: The "-" character is a hard hyphen (Ctrl+[-]), and the "–" character is an en dash (Ctrl+Shft+N on a TechType keyboard). WPWIN regards its standard hyphen character ("[-]") as a code, so the standard hyphen character is invalid within a ChemType expression.
  - *Space character:* Always left normal (no superscript or subscript), regardless of context.
  - *)] } : { [(:* Always left normal (no superscript or subscript), regardless of context.
  - *← → ↔ ⇆ ⇄ ⇇ ⇉ ● — :* Always left normal (no superscript or subscript), regardless of context. Note, however, that while WPWIN's "WP Characters" offer alternatives for some of these characters, *ChemType will recognize and accept only these specific characters:*

Character	"WP Char" #	Type Via TechType Keyboard:

←	6,22	†l	+ Ctrl+Shft+[/]
→	6,21	†r	+ Ctrl+Shft+[/]
↔	6,25	†<>	+ Ctrl+Shft+[/]
⇆	6,55	†lr	+ Ctrl+Shft+[/]
⇄	6,54	†rl	+ Ctrl+Shft+[/]
⇇	6,159	†ll	+ Ctrl+Shft+[/]
⇉	6,158	†rr	+ Ctrl+Shft+[/]
●	4,0	Ctrl+1 (the digit "one")	
— (em dash)	4,34		Ctrl+Shft+M

A Few Other Rules and Hints

▸ *For single-"word" ChemType expressions:* The expression *must* be preceded by a space, or a [HRt], [SRt], [HPg], [Tab], etc. The ChemType macro defines the content of the expression by turning on WPWIN's Select mode at the insertion point (wherever you activate the macro) and then jumping backwards to the beginning of the current or preceding word.

▸ *No codes allowed!* The ChemType macro will delete any WPWIN codes that occur within the expression. You *may,* however, enclose the *entire* expression within attribute on/off codes, as in:

[Bold On]c6h12o6[Bold Off]

... but you must then execute the macro (press Ctrl+Shft+[/]) from *inside* the [Bold Off] code (between the "6" in "o6" and the [Bold Off]).

▸ *A ChemType expression may not include WPWIN's standard hyphen character ("[-]"), which WPWIN regards as a code.* You may substitute a "hard" hyphen (Ctrl+[-]) or an en dash (Ctrl+Shft+N on any TechType keyboard) for a "minus" symbol in a valence or reaction. Most users prefer the en dash for "minus" symbols, especially in valence notations where the superscript attribute renders hard hyphens barely visible.

Setting and Changing Your ChemType Preferences

ChemType lets you customize its operations for the types of notations and compounds that appear most commonly in your work.

★ *To set or change your ChemType Preferences:*

Type any *in*valid ChemType expression (e.g., "jq" or "zz"), and then run the ChemType macro (press Alt+[/]).

Whenever you try to convert an invalid ChemType expression, ChemType aborts the conversion and displays an error message that also offers you the option of changing your ChemType Preferences:

If you choose the "No" option, ChemType simply clears the message box and leaves you in your original WPWIN edit screen. If, however, you choose the "Yes" option, ChemType displays its Specify Preferences dialog box:

As you can see, explanations of these options and their significance appear within the dialog box. Simply set each of the two pairs of buttons to your preference, and then choose "OK" to save your settings.

Your ChemType Preferences are stored in a Windows ANSI text file named TECHTYPE.INI, which must reside in your WPWIN default macro path. Like your WIN.INI, SYSTEM.INI and other INI files, TECHTYPE.INI must be saved in ANSI text format, or the macro will be unable to read it. (We strongly recommend against manually editing or even opening TECHTYPE.INI. But if you do, remember to use WPWIN's Save As feature to save the file as ANSI text.)

If you are running WPWIN and TechType on a network, different users may specify different ChemType Preferences settings *only* if WPWIN is configured with separate default macro paths for individual users, rather than a single common macro path shared by all users.

2.7 Displaced WPWIN Features in the TechType Keyboards

lmost all of the TechType features and special characters assigned to the TechType keyboards are simply added to the original default keystrokes for WPWIN's CUA- and WP-DOS-compatible keyboards. This means that you do not need to unlearn the keystrokes for other WPWIN features in order to use TechType. But there are three exceptions, and they are:

▸ *HyphenIgnoreWord:* Originally Ctrl+[/], but moved in the TechType keyboards to Alt+Shft+[/], to permit the assignment of the "†" character to Ctrl+[/]. The HyphenIgnoreWord feature inserts a [Hyph Ign Wrd] code at the insertion point. It's a handy code (see your WPWIN *Reference* if you are unfamiliar with it), but not so commonly used that this reassignment would pose an inconvenience to many users.

