The Windows Help Style Guide

About

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Help Basics

Fonts

Colors

Design & Layout

Bullets

Tables

Secondary Windows

Non-Scrolling Regions

Macros

Help Source Files

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Help Basics

What is Windows Help?

Windows Help is a system-wide standard for developing online help and its installed with every copy of Windows. It consists of a Help viewer and a Help compiler for developers. Windows Help usually called WinHelp is a standalone Windows application. The WinHelp executable ((WINHELP.EXE) uses compiled Help files (.HLP) only. WinHelp applications can be linked to another Windows application or can be used standalone as information systems.

Advantages of Online Help

Windows Help has become a very popular way of presenting reference information to a Windows user. Windows users are notorious for not using manuals: they expect to find answers to all their questions via context-sensitive help in their Windows applications. Window Help reduces the cost of goods, updates, and distribution. It adds flexibility to the product development cycle; its faster to update than a manual. Its instantly available to the user and offers the potential of almost unlimited color graphics and multimedia capability. The library of Help topics.also allows users to quickly access the information needed.

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Fonts

In a Help system you can use different fonts to emphasize text. Windows comes with several standard fonts that are installed on every Windows system. For Help systems it is best to use standard fonts - if you use a font that isn't on the Help user's system, the result can be strange looking (difficult to read) text. Sans serif fonts are best for help systems as they tend to have a crisper display on-screen.

The Arial font was used to create this Help system. Other standard Windows 3.1 fonts that the Help compiler supports include:

TrueType:

Courier New

Σψμβολ (Symbol)

Times New Roman

Standard:

Courier

MS Sans Serif

MS Serif

Small Fonts

You can also add font characteristics such as:

bold

italic

underline

~~strikethrough~~

bold italic

~~etc.~~

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Colors

This topic shows the colors that were defined as part of the help system. You can use any of the colors available to your system just by selecting a color for the font. When the help system is compiled, the text appears in the specified color.

In most cases, the default color is black, although it can be any color. Other colors available to Windows Help are:

Blue	Cyan	Green	Magenta
Red	Yellow	(white)	Dark Blue
Dark Cyan	Dark Green	Dark Magenta	Dark Red
Dark Yellow	Dark Gray	Gray	

As you can see, some of the lighter colors don't show up well on the white background. It is helpful to keep this in mind when you are selecting the colors for your help system.

The Windows Help Style Guide

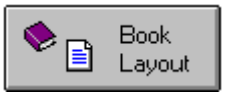
You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Design & Layout

This guide demonstrates some of the design schemes that you can use in a Windows Help system. This section illustrates just one way of organizing information for the user - using a hierarchical format with expandable/collapsible views. Below are graphical buttons - with associated jump text - that the user can select to see detailed information.

If you have the source code version of this guide, you can use this document as a template for your own help systems. For information on how to request your free copy of source code, select the About button (above).

Expandable/Collapsible views



Book Layout

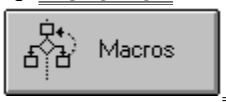
Folder Layout

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Book Layout

Overview



About

Contents

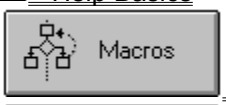
The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Book Layout

Overview

Help Basics



Help Source Files

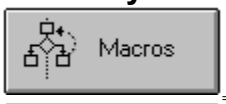
About

Contents

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Book Layout



Overview

About

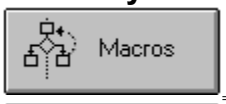
About this Help System

Contents

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Book Layout



Overview

About

Contents

Fonts

Colors

Bullets

Tables

Secondary Windows

Non-Scrolling Regions

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Folder Layout

Overview

About

Contents

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Folder Layout

Overview

Help Basics

Help Source Files

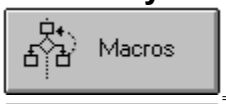
About

Contents

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Folder Layout



Overview

About

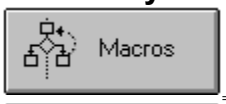
About this Help System

Contents

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Folder Layout



Overview

About

Contents

Fonts

Colors

Bullets

Tables

Secondary Windows

Non-Scrolling Regions

Help Basics

What is Windows Help?

