version
© , 1991
CIS 71171,3555
1/18/2023

~~[Install/Demo]~~

~~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.

~~Installation~~

Clicking on the Install/Demo button above will present you with a list of all the macros contained in this document. You can choose to install a macro into the global context (normal.dot) or to a specific template.

You can only install one macro at a time.

The installation routine does **not** automatically save changes and additions. You must execute FileSaveAll when you are done and save all changes.

Note that a large NORMAL.DOT will take a long time to save... perhaps longer than you've ever seen. So be patient.

~~Word Processing~~

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 distinquish 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.

Enhancements in version 1.1: This version allows you to limit the action of the macro to a specific style or font.

CreateOutlineDoc

Installation[1]

Simply double click on the MacroButton above and choose a destination. If you want to see this document's outline levels copied into a new document, simply choose Demonstrate.

Outline Document Name

This is the name used to save the newly created outline document. By default the name will be the file name portion of the document being converted with the document extension OTL appended.

By default the OTL file will be saved in the same directory as the source document.

If you wish to change either of these defaults, type the filename desired, or the full path desired, in the edit box.

Outline Document Template

By default the new document will be based on the same template as the source document. If the source document is a template itself, then the destination document will be based on NORMAL.DOT.

If you have a standard outline template, you can enter it into the macro as the default. For example, if you have a template named OUTLINE.DOT used for all outlines, simply locate and change the line

to

Copying footnotes and annotations

Simply check the appropriate box to include footnotes and/or annotations in the new document.

1This is a footnote attached to an outline level and so it will be copied

Be aware that the footnotes numbering will not be the same as the numbering in the original document since the footnotes that are in the text are ignored.

Also, be aware that footnotes that do not have automatic numbering will be ignored.

Limitations

Since the macro forces ViewFieldCodes and ShowAll off, the result of a field in the heading level will be copied, not the field itself.

CreateOutlineDoc2

This is a simpler version of CreateOutlineDoc. This version does not display a dialog box for confirmation or editing of the default outline document name and template. And it does not copy footnotes or annotations.

EditGlossary

(Previously named CustomGlossary in Gtools version 1.0)

One of the defeciencies, 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.

Additions

Like ExpandGlossary, this version of EditGlossary macro will insert the glossary item with the formatting of the destination.

Note

By default the installation routine will copy this macro over a built in macro of the same name. If you

want to retain the original Winword macro then install this macro and after installation rename it to something else.

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 so 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)

Note

By default the installation routine will copy this macro over a built in macro of the same name. If you want to retain the original Winword macro then install this macro and after installation rename it to something else.

InsertExtSymDing

This macro was written and tested using the Symbol and ZapfDingbat fonts found in Adobe Type ManagerÔ

Purpose

InsertExtSymDing presents a list box of one of three sets of symbolic charcters. It obviates the need of typing in the Alt-0xxx ANSI codes to access the extended accents and symbols found in any Windows font; and it provides a description of the characters found in the Symbol and Dingbat font sets.

All fonts have the characters necessary to access the extended ANSI accents and symbols; Word for Windows users have, or have the option to have, the Symbol sets in 8, 10, 12, 14, 18 and 24 pts. However, access to the Dingbat character set requires that you have the Dingbat font available, either in a soft font or in one of the font generators such as ATM or Facelift.

When the macro runs you will first be presented with a list of the Extended character set. If you double click on either Symbol or Dingbat, the dialog box will reload with that character set list loaded.

If you click on *Do Multiple*, then after inserting a character, the dialog box will reappear, with the cursor position on the inserted character, ready for another insertion. This is useful for drawing lines or multiple dingbats.

If you click on the *Save Character Set*, a variable is written to you WIN.INI which allows **InsertExtDingSym** to start with the character set of you choice the next time you run it.

The *point size* edit box displays the current point size of the line into which you are inserting a character. You can manually change this value. When you resume typing, the point size is restored to the value of the line before the inserted character.

History

Version 1.1 - first public release, Dec 20, 1990

Version 1.2 - second public release, Dec 24, 1990

Incorporated a fix suggested by Frank Johnson which does a better job of resetting the character formatting than did the original version.

Changed some of the descriptions slightly.

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.

UtilSpell

This is a short macro that can be installed to replace the built in macro named UtilSpelling.

What it does

It searches the directory specified in your WIN.INI as the Util-Path for a user dictionary with the same file name as the document being checked. If such a dictionary exists it is loaded as one of the supplementary dictionaries.

What it needs

You should check your win.ini file to see that you have a keyword named Util-Path in the section defined as [Microsoft Word], and that this keyword points to the path containing the Lex-am.dll and DIC and DAT files. This is normally the same as the WinWord.exe file.

Network usage

It could be that on a network system (I don't have one) the user dictionaries will be on a different path

than the standard dictionaries and utilities. If that's the case, you can modify the macro to search *any* path for a user dictionary for the current document by changing the line

If fExist(~~GetProfileString$("Util-Path") + "\"~~ + Name$) Then

The modification entails changing the portion outlined in red above to point to the path you want, either by reading a different [Microsoft Word] keyword (such as Dot-Path), a user specified keyword that you created (such as DIC-Path), or a hard coded name (such as "C:\mydics\")

If you want this to be your standard UtilSpelling macro, simply install it and rename it UtilSpelling.