▸ *SelectCellUp:* Originally Alt+Shft+[↑], but removed in the TechType keyboards to permit the assignment of WPWIN's Superscript function to Alt+ Shft+ [↑] (for TechType's "Super/Subscripter"). Our research suggests that 98.814% of WPWIN users are utterly unaware that the SelectCellUp function even exists, and that the remaining 1.186% won't mind manually turning on the Select mode and moving up one cell as a substitute.

▸ *SelectCellDown:* Originally Alt+Shft+[↓], but removed in the TechType keyboards to permit the assignment of WPWIN's Subscript function to Alt+ Shft+ [↓] (for TechType's "Super/Subscripter"). See the notes above for SelectCellUp.

If these displacements are inconvenient for you, please see section 3.1 below for information about customizing your TechType keyboards.

3. Advanced Topics

3.1 Customizing Your TechType Keyboard Files

Feel free to add your own macros to either TechType WWK file, and/or to add or reassign any WPWIN feature or function to a particular key. The TechType keyboard files are fully customizeable, just like any other WPWIN keyboard file. (See your WPWIN *Reference* for more information.) Please note, however, these strong recommendations:

▸ *Make a copy of TechType's original *.WWK file, and then edit* ***only*** *the copy.* That way you can always return to the original, as a baseline, to experiment with alternative customizations.

▸ *Do not remove the TT10D-X1.WCM and TT10D-X2.WCM macros from the Assignable Items list within WPWIN's Keyboard Editor!* TT10D-X1.WCM and TT10D-X2.WCM reside *only* within the WWK files, so they will be lost if you remove them. These are very small macros that simply provide links between the keyboard keystrokes and the much larger TT10D-01.WCM and TT10D-02.WCM macro files, which do reside separately in your default macro path.

We've designed TechType so that the large macro files are separate from the WWK files, because WPCorp's tech-support people suspect that an overly inflated WWK file may hinder performance (system speed) and may even cause memory conflicts. (Since WPWIN needs constant access to the WWK file's keystroke interpretations, WPWIN must always hold the WWK file in memory.) Without the TT10D-X1.WCM and TT10D-X2.WCM macros you would have to import the larger files directly into the WWK file, which would create a much larger WWK file.

If you should inadvertently remove TT10D-X1.WCM or TT10D-X2.WCM, see section 3.2 below for information about how to recreate them and restore them to your keyboard.

3.2 Adding TechType Features to an Existing Keyboard File

If you already use a heavily customized *.WWK keyboard definition, you may prefer adding the TechType features to your existing keyboard rather than re-creating your customizations in one of TechType's keyboards. To assign the TechType features and non-standard characters to an existing keyboard definition, you will need to use WPWIN's Keyboard Editor. (If necessary, refer to your WPWIN *Reference* for Keyboard Editor procedures.)

Before you begin work with the Keyboard Editor, however, you must create and save two small WPWIN macro files, as follows:

1. Within WPWIN, open a new document, and type the following text *exactly as it appears here:*

```
Application (WP; WPWP; Default; "WPWPUS.WCD")

GetWPData (macpath; MacroPath!)
CHAIN (macpath+"tt10d-01.wcm")
```

(The "0" characters in the macro filename are both zeros.)

2. *Without closing the file,* save it (press F3) into your WPWIN default macro path under the name TT10D-X1.WCM.

3. In the same (already open) file, edit the "CHAIN" line to read:

```
CHAIN (macpath+"tt10d-02.wcm")
```

4. Use WPWIN's Save As option (press F3) to save this *edited* version in your default macro path under the new name TT10D-X2.WCM, and then close the file (Ctrl+F4).

5. In a blank (empty) document window, play each of the two new macros once. For each macro (TT10D-X1.WCM and TT10D-X2.WCM): Press Alt+F10, type the macro file name (or select it from the file list), and choose the Play option. Since you haven't entered any expression for the macros to convert, the TechType macro will report an "invalid TechType expression," and the ChemType macro will simply abort itself. *This step is necessary to compile the new macros, since WPWIN's Keyboard Editor will not let you import a macro until it is compiled.* This step also will verify that all of the TechType macros— including the two new ones you have just created — reside in your WPWIN default macro path.

When the new macros are compiled, you may use WPWIN's Keyboard Editor to edit your existing keyboard definition. The TechType keystrokes are the same for both the CUA- and WP-DOS-compatible keyboards, so the instructions below are appropriate for editing either basic layout. Note, however, that some TechType key assignments may conflict with existing key assignments in your customized keyboard, and if so you will have to make your own decisions as to which features and characters to move to which keys. These instructions assume that you will encounter no such conflicts, so you may want to proceed slowly, to make sure that you notice any already-occupied keystrokes before you enter any changes.