When Microsoft developed Windows they saw the need for a system-wide standard for developing online help. The result was Windows Help. It consists of a Help viewer that is installed with every copy of Windows and a Help compiler (for developers) which follows a strict set of construction rules.

- Windows Help (also known as WinHelp) is a standalone Windows application.
- The WinHelp (WINHELP.EXE) executable file uses compiled Help files (.HLP).
- WinHelp is included with every copy of Windows 3.
- WinHelp applications can be linked to another application or can stand alone for information delivery.

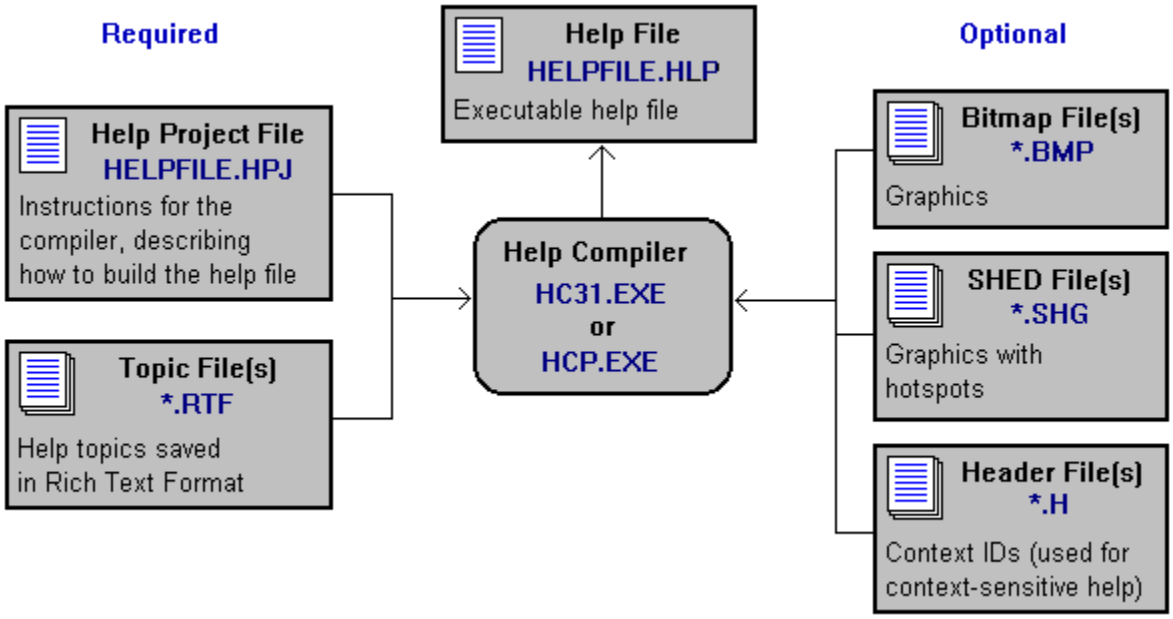
Advantages of Online Help

Windows Help has become one of the most (if not the most) popular ways of presenting reference information to a Windows user. Some of the reasons for this include:

- Reduces cost of goods, updates, and distribution.
- Adds flexibility to the product development cycle by avoiding delays required by printing.
- Eliminates unmanageable full-shelf manual sets for complex products.
- Is always available versus manuals that disappear or are not provided to all users at a site.
- Offers the potential of almost unlimited color graphics and multimedia capability.
- Provides efficient means of access to the library of Help topics.
- Integrates the process of using the product and learning about the product.

Help Source Files

A Windows help system is comprised of the following files:



Fonts

In a Help system you can use different fonts to emphasize text. Windows comes with several standard fonts that are installed on every Windows system. For Help systems it is best to use standard fonts - if you use a font that isn't on the Help user's system, the result can be strange looking (difficult to read) text. Sans serif fonts are best for help systems as they tend to have a crisper display on-screen.

The Arial font was used to create this Help system. Other standard Windows 3.1 fonts that the Help compiler supports include:

TrueType:

Courier New

Σψμβολ (Symbol)

Times New Roman

Standard:

Courier

MS Sans Serif

MS Serif

Small Fonts

You can also add font characteristics such as:

bold

italic

underline

~~strikethrough~~

bold italic

~~etc.~~

Colors

This topic shows the colors that were defined as part of the help system. You can use any of the colors available to your system just by selecting a color for the font. When the help system is compiled, the text appears in the specified color.

In most cases, the default color is black, although it can be any color. Other colors available to Windows Help are:

Blue	Cyan	Green	Magenta
Red	Yellow	(white)	Dark Blue
Dark Cyan	Dark Green	Dark Magenta	Dark Red
Dark Yellow	Dark Gray	Gray	

As you can see, some of the lighter colors don't show up well on the white background. It is helpful to keep this in mind when you are selecting the colors for your help system.

Bullets

Bullets are a common type of formatting in printed documentation. Most often used to set a list apart from the surrounding text, they can add a distinctive look to your work. The Windows Help compiler recognizes some types of bullets, but not all. After choosing your bullet, you may want to do a quick compile and check to verify the bullet shows in Help.

There are two ways to incorporate bullets into your help systems:

- Place a bullet character, such as a symbol, into the text.

OR

- Create a bitmap, and place it in the Help system as a character. This approach allows you to create custom bullets.

You can use any type of bullet (or bitmap):

- such as a diamond

OR

- a square

Tables

You can create several types of special effects in your help system using tables. One example is the Contents topic of this help system - a table was used so that different spacing (above) could be used for the icons and the text.

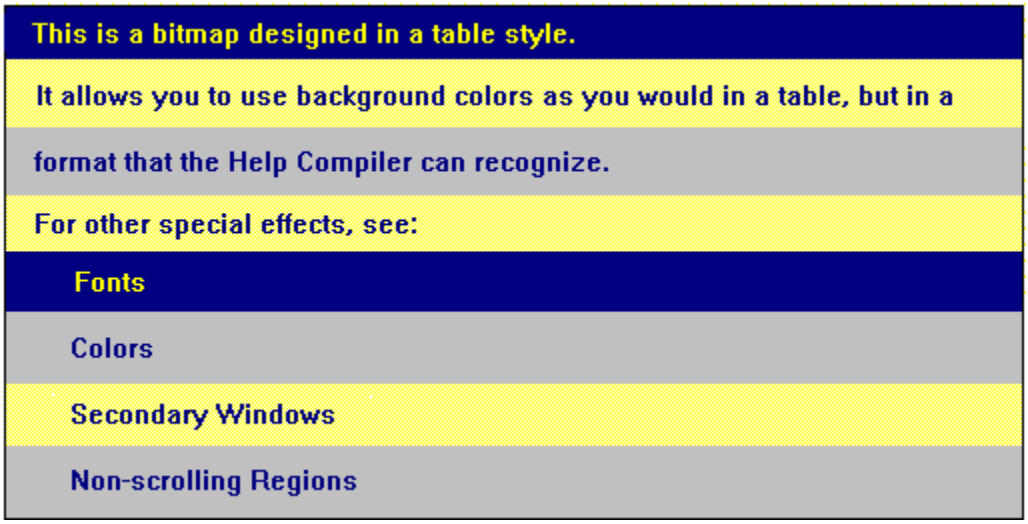
An important consideration when using tables is that the text wrap is determined by the table width, not the Help Viewer. In normal text, the Viewer automatically wraps the text when you resize the window - this doesn't happen if you use tables. Therefore, tables are best used for small bits of text.

Table Tricks

In the example below, the text overlays a graphic. This was accomplished by using a table with a narrow left column (too small for the graphic):

Overlaying Text

You can add colors to create a table-like effect by inserting a pre-built bitmap into the help system:



Secondary Windows

By default your help text displays in the "main" window. You can add "secondary" windows that display in addition to your main window. Either type of window can be positioned anywhere on the screen, and you can set the windows to be "on top" - covering all other open windows. A window is just a place where information is displayed - when you tell the help system to display a topic, you can also tell it which window is to contain the text. (Note that secondary windows are different than popups - see the samples, below.)

Non-Scrolling Regions

In most topics in this Help System, the area at the top is separate from the rest - if you scroll the text, this upper section does not move. This is a non-scrolling region.

Some rules for non-scrolling regions:

- non-scrolling regions are generally used for the topic title
- they must always come before any scrolling regions

To define a non-scrolling region, you just format the paragraph as "keep with next."