UtilSpellDocDic

This macro is similar to the modified UtilSpelling found in GTOOLS. The difference is twofold: 1) it automatically creates the DocName.DIC supplemental dictionary if it doesn't already exist, and 2) it starts spell checking immediately (bypassing the options dialog box).

If you want this to be your standard UtilSpelling macro, simply install it and rename it UtilSpelling.

~~Styles~~

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.

ApplyStyleToKey

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.

~~Printing~~

PrintRange

Version 2.1

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

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

~~Macro Writing Tools~~

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)

CopyMacro

This is a simpler macro to copy a macro. It searches both the current context (if it is a template) and the global context, to locate the named macro, and then copies it.

MacroEditTemplate

This is a simple macro that simple opens the Macro Edit dialog box and automatically switches the context to Template.

ManageMacros

This macro presents you with a list box of all the macros in the global context and all of the macros in the template context, and allows you to easily move a macro from one list to the other.

Be aware that this macro takes some time.

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.

MergeMacros

Version 1.4

This is an extremely useful macro. It allows you to copy the macros from one template to another, to create a document that contains all or selected macros from a given template, and allows you to copy macros from a document into a template.

MergeTemplate assumes that you want to move a macro into the currently active template. So when it is firt invoked it prompts you to select a source template.

The dialogue box you will see is precisely the same as the one that you see when you open a document. Select the template that holds the macros you wish to move.

A dialog box then appears allowing you four option buttons and once checkbox:

MergeMacros into: TemplateName

This option will open the source template and move each of it's macros into the template specified as TemplateName. By default this is NORMAL.DOT if no template is attached (meaning you are in a NORMAL.DOT based document or you are in a template).

Select new destination template

This option allows you to change the destination template.

Create Macro Document

This option does not copy the macros within the SourceTemplate. Rather it creates a "Macro Document" of the following format:
MacroName -- formatted as Heading 1
MacroText -- formatted as normal.
Do not change these formats (or add text) if you wish to later use the next option.

Paste From Macro Document

This option assumes that the active document when it is invoked is a macro document created with the previous step. What is does, quite simply, is paste each macro from the document into the template. This **does** require that you have selected the appropriate destination. The macros are deleted from the macro document as they are pasted. But the macro document will not be saved in this emptied state.

The check box is

Confirm each

Just what it sounds like. Each macro is presented for confirmation for copying or pasting.

Limitations:

The source template cannot be Normal.dot. To move macros from Normal.dot to another template use the MacroManage

ShowKeys

Version 1.1 (a bug in version 1.0 made the macro crash at certain key codes... I think this one works).

This is a relatively (relative to MacroKey) simple macro that displays the current key assignments -- either in the global or template context .

It can create a new document and actually insert the key assignemts -- or it will simply display them on the status bar...

When first invoked a message box appears asking if you want to examine Global key assignments. Answering No will show template assignments.

The next message box asks whether or not you wish to display the key assignments on the status bar or create a document....

~~Toggles~~

ToggleHidden

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

ToggleOrientation

This is a macro that will look at the current page width -- see if it is 8.5 or 11 inches, and toggle to the opposite orientation -- if the current page is 8.5 it will set the printer for Landscape and reformat the document as 11x8.5. If the current setting is 11 inches, it will toggle the printer to Portrait and reformat the document as 8.5x11.

The only assumption made by the macro is that Alt-R means Portrait and Alt-L means landscape in the Setup box of you printer. The newer printer drivers all seem to be standardizing on this.

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.

Window Arrangement

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*.

Please contact me, for further information, at

or via modem at

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

Registration

Click
Corporate site licenses are available. Please contact the author at the address below and a fee will be negotiated for the specific macro and use involved.

219 East 69th Street, NYC 10021

And a legal note: This collection of Word for Windows macros is provided with no warranties. It may not be published without express permission from the author.

ApplyMacroToKey
ApplyStyleToKey
ChangeQuoteDash
CreateOutlineDoc
CreateOutlineDoc2
CopyMacro
EditGlossary
ExpandGlossary
InsertExtSymDing
MacroEditTemplate
MacroKey
MakeBook
MakeWordBook
ManageMacros
MergeMacros
MergeTemplate
PrintRange
ShowKeys
ToggleHidden
ToggleOrientation
TogglePageView
ToggleRevision
ToggleStyleBar
ToggleWindow
UtilSpell
UtilSpellDocDic
WindowStack
WinSideBySide

Gadfly Macros Registration -

Guy Gallo 219 East 69th New York, NY 10021

Registration fee: $25.00

NYS Sales Tax (if appropriate)

Total enclosed:

NAME:

COMPANY:

STREET:

CITY:

STATE, ZIP:

Electronic address:

Where did you get ?

COMMENTS: