#$ About

Description Displays the WinHelp application's About topic.

Syntax **About**()

MACRO_About

$ About Macro

#$ AddAccelerator (AA)

Description Assigns a help macro to an accelerator key or key or key combination so that the macro is executed when the accelerator key is pressed.

Syntax **AddAccelerator**(*key, shift-state, `macro'*)

MACRO_AddAccelerator

$ AddAccelerator (AA)

#$ Annotate

Description Displays the WinHelp application's Annotate dialog box, as if the user had selected **Annotate** from the **Edit** menu.

Syntax **Annotate**()

MACRO_Annotate

$ Annotate

[#][$] AppendItem

Description Appends a menu item to the end of an existing menu - either one of the default menus or one created with the **InsertMenu** macro.

Syntax **AppendItem**(*`menu-id', `item-name', `macro'*)

[#] MACRO_AppendItem

[$] AppendItem

#$ Back

Description Displays the previous topic in the history list. This is a list of the topics displayed since starting WinHelp. Topics displayed in popup windows are not included in this list. The function of this macro is identical to selecting the **Back** button.

Syntax **Back**()

MACRO_Back

$ Back

[#][$] BookmarkDefine

Description Displays the Bookmark Define dialog box. Executing this macro is the same as selecting **Define** from the **Bookmark** menu.

Syntax **BookmarkDefine**()

[#] MACRO_BookmarkDefine

[$] BookmarkDefine

[#][$] BookmarkMore

Description Displays the Bookmark dialog box. Executing this macro is the same as selecting **More** from the **Bookmark** menu. The **More** item appears on the **Bookmark** menu if there are more than nine bookmarks defined.

Syntax **BookmarkMore**()

[#] MACRO_BookmarkMore

[$] BookmarkMore

#$ BrowseButtons

Description Adds the browse buttons, **<<** and **>>,** to the WinHelp button bar.

Syntax **BrowseButtons**()

MACRO_BrowseButtons

$ BrowseButtons

[#][$] ChangeButtonBinding (CBB)

Description Assigns a new macro or macro string to a help button, replacing the button's previously-defined action.

Syntax **ChangeButtonBinding**(`*button-id', `macro'*)

[#] MACRO_ChangeButtonBinding

[$] ChangeButtonBinding (CBB)

ChangeItemBinding (CIB)

Description Assigns a new macro or macro string to a help menu item, replacing the item's previously-defined action.

Syntax **ChangeItemBinding**(`*item-id', `macro'*)

MACRO_ChangeItemBinding

$ ChangeItemBinding (CIB)

$ CheckItem (CI)

Description Places a check-mark beside a help menu item.

Syntax **CheckItem**(`*menu-id*')

MACRO_CheckItem

$ CheckItem (CI)

$ CloseWindow

Description Closes either a secondary window or the main help window.

Syntax **CloseWindow**(`*window-name'*)

MACRO_CloseWindow

$ CloseWindow

#$ Command

Description Executes a WinHelp menu command, based on the command number passed to it.

Syntax **Command**(*command-number*)

#	MACRO_Command
$	Command

#$ Contents

Description Displays the Contents topic in the current help file. The Contents topic is defined by the CONTENTS option in the [OPTIONS] section of the help project file. If no Contents topic is defined, the Contents topic defaults to the first topic in the first topic file specified in the [FILES] section of the help project file.

Syntax **Contents**()

MACRO_Contents

$ Contents

#$ CopyDialog

Description Displays the Copy dialog box. Copies the text from the current topic into the Copy dialog box where portions of the text may be selected and copied to the Windows Clipboard. Executing this macro is the same as selecting **Copy** from the **Edit** menu.

Syntax **CopyDialog**()

MACRO_CopyDialog

$ CopyDialog

#$ CopyTopic

Description Copies all of the text in the currently displayed topic to the Clipboard. Executing this macro is the same as pressing **Ctrl**+**Ins** in the main help window.

Syntax **CopyTopic**()

MACRO_CopyTopic

$ CopyTopic

$ CreateButton

Description Creates a button and adds it to the button bar.

Syntax **CreateButton**(`*button-id', `name', `macro'*)

MACRO_CreateButton

$ CreateButton

#$ DeleteItem

Description Removes an item from a WinHelp menu.

Syntax **DeleteItem**(`*item-id*')

MACRO_DeleteItem

$ DeleteItem

[#][$] DeleteMark

Description Removes a text marker that was previously added with the **SaveMark** macro.

Syntax **DeleteMark**(`*marker-text'*)

[#] MACRO_DeleteMark

[$] DeleteMark

$ DestroyButton

Description Removes a button that was previously added with the **CreateButton** macro.

Syntax **DestroyButton**(`*button-id*')

MACRO_DestroyButton

$ DestroyButton

$ DisableButton (DB)

Description Disables (grays-out) a button on the WinHelp button bar. The disabled button will remain inoperative until it is re-enabled with the **EnableButton** macro.

Syntax **DisableButton**(`*button-id*')

MACRO_DisableButton

$ DisableButton (DB)

#$ DisableItem (DI)

Description Disables (grays out) a WinHelp menu item. The item will remain disabled until it is re-activated with **EnableItem** or **ExtAbleItem**, or until the menu is reset using **ResetMenu**.

Syntax **DisableItem**(`*item-id*')

MACRO_DisableItem

$ DisableItem (DI)

[#][$] EnableButton (EB)

Description Re-enables a button that was previously disabled with the **DisableButton** macro.

Syntax **EnableButton**(`*button-id*')

[#] MACRO_EnableButton

[$] EnableButton (EB)

#$ EnableItem (EI)

Description Re-enables a menu item that was previously disabled with **DisableItem** or **ExtAbleItem**.

Syntax **EnableItem**(`*item-id'*)

MACRO_EnableItem

$ EnableItem (EI)

#$ ExecProgram (EP)

Description Executes an application.

Syntax **ExecProgram**(`*command-line', display-state*)

MACRO_ExecProgram

$ ExecProgram (EP)

#$ Exit

Description Exits the WinHelp application. The action of this macro is identical to selecting **Exit** on the **File** menu.

Syntax **Exit**()

MACRO_Exit

$ Exit

[#]# [$]$ ExtAbleItem

Description Enables or disables a specified menu item.

Syntax **ExtAbleItem**(`*item-id', enabled-state*)

[#] MACRO_ExtAbleItem

[$] ExtAbleItem

[#][$] ExtInsertItem

Description Inserts an item into a WinHelp menu and allows the initial enabled state (enabled or disabled) to be specified.

Syntax **ExtInsertItem**(`*menu-id*',`*item-id*', `*item-name*',`*macro*',*position*,*enabled-state*)

[#] MACRO_ExtInsertItem

[$] ExtInsertItem

[#]# [$]$ ExtInsertMenu

Description Inserts a sub-menu as an item in a previously-defined menu, and allows the initial enabled state (enabled or disabled) to be specified.

Syntax **ExtInsertMenu**(*`parent-id',`menu-id',`menu-name',position,enabled-state*)

[#] MACRO_ExtInsertMenu

[$] ExtInsertMenu

[#]# [$]$ FileOpen

Description Displays the Open dialog box from the **File** menu.

Syntax **FileOpen()**

[#] MACRO_FileOpen

[$] FileOpen

[#]# [$]$ FloatingMenu

Description Displays the floating menu at the current mouse cursor position.

Syntax **FloatingMenu**()

[#] MACRO_FloatingMenu

[$] FloatingMenu

[#] [$] FocusWindow

Description Changes the focus to the specified window - either the main help window, or a secondary window.

Syntax **FocusWindow**(`*window-name'*)

[#] MACRO_FocusWindow

[$] FocusWindow

[#][$] GotoMark

Description Jumps to a text marker that was previously set with the **SaveMark** macro.

Syntax **GotoMark**(`*marker-text*')

[#] MACRO_GotoMark

[$] GotoMark

[#] [$] HelpOn

Description Displays the help file for the WinHelp application. Executing this macro is the same as selecting **How to Use Help** from the **Help** menu.

Syntax **HelpOn**()

[#] MACRO_HelpOn

[$] HelpOn

#$ HelpOnTop

Description Toggles the on-top state of WinHelp, checking or un-checking the **Always on Top** menu item as required. Executing this macro is equivalent to selecting **Always on Top** from the **Help** menu.

Syntax **HelpOnTop**()

MACRO_HelpOnTop

$ HelpOnTop

#$ History

Description Displays the WinHelp History window, which shows the titles of the last 40 topics that have been displayed since WinHelp was started. Executing this macro is the same as selecting the **History** button.

Syntax **History**()

MACRO_History

$ History

[#] [$] IfThen

Description Executes the specified *macro* if the *condition* is true.

Syntax **IfThen**(*condition, `macro'*)

[#] MACRO_IfThen

[$] IfThen

[#] [$] IfThenElse

Description Executes the specified *macro1* if the *condition* is true, and *macro2* if the *condition* is false.

Syntax **IfThenElse**(*condition, `macro1', `macro2'*)

[#] MACRO_IfThenElse

[$] IfThenElse

[#][$] InsertItem

Description Inserts an item in a specified position in a WinHelp menu.

Syntax **InsertItem**(`*menu-id*',`*item-id*', `*item-name*',`*macro*',*position*)

[#] MACRO_InsertItem

[$] InsertItem

#$ InsertMenu

Description Inserts a menu in a specified position on the WinHelp menu bar.

Syntax **InsertMenu**(`*menu-id*',`*menu-name*',*position*)

MACRO_InsertMenu

$ InsertMenu

[#][$] IsMark

Description Determines whether or not a particular text marker exists. Returns 1 if it does, or 0 if it does not.

Syntax **IsMark**(`*marker-text*')

[#] MACRO_IsMark

[$] IsMark

#$ JumpContents

Description Executes a jump to the Contents topic of the specified help file. Executing this macro is equivalent to opening a new help file by choosing **Open** from the **File** menu.

Syntax **JumpContents**(`*filename'*)

MACRO_JumpContents

$ JumpContents

$ JumpContext (JC)

Description Executes a jump to the topic in the specified help file that corresponds to the specified context number. Context numbers are assigned in the [MAP] section of the help project file.