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Bullets

Bullets are a common type of formatting in printed documentation. Most often used to set a list apart from the surrounding text, they can add a distinctive look to your work. The Windows Help compiler recognizes some types of bullets, but not all. After choosing your bullet, you may want to do a quick compile and check to verify the bullet shows in Help.

There are two ways to incorporate bullets into your help systems:

- Place a bullet character, such as a symbol, into the text.

OR

Create a bitmap, and place it in the Help system as a character. This approach allows you to create custom bullets.
You can use any type of bullet (or bitmap):

such as a diamond

OR

a square

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Tables

You can create several types of special effects in your help system using tables. One example is the Contents topic of this help system - a table was used so that different spacing (above) could be used for the icons and the text.

An important consideration when using tables is that the text wrap is determined by the table width, not the Help Viewer. In normal text, the Viewer automatically wraps the text when you resize the window - this doesn't happen if you use tables. Therefore, tables are best used for small bits of text.

Table Tricks

In the example below, the text overlays a graphic. This was accomplished by using a table with a narrow left column (too small for the graphic):

Overlaying Text

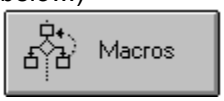
You can add colors to create a table-like effect by inserting a pre-built bitmap into the help system:

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Secondary Windows

By default your help text displays in the "main" window. You can add "secondary" windows that display in addition to your main window. Either type of window can be positioned anywhere on the screen, and you can set the windows to be "on top" - covering all other open windows. A window is just a place where information is displayed - when you tell the help system to display a topic, you can also tell it which window is to contain the text. (Note that secondary windows are different than popups - see the samples, below.)



Display in Secondary Window

You can display text using the main window, a secondary window, or a popup.

Secondary windows are movable and sizable, allowing you to place them anywhere on the screen.

They also have other typical features of windows, such as the:

- control box
- minimize button
- maximize button
- restore button

If the text is long enough, when you display it in a secondary window, you will see scroll bars. Note that the non-scrolling region at the top stays static (does not scroll with the text in the client area of the window).

Display in Popup

You can display text using the main window, a secondary window, or a popup.

Popup windows are not movable or sizable - the size and placement are determined by the Windows Help Viewer.

Popup windows do not support the use of non-scrolling regions.

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Non-Scrolling Regions

In most topics in this Help System, the area at the top is separate from the rest - if you scroll the text, this upper section does not move. This is a non-scrolling region.

Some rules for non-scrolling regions:

- non-scrolling regions are generally used for the topic title
- they must always come before any scrolling regions

To define a non-scrolling region, you just format the paragraph as "keep with next."

Non-Scrolling Region

This is an example of a topic with a non-scrolling region.

The title ("Non-Scrolling Region") was formatted as "keep with next" so that it would display in the non-scrolling region of the secondary window.

When you use the scroll button to scroll text up or down, the non-scrolling region stays static.

You can add also use non-scrolling regions to add color to your help system. You do this by defining the non-scrolling region of each window to have a different color, as in this help system.

Note that popup windows so not have the non-scrolling capability. If you use a popup to display a topic that has a "keep with next" paragraph format, only that paragraph will display (you lose everything after the non-scrolling region).

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Macros

You can use macros to tailor the behavior of your help system. For example, you could include a bitmap of a Close button that, when selected, executes the macro **CloseWindow**.

To see a list of available macros in each category, select the folder or the category name. Then, to see the macro definition, select the macro name in the list.

Macros are available for the following types of tasks:

Menu Manipulation

Button Manipulation

Navigation

Conditional Control

Dialog Control

Window Control

Miscellaneous

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how. This guide was created with RoboHELP®.

Macros

You can use macros to tailor the behavior of your help system. For example, you could include a bitmap of a Close button that, when selected, executes the macro CloseWindow.

To see a list of available macros in each category, select the folder or the category name. Then, to see the macro definition, select the macro name in the list.

Macros are available for the following types of tasks:

Menu Manipulation

Menu manipulation macros enable you to add, delete, change, disable, and enable WinHelp menus and menu items.

AppendItem

ChangeItemBinding

CheckItem

DeleteItem

DisableItem

EnableItem

ExtAbleItem

ExtInsertItem

ExtInsertMenu

FloatingMenu

InsertItem

InsertMenu

ResetMenu

UncheckItem

Button Manipulation

Navigation

Conditional Control

Dialog Control

Window Control

Miscellaneous

For further information on WinHelp macros, refer to the *WinHelp Macro Authoring Guide* - for contact information, choose the About button.

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Macros

You can use macros to tailor the behavior of your help system. For example, you could include a bitmap of a Close button that, when selected, executes the macro CloseWindow.

To see a list of available macros in each category, select the folder or the category name. Then, to see the macro definition, select the macro name in the list.

Macros are available for the following types of tasks:

Menu Manipulation

Button Manipulation

Button manipulation macros allow you to add, delete, change the function of, enable, or disable buttons on the WinHelp button bar.

BrowseButtons

ChangeButtonBinding

CreateButton

DestroyButton

DisableButton

EnableButton

Navigation

Conditional Control

Dialog Control

Window Control

Miscellaneous

For further information on WinHelp macros, refer to the *WinHelp Macro Authoring Guide* - for contact information, choose the About button.

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Macros

You can use macros to tailor the behavior of your help system. For example, you could include a bitmap of a Close button that, when selected, executes the macro CloseWindow.

To see a list of available macros in each category, select the folder or the category name. Then, to see the macro definition, select the macro name in the list.

Macros are available for the following types of tasks:

Menu Manipulation

Button Manipulation

Navigation

Navigation macros enable you to control the sequence in which topics are displayed.

Back

Contents

GoToMark

JumpContents

JumpContext

JumpHash

JumpHelpOn

JumpId

JumpKeyword

Next

PopupContext

PopupHash

PopupId

Prev

Conditional Control

Dialog Control

Window Control

Miscellaneous

For further information on WinHelp macros, refer to the *WinHelp Macro Authoring Guide* - for contact information, choose the About button.

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Macros

You can use macros to tailor the behavior of your help system. For example, you could include a bitmap of a Close button that, when selected, executes the macro CloseWindow.

To see a list of available macros in each category, select the folder or the category name. Then, to see the macro definition, select the macro name in the list.

Macros are available for the following types of tasks:

Menu Manipulation

Button Manipulation

Navigation

Conditional Control

Conditional control macros allow you to set and delete text markers, and execute other macros based on the existence (or nonexistence) of a specific text marker.

DeleteMark

IfThen

IfThenElse

IsMark

Not

SaveMark

Dialog Control

Window Control

Miscellaneous

For further information on WinHelp macros, refer to the *WinHelp Macro Authoring Guide* - for contact information, choose the About button.

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Macros

You can use macros to tailor the behavior of your help system. For example, you could include a bitmap of a Close button that, when selected, executes the macro CloseWindow.

To see a list of available macros in each category, select the folder or the category name. Then, to see the macro definition, select the macro name in the list.

Macros are available for the following types of tasks:

Menu Manipulation

Button Manipulation

Navigation

Conditional Control

Dialog Control

The dialog control macros allow you to display WinHelp dialog boxes in response to the user's input.

About

Annotate

BookmarkDefine

BookmarkMore

CopyDialog

FileOpen

History

PrinterSetup

Search

Window Control

Miscellaneous

For further information on WinHelp macros, refer to the *WinHelp Macro Authoring Guide* - for contact information, choose the About button.

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.

This guide was created with RoboHELP®.

Macros

You can use macros to tailor the behavior of your help system. For example, you could include a bitmap of a Close button that, when selected, executes the macro CloseWindow.

To see a list of available macros in each category, select the folder or the category name. Then, to see the macro definition, select the macro name in the list.

Macros are available for the following types of tasks:

Menu Manipulation

Button Manipulation

Navigation

Conditional Control

Dialog Control

Window Control

The window control macros allow you to move, size, and change the state of WinHelp windows.

CloseWindow

FocusWindow

HelpOn

HelpOnTop

PositionWindow

Miscellaneous

For further information on WinHelp macros, refer to the *WinHelp Macro Authoring Guide* - for contact information, choose the About button.

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Macros

You can use macros to tailor the behavior of your help system. For example, you could include a bitmap of a Close button that, when selected, executes the macro CloseWindow.

To see a list of available macros in each category, select the folder or the category name. Then, to see the macro definition, select the macro name in the list.

Macros are available for the following types of tasks:

Menu Manipulation

Button Manipulation

Navigation

Conditional Control

Dialog Control

Window Control

Miscellaneous

Macros are available for such tasks as copying and printing help topics, executing programs, and exiting WinHelp.

AddAccelerator

Command

CopyTopic

ExecProgram

Exit

Print

RegisterRoutine

RemoveAccelerator

SetContents

SetHelpOnFile

For further information on WinHelp macros, refer to the *WinHelp Macro Authoring Guide* - for contact information, choose the About button.

The Windows Help Style Guide

You can easily create a help system with the features demonstrated here - this guide shows you how.
This guide was created with RoboHELP®.

Help Source Files

A Windows help system is comprised of the following files:

About the Windows Help Authoring Guide

RoboHELP

The Help Authoring Guide was built with RoboHELP®, the Ultimate Help Authoring Tool for Windows and Windows NT. RoboHELP allows you to design, test and generate on-line, context sensitive hypertext Help systems for Windows. It offers document to Help system conversion and vice versa, generates the RTF, HPJ and H files, and supports both Word 2.0 and Word 6.0, and all advanced features of the Windows Help Engine.

RoboButton VBX

RoboButton - VBX Help button which eliminates the need to program context sensitive Help into your application. (You get it for free with RoboHELP.)

Blue Sky Software Corporation
7486 La Jolla Blvd., Suite 3
La Jolla, CA 92037 USA
1-800-677-4WIN
Tel: (619) 459-6365 Fax: (619) 459-6366

Form

Smart Development Tools for Windows

7486 La Jolla Blvd., Suite 3
La Jolla, CA 92037-9583, USA
1-800-677-4WIN
Tel: (619) 459-6365
Fax: (619) 459-6366

Fax Back Order Form

619-459-6366

Print Form

Ship to

Name: _____
Title: _____
Company: _____
Address: _____
City/State/Zip: _____
Phone: _____
Fax: _____

For Immediate Service

Fax it Back

Fax 619-459-6366
Voice 619-459-6365

☒ Yes!

I want to purchase Blue Sky's Visual Development tools. Please rush my shipment as indicated below.

Quantity	Price	Description	Total Price
	$499	RoboHELP	
	$995	WinMaker Pro	
	$199	Multimedia WinHelp	
	$495	Magic Fields	
	$229	RoboHELP Upgrade (registered users only)	
	$269	WinMaker Pro Upgrade (registered users only)	

Subtotal	
CA residents add applicable sales tax	
Shipping	
Total	

Method of Payment

- ☐ COD (USA only)
- ☐ VISA
- ☐ MasterCard
- ☐ American Express

Method of Payment _____ Exp. Date _____ Signature _____

About

Description Displays the WinHelp application's About topic.

Syntax **About**()

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'*)

Annotate

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

Syntax **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'*)

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**()

BookmarkDefine

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

Syntax **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**()

BrowseButtons

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

Syntax **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'*)

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*')

CheckItem (CI)

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

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

CloseWindow

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

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

Command

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

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

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**()

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**()

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**()

CreateButton

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

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

DeleteItem

Description Removes an item from a WinHelp menu.

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

DeleteMark

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

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

DestroyButton

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

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

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'*)

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'*)

EnableButton (EB)

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

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

EnableItem (EI)

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

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

ExecProgram (EP)

Description Executes an application.

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

Exit

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

Syntax **Exit**()

ExtAbleItem

Description Enables or disables a specified menu item.

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

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*)

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*)

FileOpen

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

Syntax **FileOpen()**

FloatingMenu

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

Syntax **FloatingMenu**()

FocusWindow

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

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

GotoMark

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

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

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**()

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**()

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**()

IfThen

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

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

IfThenElse

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

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

InsertItem

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

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

InsertMenu

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

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

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*')

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')*

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*)

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*)

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**()

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')*

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')*

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**()

Not

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

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

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*)

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*)

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')*

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')*

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**()

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**()

PrinterSetup

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

Syntax **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')*

RemoveAccelerator (RA)

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

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

ResetMenu

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

Syntax **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*')

Search

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

Syntax **Search**()

SetContents

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

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

SetHelpOnFile

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

Syntax **SetHelpOnFile**(filename)

UncheckItem (UI)

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

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