Important Note: Several of the TechType key assignments will require that you mark the "Allow Reassigning of a,b,c..., 1,2,3..." box in WPWIN's Keyboard Editor, so that WPWIN will be able to recognize particular key combinations. Other key assignments, however, will be recognized only when the "Allow Reassigning ..." box is *not* marked. We've found that WPWIN's Keyboard Editor can be quirky with respect to the setting in this box (marked vs. unmarked), so you may need to switch the setting back and forth as you work your way through the procedure described below. A note of caution: *Always verify that the correct function and keystroke appear in the "Change Assignment" box (lower-right corner of dialog box),* ***before*** *you choose Assign.* If WPWIN refuses to recognize a particular keystroke, change the "Allow Reassigning ..." setting. Then, if WPWIN still does not recognize the keystroke, try changing the setting back again. (For some mysterious reason, a back-and-forth change often works if the original setting and its alternate both fail.)

To add TechType features to an existing keyboard file (beginning within WPWIN's Keyboard Editor):

1. With the Item Type (under "Assignable Items") set to "Commands," assign "HyphenIgnoreWord" to the Alt+Shft+[/] keystroke. Note that this command will then be assigned to two different keystrokes (or three in a DOS-compatible keyboard). In step 3 below, however, the "†" character will replace (automatically Unassign) "HyphenIgnoreWord" at its original Ctrl+[/] location.

2. With the Item Type still set to "Commands," assign "FontSubscriptToggle" to Alt+Shft+[↓], and "FontSuperscriptToggle" to Alt+Shft+[↑]. These assignments will replace (automatically Unassign) WPWIN's "SelectCellDown" and "SelectCellUp" commands, which will *not* be reassigned. (Feel free to reassign SelectCellUp/Down to alternate keystrokes if you like, but these functions are so seldom used by most users that reassignment is usually unnecessary.)

3. Change the Item Type to "Text," and use the "Add" option to define the text items below and assign them to their corresponding keystrokes. (In the "Character" columns, the numbers in parentheses are the "WP Characters" character numbers.)

Character	Assign To:	Character	Assign To:
† (4,39)	Ctrl+[/] ("slash")	♦ (5,1)	Ctrl+4
§ (4,6)	Ctrl+Shft+S	◊ (6,95)	Ctrl+Shft+4
º (4,16)	Ctrl+Shft+O (letter "oh")	■ (4,46)	Ctrl+5
≤ (6,2)	Ctrl+[,] (comma)	□ (4,38)	Ctrl+Shft+5
≥ (6,3)	Ctrl+[.] (period)	☒ (5,25)	Ctrl+6
— (4,34)	Ctrl+Shft+M	☐ (5,24)	Ctrl+Shft+6
– (4,33)	Ctrl+Shft+N	⋆ (6,112)	Ctrl+7
… (4,56)	Ctrl+Shft+[.] (period)	★ (6,184)	Ctrl+Shft+7
● (4,0)	Ctrl+1	✓ (5,23)	Ctrl+8
○ (4,1)	Ctrl+Shft+1	» (4,10)	Ctrl+Shft+8
▸ (6,27)	Ctrl+2	¶ (4,5)	Ctrl+9
▹ (6,171)	Ctrl+Shft+2	™ (4,41)	Ctrl+Shft+9
◆ (6,96)	Ctrl+3	© (4,23)	Ctrl+0 (zero)
◇ (6,111)	Ctrl+Shft+3	® (4,22)	Ctrl+Shft+0

4. Change the Item Type to "Macros," and:

    1) Add the TT10D-X1.WCM and TT10D-X2.WCM macros to the Assignable Items list.
    2) Assign the TT10D-X1.WCM macro to Ctrl+Shft+[/], and assign TT10D-X2.WCM to Alt+[/].

5. Choose the "OK" option to save these changes.

3.3 Using TechType Features in Your Own Macros

PWIN macros can be created two different ways: by recording keystrokes, or by manually typing macro commands into a new file. Recorded-keystroke macros cannot include any other macros, because WPWIN's Macro Facility cannot play a macro while it is busy recording keystrokes for another. Manually typed macros, however, *can* include other macros via the RUN and CHAIN commands, as explained below.

Not all of TechType's features are driven by macros, so you will need to know which are and which aren't if you are considering using any TechType features in your own macros. The Keyboard-Direct Characters feature and Super/Subscripter are *not* macro-driven, so these features *may* be used in a recorded-keystroke macro simply by pressing the key combinations to which these features are assigned. All other TechType features— TechType Characters, Custom Fractions, Super/Sub Stacker and ChemType— *are* macro driven, and therefore may be included in other macros only via the RUN and CHAIN commands.

To include a macro-driven TechType feature within a WPWIN macro, you must make sure that your other macro commands leave the cursor (or reposition it) at the appropriate starting point for the feature. And of course the macro must find a valid expression, or the "nested" TechType macro will be unable to convert it.

The CHAIN command in a WPWIN macro tells the macro to run a designated (other) macro *upon completion of the* "*host*" *macro.* You can enter the CHAIN command anywhere you like within the host macro, but the chained macro will not run until the "host" macro has run its course.

The RUN command in a WPWIN macro tells the macro to run a designated (other) macro *immediately,* and then return to the "host" macro to resume the host macro's commands.

For more information about the CHAIN and RUN commands— syntaxes, how and why to use them, etc.— please refer to WPWIN's on-line Macro Reference Help.

4. About the "Real" TechType 1.0

he "real" TechType 1.0 is available from Badger SoftWerks (phone 608/437-3348) for a mere $25, plus a $3 shipping/handling charge (all domestic orders shipped by U.S. Priority Mail— add another $3 for international orders). For Wisconsin customers, we are obliged to also collect sales tax at 5% or 5½%, depending on your county. As we release this demo edition, we regret that we are not set up to accept credit-card orders, but we will be happy to process and ship your phone order on your verbal assurance that your check for payment in full will be mailed within 48 hours.

Networked users are required to buy one copy per user, but we do offer discounts for networked users and other multiple-copy orders:

		Shipping/Handling Charge	
Order:	Prod Price (each)	Domestic	Internat'l
1 copy:	$25.00	$3	$6
2–10 copies:	$22.50 (10% disc)	$5/order	$9/order
11+ copies:	$21.25 (15% disc)	$8/order	$12/order

The low $25 base price is possible because we provide the TechType 1.0 User Guide only on diskette— just as in this Demo Edition. The TechType 1.0 product package *does,* however, include a durable, self-adhesive, hard-copy Quick-Reference Card.

Advantages of Owning the "Real" TechType 1.0

- Licensed TechType 1.0 users can use TechType's features forever, rather than for only 30 days as in the 1.0-D Demo Edition. Moreover, TechType 1.0 users will not be annoyed by the first-run-of-the-day Trial Monitor messages that appear in the Demo Edition.

- TechType 1.0 features will run 15% to 35% faster on most computers, since they do not include the Trial Monitor functions of the Demo Edition.

▸ When you call us to place your order, we'll tell you how you can keep running the Demo Edition, beyond your trial-period expiration date, until your receive TechType 1.0.

▸ Licensed TechType 1.0 users are eligible to upgrade to version 1.1 (which will support both WPWIN 5.2 and 6.0) for a mere $10, and to any future versions for only half price.

▸ Licensed TechType users are eligible to upgrade to our "flagship" product, WPWINners™, at a substantial discount. (WPWINners is a collection of dozens of supplemental features and productivity enhancements for WPWIN, and includes all six of the TechType features.) You may upgrade from TechType 1.0 to WPWINners 1.0 (for WPWIN 5.2 only, list price $50) for only $30. Licensed TechType 1.1 users can upgrade to WPWINners 1.1 (for WPWIN 5.2/6.0— list price $60) for only $35.

▸ All Badger SoftWerks products come with a 30-Day Total-Satisfaction/ Total-Refund Guarantee: If you are dissatisfied for any reason, you may return it within 30 days for a full refund of your purchase price.

Technical Support for TechType 1.0/1.1

Badger SoftWerks provides unlimited free telephone support (you pay only for the call) for all of its products, forever, to resolve any problems caused by any "bugs" or design flaws in our software, or errors in our product documentation. For questions whose answers can be found in the User Guide, and for application assistance and all other questions, we provide 30 days of free telephone support (you pay only for the call) beginning on the date of your first call. After the first 30 days, we bill for these "non-bug" calls at the modest rate of $6 per call + 50¢ per minute.

What About Version 1.1?

TechType 1.1 remains under development as this demo version 1.0-D is released (July 8, 1994), but we expect to complete it and release it by early August 1994. TechType 1.1 will contain all of the same features as version 1.0, but will run in both WPWIN 5.2 and 6.0. Licensed users of TechType 1.0 are eligible to upgrade to 1.1 upon its release for only $10.