Syntax **JumpContext**(`*filename', context-number*)

MACRO_JumpContext

$ JumpContext (JC)

[#] [$] JumpHash

Description Executes a jump to the topic in the specified help file that corresponds to the specified hash code.

Syntax **JumpHash**(`*filename', hash-code*)

[#] MACRO_JumpHash

[$] JumpHash

[#][$] JumpHelpOn

Description Displays the help file for the WinHelp application. Executing this macro is the same as selecting **How to Use Help** from the **Help** menu.

Syntax **JumpHelpOn**()

[#] MACRO_JumpHelpOn

[$] JumpHelpOn

[#][$] JumpId (JI)

Description Executes a jump to the topic in the specified help file that has the specified context string.

Syntax **JumpId**(`*filename', `context-string')*

[#] MACRO_JumpId

[$] JumpId (JI)

[#][$] JumpKeyword (JK)

Description Searches the keyword table of the specified help file and displays the first topic that matches the specified keyword.

Syntax **JumpKeyword**(`*filename', `keyword')*

[#] MACRO_JumpKeyword

[$] JumpKeyword (JK)

#$ Next

Description Displays the next topic in the current browse sequence. Executing this macro is the same as selecting the forward browse button (**>>**).

Syntax **Next**()

MACRO_Next

$ Next

[#] [$] Not

Description Reverses the result (non-zero or zero) returned by a conditional macro such as **IsMark**.

Syntax **Not**(`*condition'*)

[#] MACRO_Not

[$] Not

[#][$] PopupContext (PC)

Description Displays, in a popup window, the topic in the specified help file that corresponds to the specified context number. Context numbers are assigned in the [MAP] section of the help project file.

Syntax **PopupContext**(`*filename', context-number*)

[#] MACRO_PopupContext

[$] PopupContext (PC)

[#]# [$]$ PopupHash

Description Displays, in a popup window, the topic in the specified help file that corresponds to the specified hash code.

Syntax **PopupHash**(`*filename', hash-code*)

[#] MACRO_PopupHash

[$] PopupHash

#$ PopupId (PI)

Description Displays, in a popup window, the topic in the specified help file that has the specified context string.

Syntax **PopupId**(*`filename', `context-string')*

MACRO_PopupId

$ PopupId (PI)

PositionWindow (PW)

Description Sets the size, position, and display state of either the main help window, or a secondary window.

Syntax **PositionWindow**(*x, y, width, height, state, `name')*

MACRO_PositionWindow

PositionWindow (PW)

#$ Prev

Description Displays the previous topic in the current browse sequence. Executing this macro is the same as selecting the backward browse button (**<<**).

Syntax **Prev**()

MACRO_Prev

$ Prev

[#] [$] Print

Description Prints the currently displayed topic to the printer. Executing this macro is equivalent to selecting **Print Topic** from the **File** menu.

Syntax **Print**()

[#] MACRO_Print

[$] Print

[#][$] PrinterSetup

Description Displays the Print Setup dialog box. Executing this macro is equivalent to selecting **Print Setup** from the **File** menu.

Syntax **PrinterSetup**()

[#] MACRO_PrinterSetup

[$] PrinterSetup

[#] [$] RegisterRoutine (RR)

Description Registers a function within a DLL as a custom help macro. The custom help macro may then be used as are the standard Windows help macros.

Syntax **RegisterRoutine**(*`DLL-name', `function-name', `format-spec')*

[#] MACRO_RegisterRoutine

[$] RegisterRoutine (RR)

$ RemoveAccelerator (RA)

Description Removes an accelerator key or key combination that was previously assigned with the **AddAccelerator** macro.

Syntax **RemoveAccelerator**(*key, shift-state*)

MACRO_RemoveAccelerator

$ RemoveAccelerator (RA)

#$ ResetMenu

Description Returns the WinHelp menu bar and all popup menus to their defaults.

Syntax **ResetMenu**()

MACRO_ResetMenu

$ ResetMenu

[#]# [$]$ SaveMark

Description Saves the location of the currently displayed topic and file, and associates a text marker with that location. Text markers are used by the **GoToMark** and **IsMark** macros.

Syntax **SaveMark**(`*marker-text*')

[#] MACRO_SaveMark

[$] SaveMark

#$ Search

Description Displays the Search dialog box. Executing this macro is equivalent to selecting the **Search** button on the WinHelp button bar.

Syntax **Search**()

MACRO_Search

$ Search

[#][$] SetContents

Description Designates a specific topic as the Contents topic in the specified help file.

Syntax **SetContents**(“filename”, context-number)

[#] MACRO_SetContents

[$] SetContents

[#][$] SetHelpOnFile

Description Specifies the name of the replacement How to Use Help file

Syntax **SetHelpOnFile**(“filename”)

[#] MACRO_SetHelpOnFile

[$] SetHelpOnFile

#$ UncheckItem (UI)

Description Removes the check-mark that was previously placed beside a help menu item by the **CheckItem** macro.

Syntax **UncheckItem**(`*menu-id'*)

MACRO_UncheckItem

$ UncheckItem (UI)