

GTools, version 1.0.

September 28, 1990

This document contains several short macros that I've written to show what a little ingenuity can make WinWord do. They range from simple toggles to complex dialog box macros.

The document will automatically install the macros you think might be useful. To begin the installation, after you have browsed the doc and read the short descriptions, simply double click on the button below, you will be presented with a dialog list of available macros.

{Install}

Intro...

Most of the macros in this package were developed as a consequence of conversations, questions, challenges, and friendships developed on the Microsoft Applications forum of CompuServe. I wish in particular to acknowledge James Gleick (who wrote several of the cleverest portions of these macros), Barry Simon, Tom Looker, and all the WinWord Gadfly Team.

You can modify these macros to your heart's content. That's why they are there. However, you cannot distribute modified versions of these macros without my express permission. Under no circumstances are these macros to be sold.

ApplyMacroToKey

This macro is adapted from the ApplyStyleToKey macro found in the EXAMPLES.DOC. It allows the quick assignment of a key combination to a macro. Unlike the built in Assign to Key, this allows you to assign Alt-key combinations (and Alt+Shift+Ctrl combinations)

ApplyStyleToKey

Also adapted from EXAMPLES.DOC. What is added in this version is the ability to limit the key assignment to the active template. So, for instance, once this is installed Globally, you can assign a key sequence that will be save only in LETTER.DOT, if it is run while a document based on LETTER.DOT is active.

ChangeQuoteDash

Normally, when you type a quotation mark, WinWord inserts Character #34: "This is surrounded by normal quotation marks." And, similarly, when you insert a "single quotation mark", WinWord inserts Character #39, the Apostrophe: For instance:

"We're at the 'mercy' of our computers!" he said with chagrin.

The contraction apostrophe and the quotation marks around "mercy" are the same character.

Well, that's all you need if you are using a fixed font such as Courier or Prestige Elite. These fonts don't distinguish between Open and Close Quotation marks, However, proportionally spaced fonts, such as Times Roman, does.

Compare:

"This is 'Fixed' type quotation marks."

“This is ‘Publishing’ type quotation marks.”

WinWord provides a way to insert the Publishing quotation marks with the macro found in EXAMPLES.DOC called SmartQuotes.

However, what if you don't always want Publishing Quotation marks? And what if you started a document without them and now you do want them?

ChangeQuoteDash allows you to easily toggle between Fixed quotation marks and Publishing quotation marks. It also allows you to toggle between EMDash/ENDash and the fixed font usage of double hyphens to represent dashes.

A possible way, if you are like me, to use this macro, is before printing -- that is, enter text without SmartQuotes installed, and then when you want a pretty copy with Publishing characters, run this macro. If you don't save after printing, then the document will remain in fixed format.

By convention, ChangeQuoteDash assumes that a double hyphen has been used to represent a dash.

ChooseDirectory

Version 2.3

This version has several enhancements:

This version of ChooseDirectory has several enhancements

Foremost is the use of a ListBox (instead of an OptionGroup) to display the directory choices.

WIN.INI

The macro uses a series of keywords in your WIN.INI file under [Microsoft Word] to store alternate extensions, and path descriptions/specifications

The alternate extensions are stored in a line like the following:

xExt=*,TXT,XWS,DOT

When first run, this line will not exist. ChooseDirectory will prompt you for input. Note that the extensions are **just** that: no preceding period. Also note that the default extension (DOC) should not be entered in this line (though doing so won't harm anything).

The description/directories are stored in lines such as:

Dir 0=Main Directory|C:\WINDOWS\WINWORD

Dir1

Dir2

.
. .
.

Dir25

These lines are added as needed. By default, ChooseDirectory presents a listbox of 12 entries. This value is written to WIN.INI as **DNum=11** (computers start counting at zero...)

When first run the list box will display:

Choice 1

Choice 2

.
. .
. .

Choice 12

To install a description/path/document name for one of these "blank" choices, simply double click on the choice (or highlight it and check Edit).

One of the suggested alternate extensions is DOT (to show a directory of you DOT files and allow direct loading). If this extension is selected, you will automatically be changed to the path specified in WIN.INI next to the keyword DOT-PATH (if you don't have a DOT-PATH, an error will be returned). The point is, it is possible to tie an extension to a path. To see how that is done, examine the macro and look for the following line:

If dlg.Ext = "DOT"

Configure

Edit description - allows you to edit a directory/document description and path designation

Alternate extensions - allows you to edit the alternate extensions list

Change size of list - allows to alter the size of the list of directories.

The **Create new file** check box does precisely that: it changes to the directory pointed to by the list box choice, and then creates a new file, prompting you for the template. Note that if the list box selection is on a document, then that document is opened even if **Create new file** is checked.

CustomGlossary

One of the deficiencies, in my opinion, of the built in glossary support is that although you can define a glossary entry and store it either globally (in NORMAL.DOT) or locally (in the currently active Template), you cannot toggle between displaying only global glossaries or template glossaries. Since I use glossaries quite a lot to store scrap and boilerplate for a specific document or class of documents, I would like to show a list of available scrap glossaries without cluttering the list box with global glossaries (such as accented characters, publishing glossaries, etc.). This macro, though marginally slower than the built in function, does the trick.

The screen is rather self explanatory. You might want to double click on the Demo button now and take a peek.

You can Insert, Delete, Define, and Copy from one context to another, i.e. make a template glossary global or a global glossary local.

To toggle the display from one to the other simply click on the check box at the bottom of the dialog and click on OK.

Limitations: As of now, if you have two glossaries of the same name defined in both Global and Template context, and you attempt to delete the one in the global context, you will, in fact, delete the template version first. I'm not sure why this happens. If anyone has a clue, please let me know.

ExpandGlossary

ExpandGlossary - basically, all this does that the built in ExpandGlossary doesn't do is determine the font and fontsize *from the abbreviation*, and then apply it to the inserted glossary entry. This basically means that the glossary will take on the font attributes of the destination, not of the source (which, in my opinion, is how it should work to begin with).

· The exception is if the glossary entry is formatted in the Symbol Font, this is to that bullets will print properly. You could expand this to include Italics, or Bold if you desire a glossary entry to retain that formatting. Personally, I find it easier to turn on italics, insert the glossary, and turn it off...

Thanks to (thanks to James Gleick and Herb Tyson)

MacroKey

Current version: 1.5c

Purpose:

Allows simple assignment and removal of key stroke combinations to macros (either global or template bound).

Enhancements over the built in MacroAssignToKey:

Allows the use of Alt-Char, Alt-Ctrl-Char, and Alt-Shift-Char. In the built in routine the first two combinations are disallowed. The last is reserved for WinWord's use. Note that WinWord reserves these key sequences for it's own use in many cases, so preventing you from reassigning them is a form of protection. Use this routine cautiously, and backup all DOT files before you experiment.

Limitations:

It seems that any combination of keystroke with F1 will always call help. So it isn't fair game.

It seems that NumPad 5 (the numpad 5 that is activated by turning NumLock ON) does not accept shift characters. The NumPad 5 that is active in default state (NumLock OFF) does accept shift characters. The macro accounts for this oddity. Also note that the Unshifted NumLockOFF-5 cannot be assigned to a macro (this is the key, by the way, that is described as KeyCode 12 in the Technical Reference).

No sorting of the macros. The template macros will appear in the order you created them. The built in macros (if that option is checked) will appear in the order God created them.

How it works:

First dialog: You will be presented with a dialog allowing you to choose several options:

Global - Displays the macros from Normal.dot

Template - Displays the macros from the currently active document's template if there is one (if there isn't then "No Template" will be displayed.

Show Built in Macros - In addition to the user created macros, this option, if checked, will show all of the built in macros. Note that the macros will not be displayed sorted. This is equivalent to the Show All option available on the main screen

Do Multiple - Checking this option will loop through the macro

until you select cancel from the main screen.

Main Screen:

List box of macros from the currently selected context. Select the macro you wish to assign a key to (or remove a key from).

Key Combination: Check the shift keys you wish to activate (Alt, Ctrl, Shift, in any combination), and then move to the Key box and type the key you wish to assign to the selected macro. Legal keys are any non-shifted alphanumeric key. For instance, the semi-colon is legal, the colon is not (since the colon is already a shift state key...). Likewise, there is no difference between "D" and "d"

Fkeys - double clicking on Fkeys, or clicking on it once and then selecting OK, will display a dialog box of twelve function keys, and three check boxes for the shift states.

Special - double clicking on Special, or clicking on it once and then selecting OK, will display a dialog box of all supported special keys (Esc through Del...). Try it, you'll like it.

Reload - this option reloads the macros - useful if you have changed the checkbox states of Template or Showall.

Remove - this option checks to see if the currently selected macro has a keycode assigned to it. If so you are asked if you want to remove it. If not, a message displays at the bottom of the screen.

Template - this is a checkbox (double clicking will not execute the command; you have to check it and then click on OK). It toggles between displaying only Template Macros and only Global Macros.

ShowAll - this is a checkbox that toggles between showing macros in the selected context (global or template) and the built in macros.

Additions in this version:

Will not allow you to assign anything to Ctrl-Alt-Del (duh...)

Cleaned up the screen a little and includes the working macro on all screens

Limitations: the macro works on a document, not on a template. That is, if you are editing a template directly, MacroKeys will choke. Base a blank document on that template and try again.

MakeWordBook

This macro will first check to see if there is a block of text selected. If so, that selection will be inserted as a glossary name.

If there is no selected text (just an insertion point cursor), then this macro will take the current word

and make it a glossary name.

This macro will check for duplicate glossary names and append the instance count to the word if there is already a glossary using the word as the mark. It does not, however check for illegal punctuation marks in the current selection.

MakeBook

A slightly more complicated version of MakeWordBook, this macro takes the current selection and presents it for editing.

Instead of grabbing the current word if there is not selection it grabs the current line.

It presents the multi-word selection as a proposed bookmark, with spaces converted to underline marks

It does not check for illegal punctuation.

MergeTemplate

This macro checks to see if there is a document template attached to the currently active document. If so, it merges the paragraph/character styles as they are currently defined in the template into the document. This is useful if you have changed the template or document independent of one another, and is the equivalent of FormatDefineStyle.Options.Merge.From.TemplateName.

PrintEnvelopeHPII

This macro prints an envelope, using a hard-coded Return Address and either 1) a bookmark named NameAddress, 2) the current selection, or 3) an InputBox supplied address.

This macro works best if you have adapted the AutoOpen Macro of your letter template to automatically prompt for an address, insert the address, and mark it with a bookmark NameAddress. For instance:

```
EndOfDocument
InsertBookmark "NameAddress"
On Error Goto bye
a$ = InputBox$("Type the name of the recipient for the inside address")
Insert a$
InsertPara
FormatStyles "CurrentBody"
EditGoTo "NameAddress"
ExtendSelection Chr$(13)
FormatStyles "Address"
InsertBookmark "NameAddress"
EndOfDocument
```

Please note, this macro was written for the Hewlett Packard Laserjet Model II. It should work, as is, on all subsequent models (the IID, IIP, and III), but if you are using a Laserjet I, you will have to modify the control codes contained in the macro.

To set the macro for the HP LaserJet I, the LaserJet IIP, and the DeskJet, you must change the variable TopMargin\$ to be equal to the correct value. Below is the section of code to examine and change:

HPI\$ = "27"
HPIIP\$ = "1" I believe this is also the setting for the DeskJet
HPII\$ = "15"
'Change the line below to match your printer.
TopMargin\$ = HPII\$ the default is the LaserJet II

The following variables control the placement of the printing on the envelope. You may prefer slightly different margins.

"This is the number of spaces to indent from the left edge of the envelope before printing the addressee
AddressMargin\$ = "65"
"These are the indents for the envelopes according to size -- before printing the return address
BusinessIndent\$ = "20"
PersonalIndent\$ = "41"
SmallIndent\$ = "55"

PrintEnvelopeHPIII

A PrintEnvelope macro specific to the Laserjet III. It adds several features.

PrintRange

Version 2.1

Allows you to print a range of pages, entered separated by commas, for example:

Or print only the odd or only the even pages of a document.

ToggleHidden

Toggle View Hidden -- without going through the View Preferences dialogue box.

TogglePageView

Toggles from any view to Print Preview, then back to Page View.

ToggleRevision

Toggle revision marks on and off.

ToggleStyleBar

Toggles the style bar on and off

~~ToggleWindow~~

Toggle the currently active document window split/zoomed.

~~WinSideBySide~~

Arranges the document Windows side by side. If there are more than two active windows you are prompted for which window to arrange next to the currently active document.

~~WindowStack~~

Simply stacks the current document windows, leaving the title bars visible. There are two constants in the macro, HLap and VLap, which can be adjusted if the overlap does not suit your preferences.

The macro is not device dependent. It should work at all monitor resolutions.

~~Notices and stuff~~

Most of the macros in this package were developed as a consequence of conversations, questions, challenges, and friendships developed on the Microsoft Applications forum of CompuServe. I wish in particular to acknowledge James Gleick (who wrote several of the cleverest portions of these macros), Barry Simon, Tom Looker, and all the WinWord Gadfly Team.

You can modify these macros to your heart's content. That's why they are there. However, you cannot distribute modified versions of these macros without my express permission. Under no circumstances are these macros to be sold.

The macros are placed on a public forum. They are not (cannot be) protected. They are usually fully commented. My intention was to help others learn how *WordBASIC* can be used to customize *Word for Windows*. I request no payment. However, neither will I refuse contributions, especially if these macros are used in a business setting, or if they provide a basis for consulting and customizing for a fee.

Please contact me, for further information, at

219 East 69th Street, NYC 10021 (\$25.00)

CIS 711,71,3555

© Guy J. Gallo, 1990

Distributed as FreeWare for personal use. All corporate uses require registration with the author

~~And a legal note: This macro is provided with no warranties. It may not be published without express permission from the author.~~