Welcome to What's This? Help Composer

Welcome to *What's This? Help Composer*™, the fastest and easiest way to create context-sensitive Help for all your applications. *What's This? Help Composer* scans an application, then automatically generates an entire What's This? Help project for you! You no longer need to worry about Topic IDs or map files!

"What's This?" Help is context-sensitive Help that is available for each control in a dialog box. What's This? Help topics are usually fairly brief and display in popup windows. This kind of help was introduced with Windows 95 and is used in Windows 95 and Windows NT and later applications.

What's This? Help Composer supports:

- Program files (EXE)
- Dynamic Link Libraries (DLL)
- Visual Basic Projects (VBP)
- ActiveX Controls (OCX)

What's This? Help Composer reads a program file (EXE, DLL, OCX or, in the Visual Basic environment, VBP and FRM) then automatically generates a What's This? Help topic for each control on each dialog box. *What's This? Help Composer* even adds suggested text to each Help topic so much of your Help system is written before you start!

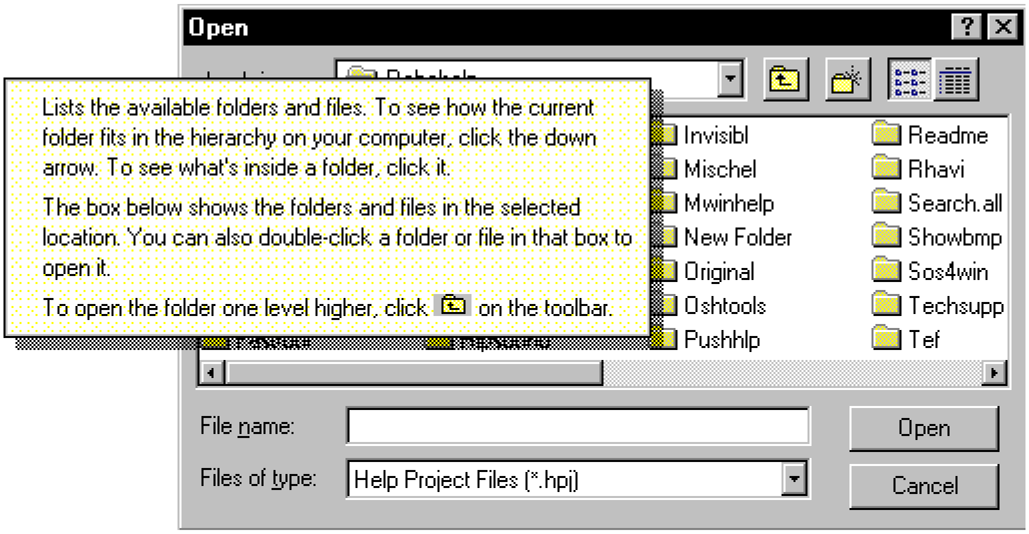
When you are ready to add your own text, *What's This? Help Composer*'s powerful editing environment lets you actually view each dialog box as you're editing the Help topics, so you can see exactly what you're documenting. (Not available for Visual Basic projects.) You can add, edit and format the text, add graphics, and compile your Help system—all from *What's This? Help Composer*.

When you are ready to try out the Help system with the application*, What's This? Help Composer*'s powerful ActiveTest feature allows you to test your What's This? Help with the application, even BEFORE the programmer has added context-sensitive Help links to the program!

What's This? Help Composer is also a time-saver for the programmer. Instead of adding several lines of code for each dialog box to call What's This? Help, the programmer just calls a .DLL that is included with *What's This? Help Composer*. It's that easy.

Note: *What's This? Help Composer* now supports applications created in Visual Basic. Topics specific to Visual Basic are distinguished by "(Visual Basic)" in their headings while those relevant to C/C++ development are identified by "(C/C++)."

What's This Help Example



Lists the available folders and files. To see how the current folder fits in the hierarchy on your computer, click the down arrow. To see what's inside a folder, click it.

The box below shows the folders and files in the selected location. You can also double-click a folder or file in that box to open it.

To open the folder one level higher, click on the toolbar.

File name:

Files of type: Help Project Files (*.hpj)

Key Features of What's This? Help Composer

- Creates an entire Help project for you in seconds!
- Automatically generates a Help topic for each control on each dialog box.
- Eliminates need to keep track of Topic IDs and map files.
- Includes suggested topic text for each control, reducing your editing time.
- Provides a Dialog view that lets you see each dialog box while you’re editing the Help text. This eliminates the need to take screen shots to write topic text.
- Offers a Control view that lets you see a complete list of control properties.
- Includes ActiveTest. This option runs the application with your Help system so you can test your What’s This? Help as you develop it—even BEFORE the programmer has added the Help links to the program.
- Includes ActiveEdit. While testing your Help system, if you spot a change you need to make, just click ActiveEdit to go right to the text in the source file.
- Includes editing and formatting capabilities such as spell check, find and replace, copy and paste, font formatting, and the ability to add graphics to your topics. This means that *What’s This? Help Composer* is the only tool you need for context-sensitive Help creation.
- Includes Toolbars that provide quick access to the most often used functions and formatting tools.
- Contains a "status" item for each Help topic. This lets you track a topic’s status (Generated, In Progress, For Review, or Complete) at any time. This allows you to easily see which topics still need editing and which are complete.
- Project Status Reports lets you see critical information at a glance, including total number of Help topics and numbers of Help topics by progress status.
- Help Author Report displays individual reports on each control in your project with such items as control name, control text and progress status.
- Makes the programmer’s job easier than ever before. To program What’s This? Help, the application developer just needs to call a .DLL—no more adding several lines of code for each dialog box to call What’s This? Help.
- Creates Help for 32-bit applications written for Windows 95, 98, NT 4.0, or 2000 and compiled using Borland, Symantec, Microsoft or Watcom C and C++ compilers.
- Creates Help for 32-bit Microsoft Visual Basic 4.0 or 5.0 applications written for Windows 95, 98, NT 4.0, or 2000.
- Makes it easy to create What’s This? Help!

Benefits of Using *What's This? Help Composer*

What's This? Help Composer is the fastest and easiest way to create context-sensitive Help for all your applications. Consider the benefits to using *What's This? Help Composer* rather than creating What's This? Help the old way.

The Manual Way	**With *What's This? Help Composer***
Print screen shots for each dialog box in the application to view while you write the topic text.	*What's This? Help Composer* displays each dialog box as your write the Help topic text. (Not available in the Visual Basic environment.)
Create a topic for each control.	*What's This? Help Composer* automatically generates the topics.
Create an ID for each control.	*What's This? Help Composer* generates and tracks the topic IDs.
If a program file is updated, find the changes and add/change topics and text.	*What's This? Help Composer* automatically reads the new program file and adds topics, listing dialog boxes that have changed.
Wait until the programmer has added the Help links so you can test your Help topics.	Use ActiveTest to view your Help BEFORE the programmer has added the links. (Not available in the Visual Basic environment.)
Once the programmer has added the Help links, spend hours trying resolve incorrect links in either the application or the Help system.	*What's This? Help Composer* automatically generates the links. They are always correct the first time.
While testing, find a needed change, make a note and go back to the source file to make the change.	Use ActiveEdit to jump right to the Help topic from ActiveTest, and make your changes.
Programmer has to write several lines of code for each dialog box to add the Help links.	Programmer calls a .DLL, spending a fraction of the time previously needed.
Hours of repetitive editing for standard controls (such as OK and Cancel buttons).	*What's This? Help Composer* creates default Help text for standard controls.
Expend much effort understanding the relationship between Topic IDs in your Help system and their corresponding identifiers in the application code.	*What's This? Help Composer* automatically links Help topics with the correct controls, dialogs, labels, etc.

Description of What's This? Help

What's This? Help is context-sensitive help that is available for each control in a dialog box. It is usually fairly short and displays in a popup window.

Example

Invoking What's This? Help

There are three ways for the user to invoke What's This? Help:

- Click ? in the upper right corner of the dialog box, then click a control.

OR

- Place the mouse pointer over a control, then press **F1**.

OR

- Right-click a control, then click **What's This?**

Visual Basic: Two Scenarios

There are two scenarios for developing What's This? Help for applications developed in Visual Basic (VB). The main distinction between the two centers on who runs the What's This Help Composer—the Help author or the developer.

Note: In some cases the Help author needs to have access to and be minimally familiar with Visual Basic.

In the first scenario, in which the Help author runs the *What's This? Help Composer* the procedure can be outlined as follows:

1. The Developer creates VB project.
2. The Developer sends (or otherwise makes available) application project files to the Help author.
3. The Help author uses *What's This? Help Composer* to scan the application project files and create the Help system files.
4. The Help author adds the content of the Help system (the text, graphical content) and compiles it.

Note: Steps 3 and 4 modify the VB project's VBP and FRM files. This is an important difference between developing What's This Help for Visual Basic applications and applications created with C/C++.

5. Help author launches the VB project (either from the What's This Help Composer or the VB interface) and tests the Help system.
6. Help author sends (or otherwise makes available) the Help (HLP) file and the modified VB project files back to the developer.

When a project file is updated (the developer makes changes), the Help author needs to update the files on his or her system (by installing or copying files), and then Update in *What's This? Help Composer*.

Note: One way to handle this is to store program files on a shared drive. When the application developer replaces a file, *What's This? Help Composer* automatically updates the Help project the next time it runs, and notifies the Help author of the changes.

In the second scenario, some of the control of the procedure as outlined above passes from the Help author to the developer.

1. Developer creates VB project.
2. Developer uses *What's This? Help Composer* to scan the project files and create the Help system (i.e., the files that comprise the shell of the Help system, not any of the content).
3. Developer sends (or otherwise makes available) the Help (HLP) file and the Visual Basic project (including the executable file) to the Help author.
4. The Help author uses the *What's This? Help Composer* to add content to the Help system.
5. The Help author compiles the Help system.
6. The Help author runs the application executable file and from within it tests the Help system.
7. The Help author sends (or otherwise makes available) the Help (HLP) file to the developer.

In this scenario, the developer has responsibility for the operation of *What's This? Help Composer* and the VB program files, thus freeing the Help author to concentrate on composing the content of the Help system.

The Help Author's Role (C/C++)

The Help author and the application programmer need to work together to implement What's This? Help, but *What's This? Help Composer* makes this interaction much easier for both.

The Help author needs to do the following to create What's This? Help:

1. Get the most recent program files from the developer.
2. Install the program, or copy the files to his or her system.
3. Run the *What's This? Help Composer* New Program Wizard. The Help structure and all the What's This? Help topics are created automatically.
4. Edit the Help topics.
5. Research and/or add any dialog boxes or controls that require special handling.
6. Compile the Help project.
7. Test the Help project using ActiveTest.
8. If a program file gets updated (the developer makes changes), update the files on your system (by installing or copying files), then do an Update in *What's This? Help Composer*.

   Note: One way to handle this is to store program files on a shared drive. When the application developer replaces a file, *What's This? Help Composer* automatically updates the Help project the next time it runs, and notifies the Help author of the changes.

9. When the application developer is ready to link in the What's This? Help, provide the needed files. See Programming an Application for What's This? Help for a list of the files.

Special Cases

The following types of dialog boxes and controls are "special cases" with respect to What's This? Help.

- dynamic dialog boxes
- custom class dialog boxes
- unused dialog boxes
- duplicate dialog boxes
- dynamic controls
- custom controls
- OLE controls

Dynamic Dialog Boxes

Dynamic dialog boxes are created when the application runs. They are not stored in the resource file, so *What's This? Help Composer* cannot find them when it searches the program files. You can run the application in ActiveTest and add the dialog box to your project using ActiveEdit. In the Dialog Boxes list, dynamic dialog boxes display in uppercase and include "dynamic" in the dialog ID or name.

If you see a dialog box while you are running your application, but you don't see it in your Help project, it is probably a dynamic dialog box.

To add a dynamic dialog box to your project:

1. Compile your Help project.
2. With your project open in *What's This? Help Composer*, click ✓ to run the application in ActiveTest mode.
3. In the application, display the dialog box you want to add.
4. Right-click to display the ActiveTest menu.
5. Click **ActiveEdit**.

   What's This? Help Composer displays, with the dialog box added to your project.

Note 1: **ActiveEdit** is not available in Visual Basic project.

Note 2: Dynamic dialog boxes do not display in dialog view - use control view to see the list of controls.

Custom Class Dialog Boxes

A custom class dialog box is one that the programmer has defined as belonging to a non-standard class. For these dialog boxes, you will need to create What's This? Help by adding the topics to your RoboHELP project. To link in these Help topics, the developer will need to write the What's This? Help calls for these dialog boxes.

Dynamic Controls

Dynamic controls are created when the application runs. They are not stored in the resource file, so *What's This? Help Composer* cannot find them when it searches the program files. You can view the dialog box in ActiveTest and add the control to your project using ActiveEdit.

To add a dynamic control to your project:

1. Compile your Help project.
2. With your project open in *What's This? Help Composer*, click ✓ to run the application in ActiveTest mode. (Not available in Visual Basic projects.)
3. In the application, display the dialog box that contains the control.
4. Right-click on the control to display the ActiveTest menu.
5. Click **ActiveEdit**.

   What's This? Help Composer displays, with the control added to your project.

Note: **ActiveEdit** is not available in Visual Basic project.

Unused Dialog Boxes

What's This? Help Composer reads the resource file for the main program file to get its list of dialog boxes and controls. If the application developer has dialog boxes in the resource file that are no longer used, they will still appear in your Help project. This type of dialog box can exist for several reasons, such as the development tool creates it, but the programmer chooses not to use it. If you think a dialog box may be unused but you aren't sure, the best solution is to ask the developer.

If you find an unused dialog box, you can either leave it in your project but not write Help for it, or you can remove it. By removing these dialog boxes from your active project, you keep the Dialog Boxes list more manageable—you can focus on the dialog boxes that need to have Help text, ignoring those that don't. When you remove a dialog box, it is stored in the Recycle Bin.

Note: This only removes the dialog box from the active Help project, it does not affect the application.

To remove a dialog box:

1. Select the dialog box in the Dialog Boxes list.
2. From the **Edit** menu, select **Remove Dialog**.

Duplicate Dialog Boxes

Duplicate dialog boxes appear in the resource file more than once but look the same. Dialog boxes of this type are only added into What's This? Help once. With a small programming change, you can write Help for the duplicates (maybe the same dialog box is used in two separate places in the program but will need different Help text). The programmer can:

- Register the duplicate dialog using the registration function provided with *What's This? Help Composer*. See CSHUSER.H for details on using this function.

OR

- Change the duplicate in some way such that it is no longer a duplicate. For example, the programmer could change the dialog box caption.

Custom Controls

These types of controls do not display in Dialog View, but they are listed in Control View. You can add What's This? Help for these types of controls.

OLE Controls

These types of controls do not support What's This? Help. If you view the dialog box in Dialog View, an OLE control appears as just a frame (border).

The Programmer's Role (Visual Basic)

Though the Help author and the application programmer need to work together to implement What's This? Help, the *What's This? Help Composer* makes this interaction much easier for both.

During project development, the programmer can expedite the Help development process by doing the following:

- Give frequent program updates to the Help author.

  Note: One way to handle this is to store program files on a shared drive. When you replace a file, *What's This? Help Composer* automatically updates the Help project, notifying the Help author of the changes.

- Notify the Help author if there are any dynamic dialog boxes, custom class dialog boxes, unused dialog boxes, or dynamic controls in the application and, if so, where they can be found.
- Where possible, set a tab order for the controls in the dialog box, linking controls where appropriate.

When it's time to integrate the What's This? Help into the application, use the steps described in Programming an Application for What's This? Help.

The Programmer's Role (C/C++)

Though the Help author and the application programmer need to work together to implement What's This? Help, the *What's This? Help Composer* makes this interaction much easier for both.

As a programmer, you no longer need to add several lines of code for each dialog box just to call What's This? Help. You just make one call to a DLL that contains all the code you need.

During project development, the programmer can expedite the Help development process by doing the following:

- Give frequent program updates to the Help author.

  Note: One way to handle this is to store program files on a shared drive. When you replace a file, *What's This? Help Composer* automatically updates the Help project, notifying the Help author of the changes.

- Notify the Help author if there are any dynamic dialog boxes, custom class dialog boxes, unused dialog boxes, or dynamic controls in the application and, if so, where they can be found.
- Where possible, set a tab order for the controls in the dialog box, linking controls where appropriate.

When it's time to integrate the What's This? Help into the application, use the steps described in Programming an Application for What's This? Help.

Starting a New Project - Standalone

You can create a Help project that contains only the What's This? Help for the application. The result will be a compiled Help (.HLP) file with no Contents tab, just What's This? Help topics. If additional Help files (such as step-by-step Help) are also being created for the application, all of the Help files, no matter how they were created, can be shipped with the application.

To start a new standalone project:

1. Start **What's This? Help Composer**.
2. Select New under the File Menu or click the button on the *What's This? Help Composer* toolbar.

   The first screen of the New Project Wizard displays.
3. Continue through the Wizard to create a new What's This? Help project.

Starting a New Project - From within a RoboHELP Project

If you are working in RoboHELP to create the Help system for an application, you can add What's This? Help to your project. The result, after you compile, will be a single Help (.HLP) file that contains What's This? as well as any other Help you have added.

To add What's This? Help to a RoboHELP project:

1. On the RoboHELP tool palette, click .

   OR

   On the RoboHELP **Tools** menu, click **What’s This Help Composer**.

   The first screen of the New Project Wizard displays.

2. Continue through the Wizard to create the What's This? Help project.

What’s This? Help Composer stores its data in a database, then generates the .RTF file each time the Help file is compiled in RoboHELP. You can edit the .RTF but keep in mind that the changes won' t be reflected in the *What’s This? Help Composer* database. If you go back to *What’s This? Help Composer* and edit Help text, the .RTF file will be regenerated, losing changes made directly to the .RTF file. See Doing Final Edits in RoboHELP.

Using the New Project Wizard

The New Project Wizard generates the What's This? Help topics based on the program files you select. If you run the Wizard from RoboHELP, it adds the What's This? Help to your current project. If you run the Wizard directly from *What's This? Help Composer* it creates a standalone project that only contains What's This? Help.

Each screen of the Wizard prompts you for the information needed to create the What's This? Help. If you need assistance with any screen of the Wizard, use the Wizard's What's This? Help for more information.

What's This? Help Composer generates the Help project. When it is finished, the project results display. When you click **OK** to close the Results dialog, the *What's This? Help Composer* main screen displays, with your project open and ready for editing.

{button ,AL(`New project',0,`',`')}   Related Topics

The What's This? Help Composer Environment

The *What's This? Help Composer* display has several areas, described below. You can adjust the size of the windows by clicking and dragging the separator bars.

Menus

Tool Bars

Status Bar

Dialog Boxes List

Help Text

Dialog View

Control View

Navigating in the *What's This? Help Composer* Screen

Right Click Menus

Menus

File

Edit

View

Format

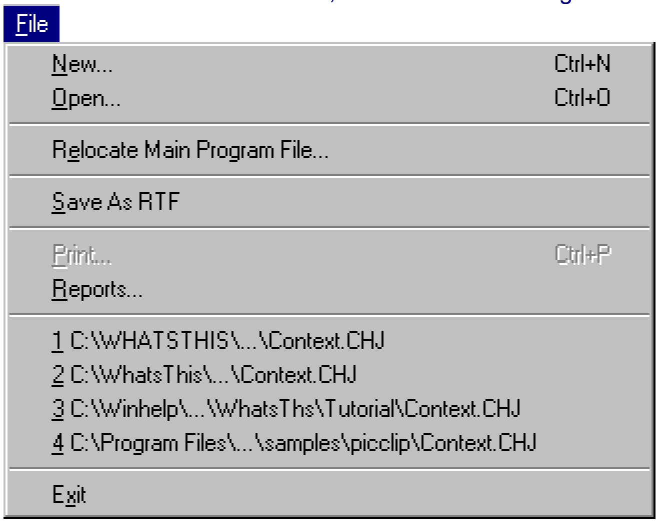
Project

Tools

Help

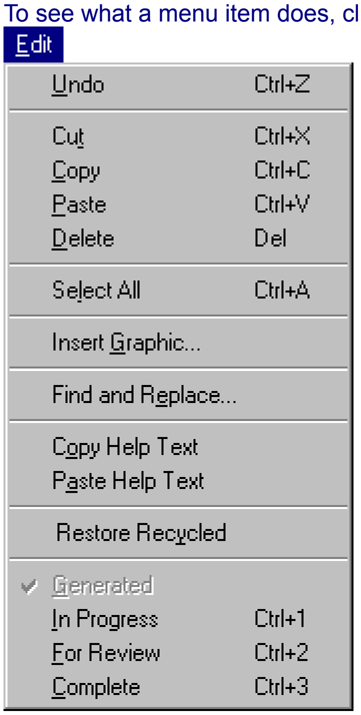
File Menu

To see what a menu item does, click it in the following illustration.


File
New... Ctrl+N
Open... Ctrl+O
Relocate Main Program File...
Save As RTF
Print... Ctrl+P
Reports...
1 C:\WHATSTHIS\...\Context.CHJ
2 C:\WhatsThis\...\Context.CHJ
3 C:\Winhelp\...\WhatsThs\Tutorial\Context.CHJ
4 C:\Program Files\...\samples\picclip\Context.CHJ
Exit

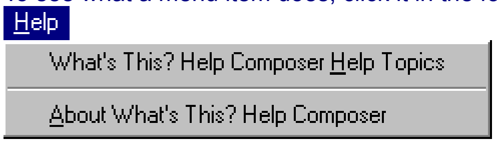
Edit Menu

To see what a menu item does, click it in the following illustration.



View Menu

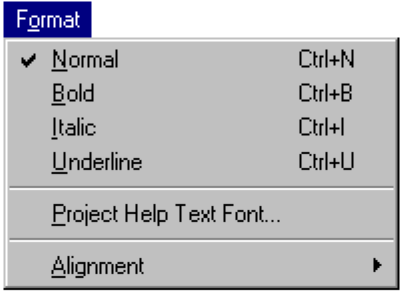
To see what a menu item does, click it in the following illustration.


Help
What's This? Help Composer Help Topics
About What's This? Help Composer

Note: Dialog View and Control View are not available in the Visual Basic environment.

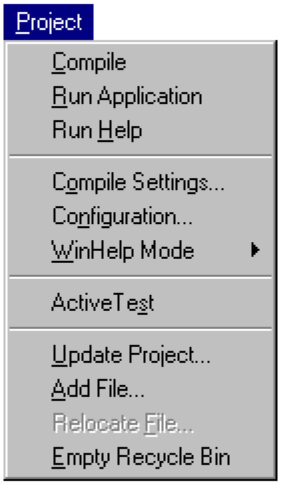
Format Menu

To see what a menu item does, click it in the following illustration.



Project Menu

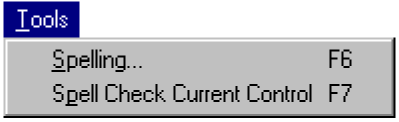
To see what a menu item does, click it in the following illustration.



Note: Active Test is not available in the Visual Basic environment.

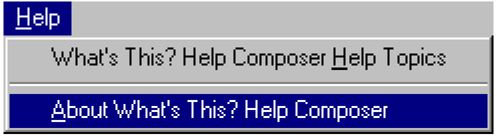
Tools Menu

To see what a menu item does, click it in the following illustration.



Help Menu

To see what a menu item does, click it in the following illustration.

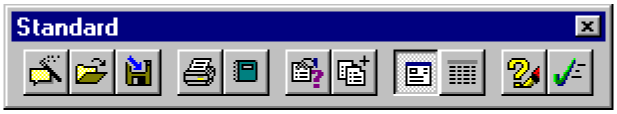

Help
What's This? Help Composer Help Topics
About What's This? Help Composer

Tool Bars

What's This? Help Composer has two toolbars that you can use to access the most commonly used functions. Both toolbars are movable and dockable.

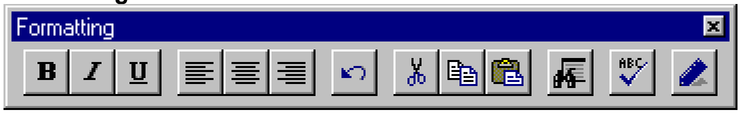
To see what a toolbar button does, click it in the following illustration.

Standard toolbar



Note: Dialog View, ActiveTest and ActiveEdit are not available in Visual Basic projects.

Formatting toolbar



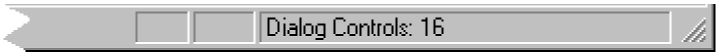
To move a toolbar:

Click on a blank portion of the toolbar (where there isn't a button), and drag the toolbar to its new position. If you drag it to the upper part of the *What's This? Help Composer* window, it will "dock" in the standard toolbar position. If you drag it from this position, it will become a dialog that you can close by clicking the upper right corner.

To view a toolbar:

If a toolbar is not displayed, from the **View** menu click **Toolbars**, then click the toolbar name to display it. These options are toggles - to hide the toolbar click the option again.

Status Bar


Dialog Controls: 16

The *What's This? Help Composer* status bar, located at the bottom of the screen, provides information to help you as you work. The leftmost status field displays Help text. The rightmost field lists the total number of controls in the currently selected dialog box.

To view/hide the status bar:

- From the **View** menu, click **Status Bar**. This option is a toggle - click it again to reverse the show/hide command.

Note: If the total number of Topics in the Project Status Report and the number of dialog controls in the *What's This? Help Composer*'s status line (lower right hand corner) disagree, it is because the status line counts all of the items in a dialog, including "linked controls," which do not support What's This? Help text.

Dialog Boxes List

The left side of the *What's This? Help Composer* screen lists the application modules (files) and dialog boxes. The display is a tree view, with the modules at the highest (leftmost) level. Within each module, the dialog boxes are listed in order by caption name (title) and ID (programmer assigned).

The dialog boxes list also includes the Recycle Bin. For details, see Using the Recycle Bin.

To work with the dialog boxes list:

- Click ⊟ to collapse a section of the tree.
- Click ⊞ to expand a section of the tree.
- Choose **Collapse All** from the **View** menu to collapse the entire tree, displaying only the module names.
- Choose **Expand All** from the **View** menu to expand the entire tree, listing all dialog boxes.
- Click a dialog box name to display either a list of the controls (Control View) or an image of the dialog box (Dialog View).
- Click a module name to list all of the controls in the module (Control View).

Note: In Visual Basic projects, only Control View is available.

Tip
If the status is Completed for all of the controls in a dialog box, the symbol turns to blue in the dialog boxes list.

See Also
Topic Status

Help Text

The Help Text window of the *What's This? Help Composer* display contains the actual Help topic text. This is the text that will display when the user selects What's This? Help. You probably won't do a lot of custom formatting, as this is not standard with What's This? Help, but some formatting options are available. See Formatting Help Text.

The typical edit functions are available in this window, including cut, copy, and paste. You can also copy the entire topic (Edit - Copy Help Text) and then paste it into the Help Text window for another control (Edit - Paste Help Text).

Note: When you switch to another control, or otherwise exit this window, the topic text is automatically saved in the database - you do not need to manually save your Help text. However, *What's This? Help Composer* also provides the ability to "force" a save - choose **Save as RTF** from the **File** menu.

Dialog View

Note: Dialog view is not available in the Visual Basic environment.

In Dialog View, when you select a dialog box an image of the dialog box displays in the lower right section of the *What's This? Help Composer* screen. The currently selected control is outlined in blue. You can select controls either by clicking on them in the dialog box image, or by using the selection keys. When you select a control the Help text for it displays in the Help Text window.

Tips

- If the dialog box is expandable (for example, it may contain a "More" button that shows more selections), to expand it choose **Expand Dialog** from the **View** menu. This option is only enabled for expandable dialog boxes.
- If you see two controls highlighted (outlined in blue) at the same time, they are probably linked.

Control View

In Control View, the lower right section of the *What's This? Help Composer* screen lists the controls for the currently selected module or dialog box. If a module is selected in the Dialog Boxes list, all of the controls for all of the dialog boxes within the module are listed. If a dialog box is selected, only the controls that are contained within the dialog box are listed.

For each control, the following information is shown:

Control Type The control type, such as button, check box, or radio button.

Control Text The label, or text, that displays on the control.

Status The topic status.

Topic ID The Topic Ids. This ID is generated by *What's This? Help Composer* and cannot be changed.

Dialog The name or ID of the dialog box that contains the control.

Sorting the Controls List

By default, the list of controls is sorted by topic ID. You can click a heading label to sort the list by that type. For example, click Status to list the controls in order of status. Note that linked controls always stay together.

Linked Controls

You can choose whether the list includes linked controls. Since you do not need to write Help text for linked controls, you can hide them in the list, making it easier to work with the controls that do need Help text.

To toggle the display of linked controls:

- From the **View** menu, choose **Linked Controls**.

Note: View Linked Controls option is not available in Visual Basic Projects.

Types of Controls

When you view a dialog box in Control View, the type is listed for each control. The default Help text is based on this type, but the type can also help you, the Help author, decide what to write for the control. The following list includes most of the control types you are likely to encounter, but you may see others as well.

Aa Static Text

ab| Edit Control

List Box

Combo Box

Check Box

Group Box

Radio Button

Button

Owner Draw Button

List View Control

Slider Control

Frame

Image

Rectangle

Tab Control

Progress Bar Control

Horizontal Scroll Bar

Vertical Scroll Bar

Tree View Control

Navigating in the What's This? Help Composer Screen

As you work in *What’s This? Help Composer* you can use several methods of navigation. These include hot keys as well as menu items. The navigation options are located on the View menu.

Menu Option	Hot Keys
Next Control	Ctrl+Dn
Previous Control	Ctrl+Up
Next Dialog	Ctrl+PgDn
Previous Dialog	Ctrl+PgUp

See Also

Dialog Boxes List

Right Click Menus

What's This? Help Composer has right-click menus that allow quick access to the most commonly used functions, based on where the mouse pointer is located.

To display a right-click menu, click the right mouse button.

Adding/Updating Text

To add or change Help topic text, select the control to display the text in the Help Text window, then make your changes.

{button ,AL(`edit',0,`',`')}   Related Topics

Selecting a Control

- Use the navigation keys or **View** menu options to sequence through the dialog boxes and controls until the control you want is selected.

OR

- Select the dialog box in the Dialog Boxes list, then click the control in either Dialog View or Control View.

Note: You will probably want to toggle between Dialog and Control Views to access all of the controls and see what they are so you can write the Help text. If a control does not display in either of these views, but it displays in the application, it is probably a dynamic control. You can add dynamic controls to your project using ActiveTest.

See Also

Dynamic Controls

Deleting Help Text

If you delete all of the text for the topic, this is the equivalent of removing the topic - there will not be What’s This? Help for the control and the user will receive the message "No Help topic associated with this item."

To delete Help text:

1. In the Help Text window, select the text you want to delete.
2. Press **Del**.

   OR

   From the **Edit** menu, choose **Delete**.

Adding Graphics

1. Position the insertion point where you want to add the graphic.
2. Click ✏.
3. Select the folder where the graphic is located.
4. Double-click the graphic name.

   The code for the graphic displays.

Jumps and Popups

WinHelp 4.0 **does not** support the use of popups in What's This? Help. If you want to popup a topic from a What's This? Help topic, use the PopupID macro (see the RoboHELP or WinHelp Help system for information on using this macro).

WinHelp 4.0 **does** support the use of jumps in What's This? Help, but always displays the "jump to" topic in the Main Help window. If you want to jump to a topic and display it in a secondary window, use the JumpID macro instead of a regular jump (see the RoboHELP or WinHelp Help system for information on using this macro).

Spell Checking

1. Click ABC✓.

   OR

   From the **Tools** menu, choose **Spelling**.

   The Project Spell Check dialog box displays. The options on this dialog box change depending on what you have selected when you choose Spelling.

2. Select an option.

3. Click **OK** to begin the spell check. (To stop spell checking press the **ESC** key.)

Tip: If you just want to spell check the Help text for the current control, set the focus to the Help Text field and select **Spell Check Current Control** from the **Tools** menu (or press **F7**).

Finding and Replacing Text

What’s This? Help Composer’s Find and Replace feature lets you quickly and easily find and replace works and phrases in your What’s This? Help project. Find and Replace searches through each control in every dialog box in your project for the specified text.

1. Click 🔍.

   OR

   From the **Edit** menu, choose **Find and Replace**.

   The Find and Replace dialog box displays.
2. Type in the text to be found.
3. Type in the text to be entered in place of the found text.
4. If you want to find occurrences that are whole words, not part of a larger word, select **Match Whole Word Only**.
5. If you want the search to distinguish between upper and lower case characters, select **Match Case**.
6. To replace with confirmation, click **Find Next** to find the first occurrence of the text. Then click **Replace** to replace the text, or **Find Next** to continue without replacing.

   OR

   To replace all occurrences, without confirmation, click **Replace All**.
7. When you have finished replacing text, click **Cancel** to close the Find and Replace dialog box.

{button ,AL(`edit',0,`',`')} Related Topics

Copying and Pasting Text

1. In the Help Text window, select the text you want to copy (or cut).
2. Click ⎘ to copy, or ✂ to cut.
3. Place the insertion point where you want to place the text.
4. Click 📋.

Tip

If you want to copy an entire topic, you can use the **Copy Help Text** command on the **Edit** menu. Then you can use the **Paste Help Text** command to place the entire topic into a new topic, replacing the existing text.

{button ,AL(`edit',0,`',`')} Related Topics

Formatting What's This? Help Text

Setting the Font Type

Setting the Font Size

Setting the Font Style

Setting Paragraph Alignment

Bulleted Lists

Numbered Lists

{button ,AL(`edit',0,`',`')}   Related Topics

Setting the Font Type

1. From the **Format** menu, select **Project Help Text Font**.

   The Help Text Font dialog box displays.

2. Select the font from the list.
3. Click **OK** to close the dialog box.

Note: For consistency in What's This? Help, all text must be the same type. Therefore, the font type affects all the text in the What’s This? Help project.

{button ,AL(`font',0,`',`')} Related Topics

Setting the Font Size

1. From the **Format** menu, select **Project Help Text Font**.

   The Help Text Font dialog box displays.

2. Select the font size from the list.
3. Click **OK** to close the dialog box.

Note: For consistency in What's This? Help, all text must be the same size. Therefore, the font size affects all the text in the What’s This? Help project.

{button ,AL(`font',0,`',`')} Related Topics

Setting the Font Style

1. Select the text to be formatted in the new style.
2. Select the style, Bold, Italics, or Underline:

   - Click a button on the toolbar.

   OR

   - Select an option from the **Format** menu.

Tip

You can reformat text to normal by selecting it, then selecting **Normal** from the **Format** menu.

{button ,AL(`font',0,`',`')} Related Topics

Setting Paragraph Alignment

1. Place the insertion point anywhere in the paragraph to be aligned.
2. Select the alignment option, Left, Center, or Right:
   - Click a button on the toolbar.

   OR

   - From the **Format** menu, select **Alignment**, then select the option.

Bulleted Lists

To create a bulleted list in What's This? Help, use a bitmap as a bullet.

1. Place the insertion point where you want the bullet to appear.
2. Click ▭.
3. Select the folder where the Bullet graphic is located.
4. Double-click the graphic name.

   The code for the graphic displays.

Tip

A default bullet bitmap is located in the \WinHelp\RoboHELP\Special folder.

Numbered Lists

Numbered lists are not often seen in What's This? Help. However, if you want to create short numbered lists, simply type in the numbers manually.

Topic Status

The topic status displays for each control in Control View. You can sort the list of controls by Status to get an overall view of how close the project is to completion. If the status is Completed for all of the controls in a dialog box, the symbol turns to blue ▭ in the Dialog Boxes list.
When a Help topic is generated (by the New Project Wizard or by an Update), the status is set to Generated (linked controls are the exception - see Note, below). After you edit a topic for the first time, its status automatically changes to In Progress. The other status codes are for use by the Help author(s).

You can change a topic's status using any of the following methods:

- Right-click in the Help Text window and select the new status.
- Right-click the Dialog in the Dialog Tree and select Set Dialog Status:
- Right-click the control (Dialog View or Control View) and select the new status.
- Click the control (Dialog View or Control View) and select the new status from the **Edit** menu.
- Click the control and press the hot keys for the new status:

In Progress	Ctrl+1
For Review	Ctrl+2
Complete	Ctrl+3

Note: Linked controls always have a status of Linked.

Opening an Existing Project

If your What's This? Help project is part of a RoboHELP project, you can open it directly from RoboHELP. If it's a standalone project, open it from within *What's This? Help Composer*.

From RoboHELP:

- On the RoboHELP tool palette, click ▭.

OR

- On the RoboHELP Tools menu, click **What's This Help Composer**.

The *What's This? Help Composer* window displays, with your project open and ready for editing.

From *What's This? Help Composer*:

1. Start What's This? Help Composer.
2. Click 📂.
3. Select the folder where the project is stored.
4. Double-click the project file name.

Configuring the Project

The Configuration settings describe how your project is set up, including such information as where your files are located. All of these settings are determined when you create a new project - the New Project Wizard records them.

To change your configuration settings:

1. From the **Project** menu, choose **Configuration**.

   The Configuration dialog box displays.

2. Change the settings as needed. If you need assistance, press F1 in any field for Help.
3. Click **OK** to save the settings and close the Configuration dialog box.

The Configuration dialog box contains the following configuration settings.

- Main Program File
- Run Command
- Help File to Build
- Starting Map Number
- Highest Map Number
- Help Topic ID Prefix
- Use Microsoft Visual C++ Help IDs
- Auto-Open Last Project on Startup
- Create Default Help Text for Added Dialogs
- Confirm Removal to Recycle Bin

Summary Report

Anytime you create or update a What's This? Help project, *What’s This? Help Composer* generates a Summary Report. **You should always keep a copy of this report!** This report lists the most current *What's This? Help Composer* activity. For certain types of information, such as dialogs added, only the most current data is included. For other types of information, such as duplicate dialogs, the data is cumulative because it doesn't change.

Project Information

This section of the report shows totals for the project.

Project - The path and name of the project (.CHJ) file.

Date/Time - The date and time the report was generated.

Successfully scanned files - The number of files scanned.

Files not scanned - The number of files that were selected but not scanned. These may be 16-bit files, or they may be files that do not contain any dialog boxes.

Total changed dialogs - If you ran an update, the number of dialog boxes that changed.

Total added dialogs - The number of dialog boxes added to the project on this pass.

Total removed dialogs - If you ran an update, the number of dialog boxes that were removed. Removed dialog boxes (and their Help text) are stored in the Recycle Bin.

Total duplicate dialogs - The number of dialog boxes that were NOT added to the Help project because they appear in the resource file more than once. This is the total number for the project, not just for the current update. The programmer needs a list of these so they can be registered for What's This? Help.

Total custom class dialogs - The number of dialog boxes that were NOT added to the Help project because they belong to a custom class, created by the application developer. This is the total number for the project, not just for the current update. You will need to create What's This? Help for these dialog boxes manually.

File Information

This section of the report shows details for each file in the program.

File - The name of the file (program module).

Path - The full path and name of the file.

Version - The version number of the file, if applicable.

Modified - The date and time of the file.

Comments - If the file could not be scanned, or it does not contain any dialog boxes, a comment appears, describing the issue.

Changed dialogs - If you ran an update, a list of the dialog boxes that changed.

Added dialogs - A list of the dialog boxes that were added to the project on this pass.

Removed dialogs - If you ran an update, a list of the dialog boxes that were removed.

Duplicate dialogs - A list of the dialog boxes that were NOT added to the Help project because they appear in the resource file more than once. The programmer needs this list so the dialog boxes can be registered for What's This? Help.

Custom class dialogs - A list of the dialog boxes that were NOT added to the Help project because they belong to a custom class, created by the application developer. You will need to create What's This? Help for these dialog boxes manually.

Adding a Standalone Project to a RoboHELP Project

If you created separate What's This? and RoboHELP projects, you can combine them, resulting in a single .HLP file after compile. However, you don't need to do this - you can ship multiple .HLP files with the application.

1. Open the .CHJ file of your *What's This? Help Composer* project (use any text editor, such as Notepad or Word).
2. Change the path on the line "Help file to build" to the path/name of your RoboHELP .HLP file.
3. Change the path on the line "Help project file" to the path/name of your RoboHELP .HPJ file.
4. Save and close the file.
5. Open the .RBH file associated with the primary document in your RoboHELP project (use any text editor, such as Notepad or Word).
6. Add the following line to the first section of the RBH:

   ```
   CHFile=path
   ```

   where "path" is the path/name of your What's This? .CHJ file.
7. Save and close the file.
8. Open the RoboHELP project, go to **Project Setup**, and select the **Advanced** tab.
9. Double-click the **[FILES]** section. Select the .RTF file from your What's This? project and click **Add**. Then click **OK** to close.
10. Double-click the **[BAGGAGE]** section. In the List Files of Type box, select **All Files**. Select **DIALOG.CID** from your What's This? project and click **Add**. Then click **OK** to close.
11. Click **OK** to exit Project Setup.

The projects are now linked. When you run *What's This? Help Composer* from RoboHELP, it will launch the correct What's This? Help project. When you compile, both projects will be combined as one .HLP file.

Removing the What's This? Help from a RoboHELP Project

1. Open the RoboHELP project, go to **Project Setup**, and select the **Advanced** tab.
2. Double-click the **[FILES]** section. Select the .RTF file from your What's This? project and click **Remove**. Then click **OK** to close.
3. Click **Yes** to remove the associated .HH file.
4. Click **OK** to exit Project Setup.

Note: If you go back to *What's This? Help Composer* and compile the project, the file will be added back to the RoboHELP project.

Tips

- If you have a large What's This? project, you can remove it to speed up compile time while developing the "how to" portion of the Help system. Remember to add it back in before compiling the final Help system.
- This process removes the What's This? .RTF from the RoboHELP compile - it doesn't delete the .RTF file from the project or from your system.

Moving a What's This? Project

Standalone

If you are working with a standalone What's This? Help project you can move it without having to make any changes to the files. Since all the files for a What's This? Help project are stored in the same directory, all you need to do is move the files to a new folder and you're done.

Part of a RoboHELP Project

If you need to move a What's This? Help project that is part of a RoboHELP project, do the following:

1. Move the What's This? project files to the new folder.
2. Open the .RBH file associated with the primary document in your RoboHELP project (use any text editor, such as Notepad or Word).
3. Change the path on the line "CHFile=" to the new path of your What's This? .CHJ file.
4. Save and close the file.
5. Open the RoboHELP project, go to **Project Setup**, and select the **Advanced** tab.
6. Double-click the **[FILES]** section. Remove the old location of the What's This? .RTF file and then add the new one. Then click **OK** to close.
7. Double-click the **[BAGGAGE]** section. In the List Files of Type box, select **All Files**. **Remove** the old location of the DIALOG.CID file and then **Add** the new one. Then click **OK** to close.
8. Click **OK** to exit Project Setup.

{button ,AL(`moving',0,`',`')} Related Topics

Moving a Program File

1. Move the file.
2. In the *What's This? Help Composer* Dialog Boxes list, select the progam file.
3. From the *What's This? Help Composer* **Project** menu, choose **Relocate File**.

   The Locate File dialog box displays.
4. Enter the new location, or use **Browse** to select it.
5. Click **OK** to close the Locate File dialog box.

{button ,AL(`moving',0,`',`')} Related Topics

Renaming a Standalone What's This? Project

1. Rename the What's This? files (use Windows Explorer or My Computer).
2. Open the .CHJ file of your *What's This? Help Composer* project (use any text editor, such as Notepad or Word).
3. Change the file prefix on the following lines to the new name of your project.

File Type	Extension
Help file to build	.HLP
Database file	.DBF
Help project file	.HPJ
Default topic file	.DTO
Help map file	.HH file

4. Save and close the file.
5. Open the .HPJ file of your *What’s This? Help Composer* project (use any text editor, such as Notepad or Word).
6. In the [OPTIONS] section, change the name on the line "HLP=" to the new name of your project Help file.
7. In the [FILES] section, change the prefix for the .RTF file to the new name of your project.
8. In the [MAP] section, change the prefix for the .HH file to the new name of your project.
9. Save and close the file.

Note: If the renamed project was the last one opened in *What’s This? Help Composer*, you will get a "file not found" error when you start *What’s This? Help Composer*. Click **OK** to clear the message, then open the project using **File** - **Open**.

{button ,AL(`moving',0,`',`')} Related Topics

Renaming a What's This? Project that is Part of a RoboHELP Project

1. Rename the What's This? files (use Windows Explorer or My Computer).
2. Open the .CHJ file of your *What's This? Help Composer* project (use any text editor, such as Notepad or Word).
3. Change the file prefix on the following lines to the new name of your project.

File Type	Extension
Database file	.DBF
Default topic file	.DTO
Help map file	.HH file

4. Save and close the file.
5. Open the .RBH file associated with the primary document in your RoboHELP project (use any text editor, such as Notepad or Word).
6. Change the name on the line "CHFile=" to the new name of your What's This? .CHJ file.
7. Save and close the file.
8. Open the RoboHELP project, go to **Project Setup**, and select the **Advanced** tab.
9. Double-click the **[FILES]** section. **Remove** the old name of the What's This? .RTF file and then **Add** the new one. Then click **OK** to close. Click **Yes** to remove the old Map file.
10. Double-click the **[MAP]** section. **Add** the new .HH file and click **OK** to close.
11. Click **OK** to exit Project Setup.

{button ,AL(`moving',0,`',`')} Related Topics

Doing Final Edits in RoboHELP

Normally, you will do all of your editing within *What's This? Help Composer*. However, if you want to add a custom feature (such as additional formatting) that is not available in *What's This? Help Composer*, you can bring the .RTF file into RoboHELP and make final changes.

Note: *What's This? Help Composer* stores its data in a database, then generates the .RTF file when it compiles the Help file. When you edit the .RTF file in RoboHELP, the changes are not copied to the *What's This? Help Composer* database. If you go back to *What's This? Help Composer* and compile, the .RTF file will be regenerated, losing changes made in RoboHELP. Remember, you need to recompile in *What's This? Help Composer* if ANY change is made to the application.

For details, see:

Viewing What's This? Help Source

Editing the What's This? .RTF File in RoboHELP

Viewing What's This? Help Source

During project development, if you want to view the What’s This? .RTF file, you can open it as a read-only file in RoboHELP. You won’t be able to edit the file, but you can see it just as you would view any other .RTF in RoboHELP.

To view What’s This? Help Source:

- From the RoboHELP **Project** menu, choose **View What’s This? Help Source**.

The .RTF file displays in RoboHELP.

Editing the What's This? .RTF File in RoboHELP

Note: This should be the VERY LAST THING you do on your project. Be sure you have gotten the latest source from the programmer, and done one last compile in *What’s This? Help Composer* before opening your What’s This? document in RoboHELP.

To make final changes in RoboHELP:

1. From the RoboHELP Project menu, choose Unlock What’s This? Help Source.

   This attaches the RoboHELP template to the What’s This? .RTF file. The topics are sorted by dialog box title (alphabetical) and within dialog box, by tab order.

2. Make the changes as needed.
3. Compile the project from within RoboHELP.

Using Programmer Generated Topic IDs

What's This? Help Composer generates all your Help topics, assigning topic IDs in the process. So what do you do if the programmer has already generated the IDs? You have two options:

1. Use the IDs generated by *What's This? Help Composer*. If IDs have already been coded into the application, the programmer will need to remove them and use the *What's This? Help Composer* .DLL instead. See Programming an Application for What's This? Help. This is the preferred method as it allows you, and the programmer, to make the most use of the automation that *What's This? Help Composer* provides.

OR

2. If (1) is not an option, you can use the IDs that the programmer has generated. You can either edit the RoboHELP map file to match the programmer's IDs (preferred method), or you can include a programmer generated map file in your project.

Caution: These steps require a good working knowledge of RoboHELP - if you are new to the product, be very careful when changing topic IDs, and refer to your RoboHELP User's Guide for specifics where RoboHELP is used.

To use programmer generated topic IDs:

1. Create What's This? Help as a standalone project in *What's This? Help Composer*.

   Note: If you are also creating step-by-step Help using RoboHELP, create the RoboHELP project and What's This? Projects separately.

2. Compile your What's This? Help system.
3. Test your What's This? Help system using ActiveTest.
4. If you have not yet created a RoboHELP project, create one now.

   Note: If you only need What's This? Help, the RoboHELP project will be used as a "shell" to facilitate changing the What's This? topic IDs.

5. Open your RoboHELP project.
6. Add the What's This? .RTF file to the project.
7. Edit the topic IDs:

   - Use a RoboHELP generated map file and edit the IDs to match those the programmer has provided.

   OR

   - Use a programmer-generated map file, including it in your project.

Editing the RoboHELP Generated Map File

Complete the steps listed in Using Programmer Generated Topic IDs before editing the map file.

1. In RoboHELP, open the What's This? .RTF file.
2. Attach the **robortf.dot** template to the file.
3. Save the .RTF file as a .DOC file (use **Save As**).
4. Close the file.
5. Open the .DOC file you just created.
6. Turn on map file generation for the What's This? document in RoboHELP (see the RoboHELP User's Guide for details if you don't know how to do this).
7. Compile the Help project in RoboHELP.
8. Edit the generated .HH file (use any text editor, such as Notepad), changing the topic IDs to those provided by the programmer.

Using a Programmer Generated Map File

Note: This is not the preferred method of changing topic IDs, so you should only use it if you have no other options, i.e., you have to use the .HH provided by the programmer. Another option is to edit the RoboHELP generated map file to use the programmer's IDs.

Complete the steps listed in Using Programmer Generated Topic IDs before including the map file.

1. Place the map file in the same folder as your RoboHELP project.
2. Add the map file to your RoboHELP project.
3. In RoboHELP, open the What's This? .RTF file.
4. Attach the **robortf.dot** template to the file.
5. Save the .RTF file as a .DOC file (use **Save As**).
6. Close the file.
7. Open the .DOC file you just created.
8. For **each topic** where you need to change the ID:
    - Place the mouse pointer anywhere in the Help topic.
    - Click.
    - Click **Choose**.

    Note: If the **Choose** button is grayed out (unavailable), go to **Document Properties** (**File** menu), choose **More**, and turn off map file creation for this document.

    - Select the correct topic ID in the list.
    - Click **OK** to assign the ID to the topic.
    - Click **OK** to close the Edit Help Topic dialog.

Compiling a What's This? Project

If your What's This? project is part of a RoboHELP project, you can compile from within RoboHELP or from within *What's This? Help Composer*.

If you are working with a standalone What's This? project, compile from within *What's This? Help Composer*.

Note: At the end of the project, when you are sure that all changes to the application have been completed, do a compile from within *What's This? Help Composer*. This updates the Dialog.CID file to match the dialogs in the application. If this is not done, What's This? Help will not work for any controls that have changed since the last *What's This? Help Composer* compile.

Compiling from RoboHELP:

1. Click ?.
2. Click Make.

Compiling from *What's This? Help Composer:*

- Click ?.

What's This? Project Files (Visual Basic)

What’s This? Help Composer creates several files in your project folder, indicated by the following extensions:

.CDX Database indexing file.

.CHJ *What’s This? Help Composer* project file. This is similar to the .HPJ file for a regular Help project.

.DBF Main database file.

.DTO Default topic.

.FPT Database indexing file.

.HH What's this? map file. This is NOT the equivalent of the .HH for a regular (non-What's This?) context sensitive map file, and should not be edited.

.HLP Help file. This file is only present after compile, if the What's This? project is not part of a RoboHELP project.

.HPJ What's this? project file, similar to the HPJ for a regular Help project.

.RTF Help document. This is created when you compile the Help project.

Visual Basic Files Modified by the What’s This Help Composer

When you use the *What’s This? Help Composer* in the Visual Basic environment, it not only creates the files above but modifies some files in the Visual Basic project itself. These files are:

.VBP The Visual Basic project file. This file contains lists of the files and resources contained in the project.

.FRM Each Visual Basic form (in essence, a dialog box), is described by data contained in its corresponding FRM file. The What’s This Help Composer adds information to FRM files to implement What’s This Help.

What's This? Project Files (C/C++)

What's This? Help Composer creates several files in your project folder, indicated by the following extensions:

.CDX Database indexing file.

.CHJ *What's This? Help Composer* project file. This is similar to the .HPJ file for a regular Help project.

.CID Contains the links between the dialog box controls and the Help topic IDs. This file is stored as baggage in the compiled .HLP file.

.DBF Main database file.

.DTO Default topic.

.FPT Database indexing file.

.HH What's this? map file. This is NOT the equivalent of the .HH for a regular (non-What's This?) context sensitive map file, and should not be edited.

.HLP Help file. This file is only present after compile, if the What's This? project is not part of a RoboHELP project.

.HPJ What's this? project file, similar to the .HPJ for a regular Help project.

.RTF Help document. This is created when you compile the Help project.

What's This? Topic IDs

What's This? Help Composer automatically generates the IDs for all What's This? Help topics. You cannot change these IDs. The ID for each control displays in Control View.

Note: *What's This? Help Composer* generates all the code needed to map control IDs to What's This? Help IDs, so the programmer doesn't have to manually enter this information. See Programming an Application for What's This? Help for more information.

Changing the Compile Settings

The Compile settings describe how your What's This? project is to be compiled. They are a subset of the RoboHELP compile settings - if you are working with a project that is part of a RoboHELP project, you can change settings in either place and they will be updated for the entire project.

To change your What's This? Compile settings:

1. From the **Project** menu, choose **Compile Settings**.

   The Compile Settings dialog box displays.
2. Change the settings as needed. If you need assistance, press F1 in any field for Help.
3. Click **OK** to save the settings and close the Compile Settings dialog box.

Selecting the WinHelp Mode

When you run your Help system in Help Author mode, the topic number displays in the title bar, and you can see the jump text.

To turn on Help Author mode:

1. From the **Project** menu, select **WinHelp Mode**.
2. Click **Help Author**.

   A check mark indicates that the option is selected.

Running the Application (Visual Basic)

To help you conveniently test your Help system, you can run the application anytime, directly from *What's This? Help Composer*. To run the application,

- Click the What's This? Help Composer's **Project** menu,
- Select **Run Application**.

Notes: Before you ship the application, you should run it independent of *What's This? Help Composer* and test your Help.

If the developer is performing the *What's This? Help Composer* creation and updating tasks, and thus the Help author is working only with the application executable file and not the project files, then, in order to run the application from within *What's This? Help Composer*, the Help author must,

1. From the *What's This? Help Composer*'s **Project** menu, select **Configuration…**
2. In the field marked **Run Command:** change the extension following the project filename from **VBP** to **EXE**.

Running the Application (C/C++)

You can run the application anytime, directly from *What's This? Help Composer*.

- If you do this before the programmer has added What's This? Help, you won't be able to see your Help topics - you can use ActiveTest if you want to test your Help topics.
- After the programmer has added What's This? Help capability into the application, you will be able to view your Help topics while running the application. This method uses the programmed Help links, whereas ActiveTest simulates the links.

Note: Before you ship the application, you should run it independent of *What's This? Help Composer* and test your Help.

To run the application from *What's This? Help Composer*:

From the **Project** menu, choose **Run Application**.

Running the Help System

You can run your Help system directly from *What's This? Help Composer*. This option does not allow you to test your What's This? Help (you need the application to access it), but you can display the rest of your Help system.

Note: To test your What's This? Help, use ActiveTest.

To run the Help system from *What's This? Help Composer*:

- From the **Project** menu, choose **Run Help**.

Using ActiveTest to Edit Help Topics

ActiveTest is a powerful feature that allows you to test your Help anytime, even if the developer hasn't yet enabled What's This? Help in the application. If you need to make changes, you can jump right to your Help source using ActiveEdit.

1. Compile the Help project.
2. In *What's This? Help Composer*, click ▭.

   The application displays.

3. Access a dialog box for which you have written What's This? Help.
4. Right-click on the dialog box.

   A menu displays, with the options ActiveEdit and What's This?

5. Click **ActiveEdit** to bring up *What's This? Help Composer* with the Help topic.

   OR

   Click **What's This** to display the What's This? Help.

Note: ActiveTest and ActiveEdit are not available in Visual Basic projects. Clicking the ▭ will load the current .VBP file in Visual Basic.

Adding Program Files to a Project

You can add program files to your project any time. If a file has been added to the .EXE since you created the project, it will automatically be added when you do an Update. What’s This? Help Composer can also update your projects to include .DLL and .OCX files.

1. Click .

   The Add Program File dialog box displays.

2. Select the folder where the file is located.
3. Double-click the file to add.

Note: The **Adding Program Files** option is not available in Visual Basic projects.

Removing a Program File from a What's This? Project

1. Highlight the file name in the Dialog Boxes list.
2. From the **Edit** menu, choose **Remove File**.

   The file is stored in the Recycle Bin. See Using the Recycle Bin.

Updating a Project

Each time you open a What's This? Help project, *What's This? Help Composer* checks the main program file to be sure it hasn't changed. If it has changed, the project is updated. You can also do a manual update.

When *What's This? Help Composer* updates your project, it compares the program files and makes sure that any changes are reflected in the Help system. For example, if a dialog box has been added to the program, it is added to your Help project.

To do a manual update:

1. Click .

   The Update Project dialog box lists the files in the project, along with their status.

2. Select (highlight) the files to be updated.
3. Click **Update**.

Default Help Text

Default Help text is generic text that can be automatically generated for the basic controls (such as check boxes and list boxes). You can generate the default Help text and then go back and edit it, making small changes where needed.

When you create a new project using the New Project Wizard, you have the option of including default Help text. After the project is generated, you can turn default Help text on and off for subsequent updates. If it's turned on, default Help text will be created for controls that are added during the update. See Configuring the Project for details.

Note: The default Help text are stored in the Registry and can be configured. To define the default Help text change the string values for the following registry key "\Hkey_Current_User\Software\Blue Sky Software\What's This? Help Composer\1.00\Default Text".

Using the Recycle Bin

If a program contains dialogs for which you do not need to write help, you can remove them from your What’s This? Help project. When you remove a dialog box or program file from your project it is automatically stored in the Recycle Bin. Items in the Recycle Bin are removed from the active project but they are still on file—you can add them back if you need to, and a project update will find them.

The Recycle Bin appears at the end of the Dialog Boxes list on the left side of the *What’s This? Help Composer* display. It stores files, dialog boxes and controls that you have removed from the active project as well as files, dialog boxes and controls that were removed by a project update.

Tips

- If you empty the Recycle Bin (not recommended), a project update is likely to add the files and dialog boxes back into your project—it doesn’t find them so thinks they are new.
- Dialog View is not available for dialog boxes in the Recycle Bin—*What’s This? Help Composer* automatically switches to Control View when you select one of these files or dialog boxes (if you were previously working in Dialog View, it switches back when you select an active dialog box).

{button ,AL(`recycle',0,`',`')} Related Topics

Restoring Recycled Items

You can move dialog boxes and files from the Recycle Bin back into the active project one at a time, or you can move the entire contents of the Recycle Bin back into the active project.

To move a dialog box or file back to the active project:

1. Click the item in the Recycle Bin to select it.
2. Right-click the Recycle Bin and select **Restore Recycled**.

OR

1. Click the item in the Recycle Bin to select it.
2. Select **Restore Recycled** from the Edit menu.

{button ,AL(`recycle',0,`',`')} Related Topics

Emptying the Recycle Bin

Caution: When you empty the Recycle Bin, all of its contents are permanently removed - there is no Undo to reverse this action. If you Update your project, any items that were stored in the Recycle Bin but still exist in the application resource files, will be re-added to your project.

To empty the Recycle Bin:

- Select the Recycle Bin in the Dialog Boxes list and select **Empty Recycle Bin** from the **Project** menu.

OR

- Right-click the Recycle Bin and select **Empty Recycle Bin**.

You can also permanently delete individual dialogs from the Recycle Bin. To do so:

- In the Recycle Bin, right click the dialog you wish to delete.
- From the right click popup menu, select **Delete**.

{button ,AL(`recycle',0,`',`')} Related Topics

Printing Topics

When you print a Help topic from *What's This? Help Composer* you get a complete description of the current control as well as the Help text. The description prints in report form, showing the project location, control type, control title, topic status, topic ID, and dialog box name.

To print a topic from *What's This? Help Composer:*

1. Display the topic in the *What's This? Help Composer* window.
2. Click 🖨.

   OR

   From the **File** menu, choose **Print**.

Before You Ship

At the end of the project, there are just two things you need to do to be sure the files are ready.

- After the programmer has added the code for What's This? Help, test the Help by running the application independent of *What's This? Help Composer* (don't start the application from within *What's This? Help Composer*). This tests the links.
- When you are sure that all changes to the application have been completed, do a compile from within *What's This? Help Composer.* This updates the Dialog.CID file to match the dialogs in the application. If this is not done, What's This? Help will not work for any controls that have changed since the last *What's This? Help Composer* compile. See Compiling a What's This? Project, earlier in this chapter, for details.

Note: The Dialog.CID file stores information about the controls for each dialog box. The information in this file must match the actual dialog box for What's This? Help to work properly.

Files to Ship (Visual Basic)

When you ship an application that includes What's This? Help created with *What’s This? Help Composer*, you need to include the following files:

<project>.HLP The Help file for the application. What's This? Help is part of this file. This file is usually installed into the user's application directory.

<project>.CNT The contents file for the Help system. This file must be installed into the same directory as the .HLP file.

Note: If only What’s This? Help is being provided, the CNT file is not needed.

Files to Ship (C/C++)

When you ship an application that includes What's This? Help created with *What’s This? Help Composer*, you need to include the following files:

<project>.HLP The Help file for the application. What's This? Help is part of this file. This file is usually installed into the user's application directory.

<project>.CNT The contents file for the Help system. This file must be installed into the same directory as the .HLP file.

Note: If only What’s This? Help is being provided, the CNT file is not needed.

CSH.DLL This file contains the code to call What's This? Help. It should be placed in the user's Windows\System directory, or in the same directory as the application.

Troubleshooting

A control appears in the application but not in *What's This? Help Composer*.

It may be a dynamic control. If so, you can add it to your project using ActiveEdit.

A dialog box appears in the application but not in *What's This? Help Composer*.

It may be a dynamic dialog box. If so, you can add it to your project using ActiveEdit.

Part of a dialog appears, but not the entire dialog.

This may be part of a dialog that is placed on a common dialog at run time. You can view the entire dialog using ActiveTest.

A dialog box appears in *What's This? Help Composer* but not in the application.

It may be a dialog box that is no longer used. Check with the programmer. If this is the case, you can remove it from your project.

Wizard displays the following message: "Could not load this program file. It may be a 16-bit file. What's This? Help is not supported by 16-bit programs."

What's This? Help Composer could not read the program file. Most likely this is because it is a 16-bit application (16-bit applications do not have the capability of enabling What's This? Help). If you aren't sure, check with the application developer.

A dialog doesn't display in Dialog View.

- The dialog may be located in the Recycle Bin. Use Control View to see a list of the controls. If you add the dialog back into the active project, you will be able to view the dialog box image in Dialog View.

OR

- The dialog may be a dynamic dialog, added via ActiveTest. These types of dialogs do not display in Dialog View - use Control View to see a list of the controls.

When running the application in ActiveTest or Run Application, What's This? Help won't display for the first window of a dialog-based application.

This happens if the main window of the application is programmed as a dialog box rather than a window. You are not able to use ActiveTest and/or Run Application for this type of application. However, you can still create What's This? Help for the application with the *What's This? Help Composer.* You just need to do the proper initialization and the What's This? Help will display at run-time.

When running the application in ActiveTest or Run Application, What's This? Help won't display.

If you are working in the first window, see above. If you are working on any other window/dialog box of your application, the cause of the problem may be either of the following:

- If the dialog box has been updated in the application, and your What's This? Help has not been recompiled, What's This? Help cannot display for the updated control. If you suspect this might be the case, Update your What's This? project and compile from within *What's This? Help Composer*.
- You may be viewing a dynamic dialog box that has not yet been added to the What's This? project. If so, you can add it to your project using ActiveEdit. For additional information, see Dynamic Dialog Boxes.

When running the application in ActiveTest or Run Application, the same What's This? Help text is displayed for different controls.

- This error might occur when the "Use Microsoft Visual C++ Help IDs" option on the Configuration dialog is selected. If so, turn this option off and recompile the Help file.

Note: To turn this option off select **Configuration** under the **Project** menu and make sure the check box for this option is not marked.

When making the Help file the following error message is displayed *The map value for "IDH_Number" is the same as the map value for "IDH_Number"*

- This error message might be displayed when the "Use Microsoft Visual C++ Help IDs" option on the Configuration dialog is selected. If so, turn this option off and recompile the Help file.

Note: To turn this option off select **Configuration** under the **Project** menu and make sure the check box for this option is not marked.

Programming an Application for What's This? Help

What's This? Help Composer generates the Help topics for any 32-bit .EXE, .DLL, or .OCX. It supports applications compiled using Borland, Symantec, Microsoft or Watcom C and C++ compilers. When *What's This? Help Composer* generates the Help it also creates the files needed to link the What's This? Help to your application.

For details, see:

Getting Files From the Help Author

Adding What's This? Help to Your Application

The Importance of Tab Order in Dialog Boxes

Getting Files From the Help Author

You will need to get the following files from the Help author, and place them in your application folder.

<project>.HLP The Help file for the application. What's This? Help is part of this file.

<project>.CNT The contents file for the Help system.

Note: If only What's This? Help is being provided, the .CNT file is not needed. If the Help author has not yet created the .HLP and .CNT files, you can still add the file names to your application, and then copy the files when they are ready for testing.

CSH.DLL Contains the code to call What's This? Help.

<compiler>.LIB The library file you will link to. The one you use depends on the compiler you are using for your application.

Compiler	.LIB File
Microsoft	MSCSH.LIB
Borland	BorCSH.LIB
Symantec	SymCSH.LIB
Watcom	WatCSH.LIB

CSHUSER.H Defines functions that are available for enabling What's This? Help created using *What's This? Help Composer.* For more information, see CSHUSER.H.

CSH.DEF Module definition file, lists the imports and exports for CSH.DLL. You need this file if you are using a Borland, Watcom, or Symantec compiler.

Note: These files are stored in the Help author's WinHelp\RoboHELP\WhatsThs\ProgDlls folder.

{button ,AL(`programming',0,`',`')} Related Topics

Adding What's This? Help to Your Application

To add What’s This? Help to your C or C++ application:

1. Include CSHUSER.H.

   This file contains functions used for What’s This? Help. See details in CSHUSER.H.

2. Initialize What's This? Help (where you do your other program initializations). Note that you must Initialize What’s This? Help BEFORE you create any Windows or Dialog Boxes. For example, you could do it in InitInstance for your application class for an MFC application. For C applications, you should initialize it at the top of WinMain.

   ```
   CSHInitialize(param1,param2)
   ```

   - param1 is the name of the Help file, <project.HLP>. Note that you do not need to provide the full path to the Help file if it resides in the same directory as your executable or in the PATH.
   - param2 is a Boolean telling whether to display the What's This? menu item on the right-click menu.

3. Rename the .LIB file for your compiler to CSH.LIB.

4. Optional: You may wish to make this folder is part of the library paths used by your compiler, or copy CSH.LIB to your LIB folder. Otherwise, include the full path when you specify CSH.LIB.

5. Include CSH.LIB in your project.

{button ,AL(`programming;roles',0,`',`')} Related Topics

CSHUSER.H

This file contains functions that are used in the application program to enable the Help created by *What's This? Help Composer*.

The following functions are included:

CSHInitialize

The CSHInitialize function initializes the What's This? Help engine.

```
BOOL CSHInitialize(
    LPSTR a_pHelpFile,        // address of Help file name
    BOOL bContextMenu         // flag for What's This? menu display
    );
```

Parameters

a_pHelpFile

Points to the Help file name for the application. An application can have only one Help file. This can be a file in the PATH, a relative path, or a full path name of a Help file.

bContextMenu

What's This? Help menu flag. If TRUE the What's This? menu will display when the user right clicks on any control that has What's This? Help enabled. If FALSE the What's This? menu will not display when controls are right clicked. If bContextMenu is FALSE the F1 key still displays What's This? Help for the selected control. Use FALSE for bContextMenu to override the normal operation of WinHelp.

Return Value

If the function succeeds, the return value is TRUE.

If the function fails, the return value is FALSE. If this function fails What's This? Help will not be enabled in the application.

Remarks

Example Call:

```
// Perform initializations that apply to a specific instance
if (!InitInstance(hInstance, nCmdShow))
    return (FALSE);

// Initialize Blue Sky What's This? Help
if (!CSHInitialize("context.hlp", TRUE))
    return (FALSE);
```

CSHRegisterDialog

The CSHRegisterDialog function registers a Unique ID of your choosing for the specified dialog.

```
BOOL CSHRegisterDialog(
    HWND a_hDialog,       // window handle of the dialog to register
    long a_UniqueID,// unique ID for this dialog
    );
```

Parameter

a_hDialog

Specifies the dialog box that is being registered.

Return Value

If the function succeeds, the return value is TRUE.

If the function fails, the return value is FALSE. If this function fails What's This? Help will not be enabled in the specified dialog.

Remarks

Example Call:

```
    case WM_INITDIALOG:
        // Register What's This Help for Duplicate dialog "AboutBox2"
        CSHRegisterDialog(hDlg, (LONG)(55555));
        return (TRUE);
```

CSHContextMenu

The CSHContextMenu function displays a What's This? menu in the specified dialog.

```
BOOL CSHContextMenu(
    HWND hDlg,              // window handle of the current dialog
    HWND hWndPoint          // handle to the window in which the user right clicked the mouse
    LPPOINT pPt             // coordinates of where to display the What's This? menu
    );
```

Parameters

hDlg

Identifies the dialog box that will own the What's This? menu.

hWndPont

Identifies the window in which the user right clicked the mouse. This can be a child window of the window receiving the message.

pPt

Points to a POINT structure. This point is where the What's This? menu will be displayed.

Return Value

If the function succeeds, the return value is TRUE.

If the function fails, the return value is FALSE. If this function fails the What's This? menu will not display.

Remarks

Example Call:

```
    case WM_CONTEXTMENU:
    {
        POINT pt = {LOWORD(lParam), HIWORD(lParam)};
        HWND hWndPoint = (HWND)wParam;

        if (hWndPoint == hWndOK || hWndPoint == hWndCan)
        {
            CSHContextMenu(hDlg, hWndPoint, &pt);
        }
        else if (hWndPoint == hWndBtn1)
        {
            if ((hMenu = LoadMenu(hInst, "RTTEST1")) != NULL)
            {
                if ((hMenuTrack = GetSubMenu(hMenu, (int)0)) != NULL)
                {
                    TrackPopupMenuEx(hMenuTrack, TPM_LEFTALIGN | M_RIGHTBUTTON,
                                        pt.x, pt.y, hDlg, NULL);
                }

                DestroyMenu(hMenu);
            }
        }
    ...
```

CSHContextHelp

The CSHContextHelp function displays What's This? Help for the specified control in the specified dialog.

BOOL CSHContextHelp(
    HWND *hDlg*,   // window handle of the current dialog
    HWND *hCtrl*   // window handle of the control for which to display What’s This? Help
    int *iContextType* // from HELPINFO structure
    );

Parameters

hDlg

Identifies the dialog box that is the parent window of the control for which to display What’s This? Help.

hCtrl

Identifies the dialog box control for which to display What’s This? Help.

Return Value

If the function succeeds, the return value is TRUE.

If the function fails, the return value is FALSE. If this function fails What’s This? Help will not display for the specified control.

Remarks

Example Call:

```
case ID_MYWHATSTHIS:
    CSHContextHelp(hDlg, GetDlgItem(hDlg, IDC_BUTTON1), 0);
    break;
```

{button ,AL(`programming',0,`',`')}   Related Topics

Setting the Tab Order

The Tab sequence of the controls in a dialog box, while helpful for users, is critical for What's This? Help. It defines the order of Help topics as well as the links between Help topics. For example, if you set the ID to -1 for a control, it is not included in the Tab order, and it cannot have its own Help topic.

To make the What's This? Help creation process easier for both you and the Help author, here's what you can do:

- For static text, set the control ID to -1. In addition, the control it describes should follow it in the order.
- For a group box, use a unique ID, or use -1. The -1 allows the next control in the Tab order to use the same Help text as the group box.
- Make sure the Tab order makes sense, and flows logically according to generally accepted programming standards - it will save you time in the long run.

{button ,AL(`programming',0,`',`')} Related Topics

Creating What's This? Help in Visual Basic Applications

If you are creating What’s This? Help for applications developed in Visual Basic, the procedure is slightly different than that involved with other development environments, such as C++. Some of these differences are discussed below:

- *What’s This? Help Composer* works differently in the Visual Basic environment. Rather than gathering information from executable files and storing it in its own project files, *What’s This? Help Composer* adds information to Visual Basic FRM and VBP files. For example, the line

  ```
  WhatsThisButton =   -1              'Auto-Generated By What’s This? Help Composer
  ```

  inserted into an FRM file, enables the ? button in a given dialog box of your Help project.

  Note: The text 'Auto-Generated By *What’s This? Help Composer*, is a comment, not a part of the data added by *What’s This? Help Composer*.

- Visual Basic Help projects do not have a dialog view in *What’s This? Help Composer*. This presents no real limitation however, as you can as easily compose Help text for any Visual Basic dialog in Control View.

- To test Help in the Visual Basic environment you must

1. Compile the Help project.
2. Run the application.
3. Run the Help project from within the application.

In other respects, operation of *What’s This? Help Composer* in the Visual Basic environment is mostly the same as in the other supported environments.

Help Author Report

The *What's This? Help Composer* can generate a Help Author report to let you track the progress of the individual controls of your Help project.

The Help Author report displays or prints the following information:

- Date the report was generated
- Name of the project file
- Range of the report, i.e., current dialog, current file or entire project.
- Individual reports on each control in your project

To generate and display the Help Author Report,

1. From the **File** menu, click **Report**, or click 📓 in the icon bar. The Project Reports dialog appears.
2. From the Report list at the upper left, select **Help Author Report**.
3. Select the range for which you wish to generate the report. This can be one of three options:
    - **Current dialog.** Generates a report on the dialog currently highlighted in the *What's This? Help Composer*'s **Dialog Boxes:** window.
    - **Current file.** Use this option to generate a report for only the current file of a multi-file Help project.
    - **Entire Project.** Generates a report on all dialogs in all files of the current Help project.

When you have selected the desired options,

- Click **Run**. *What's This? Help Composer* generates the report according to your request and displays it in the Project Report dialog. If you would like a hard copy of the report,
- Click **Print**.

Project Status Report

The Project Status Report gives you the status of your project's Help Topics at a glance, displaying or printing the following information:

- Help Topics Marked Generated
- Help Topics Marked In Progress
- Help Topics Marked For Review
- Help Topics Marked Complete
- Total Topics (the sum of the above categories)

To generate and display the Project Status Report,

1. From the **File** menu, click **Report**, or click 📗 in the icon bar. The Project Reports dialog appears.
2. From the Report list at the upper left, select **Project Status Report**.
3. Select the range for which you wish to generate the report. This can be one of three options:
   - Current dialog
   - Current file
   - Entire Project
4. Click Run.

Note: If the total number of Topics in the Project Status Report and the number of dialog controls in the *What's This? Help Composer*'s status line (lower right hand corner) disagree, it is because the status line counts all of the controls in a dialog, while the Project Status Report counts only those controls which support context help. For example, an "About…" box typically contains dates, version information, publishers' names (each an instance of static text, thus non-supportive of context help) and an **OK** button (which does support context help). In such a case the status line shows "four Dialog Controls," while the Project Status Report shows one Total Topic.

New Project Wizard

Program File Enter the name of the program file (or select using **Browse**) for which you want to generate What's This? Help. This can be any 32 bit executable (.EXE), a Visual Basic applications (.VBP) a dynamic link library (.DLL), or an OLE control (.OCX) created in C/C++ and compiled with either Visual C++, or the Borland, Symantec or Watcom compilers.

Browse You can choose the **Browse** button to look though your system's folders and find the program file.

After you have selected or entered the program file name, click **Next** to continue to the next Wizard screen.

New Project Wizard

Project Type

If you started *What's This? Help Composer* independently (outside of RoboHELP), you have the option of attaching the What's This? project to the RoboHELP project, resulting in one HLP file. If you want to do this, click **Attach** and then enter (or select using **Browse**) the RoboHELP project filename. If you want your project to contain only What's This? Help, choose **Standalone**.

Filename

Enter a name for the file that will contain the What's This? Help. *What's This? Help Composer* will generate several files, all using this name, with different extensions. The default filename is Context.

Location

Enter the location (or select using **Browse**) for the What's This? Help files. If you enter a folder name that doesn't exist, *What's This? Help Composer* creates a new folder. The default location is the folder that contains the program file.

Browse

You can choose the **Browse** button to look though your system's folders and select a location for the Help files.

After you have selected or entered the filename and location, click **Next** to continue to the next Wizard screen.

New Project Wizard

Other Files

If there are any files associated with the program file you selected, they list in this screen of the Wizard. Files listed here are those that are either located in the same directoy as the program file, or called by the program file. Help will be generated for all files in this list. You can add files to the list or remove files from the list - see below. If you aren't sure which files should be included, refer to the application developer.

Note: In most cases, you will want to create a separate Help project for each executable (.EXE) file. *What’s This? Help Composer* allows more than one EXE in a project, but ActiveTest will ony run the main program file for testing (you will not have access to ActiveTest for the additional EXE files). Other types of files will be accessible via ActiveTest, so you can add as many of those as you need to.

To add a file to the list, click **Add** and select the file.

To remove a file from the list, select the file and click **Remove**. You can add the file to the project later if needed, using Add File (Project menu).

After you have selected the files, click **Next** to continue to the next Wizard screen.

New Project Wizard

Default Help Text
If you want *What's This? Help Composer* to generate default Help text for most common controls, select this check box. The generated Help text is generic, and you can edit the text for each topic after you complete the Wizard.

Set up Forms for What’s This Help
This option is only displayed for Visual Basic projects. If selected, What’s This? Help Composer will automatically format all Forms to display the What’s This question mark in the upper right corner. This selection will also set all Forms to Fixed Size.

After you have made your selection, click **Finish** to generate the What's This? Help.

Application has not been configured

This message is just a reminder that the programmer needs to link the What's This? Help into the application. You can test your Help project but the links must be coded in before the product ships - otherwise the users won't be able to access your Help.

What's This? Help Composer simulates the program to Help links so that you can test your Help before the programmer has coded the links. However, you should make one last pass through your Help system, testing the actual links, before shipping the product.

Updating while running Visual Basic

During the upgrade, What’s This? Help Composer may need to write to the Form files that has changed. Therefor, running the project simultaneously in Visual Basic is not recommended.

ActiveEdit

Jumps right to the Help source, allowing you to add dialog boxes and controls to What's This? Help and to edit existing What's This? Help text as you go through the application in ActiveTest.

Note: ActiveEdit is not available in Visual Basic projects.

ActiveTest

A powerful feature that allows you to interactively run the application and your Help system, before the programmer has added the Help links to the application. If you see a topic that needs editing, you can quickly display the Help source and make changes using ActiveEdit.

Note: ActiveTest is not available in Visual Basic projects.

Button

A control that, when clicked, performs an action. OK and Cancel are typical buttons that appear on dialog boxes. Buttons generally need Help text written for them.

Note: *What’s This? Help Composer* generates Help text for the OK and Cancel buttons if default Help text is turned on.

Check Box

A type of control that allows the user to select an option to use. In a group of check boxes, the user can select more than one option.

Combo Box

A box that contains a list of selectable items. The user may either select from the list or type in the entry.

Control

A unit of information in a dialog box. Controls are used to tell a user what a field is for (static text), to allow input, or allow the user to select an option (such as a radio button, check box, or button).

Database

The type of storage used by *What's This? Help Composer*. When you enter topic text, the text is stored in the database. Then, when you compile, the RTF is generated.

Note: If you make changes to the RTF file in RoboHELP, the database and the RTF file are not synchronized. If you compile again from *What's This? Help Composer* the changes made in the RTF file will not be reflected, as the file is regenerated from the database. See Doing Final Edits in RoboHELP for more information.

Dynamic Control

A control that is created when the dialog box is displayed at run time. It usually contains information that is passed to the control from the program, based on the user's input.

Dynamic Dialog Box

A dialog box that is created when the application runs. Its contents usually contain information that is passed to the dialog box from the program, based on the user's place in the program. The contents may also contain information that could be different for each user's system, such as a directory name.

Edit Control

A box where the user can type or edit text. Edit controls can have additional features such as the ability to scroll, and they can contain multiple lines of text.

Frame

A box that encloses two or more related controls, such as radio buttons. A frame is usually linked to the first control in the group and does not need to have its own Help text.

Group Box

A box that encloses two or more related controls, such as radio buttons. A group box is usually linked to the first control in the group and does not need to have its own Help text.

Horizontal Scroll Bar

A scroll bar that allows the user to move the contents of the window horizontally (across).

Image

A graphical control that is used to display a picture on the dialog box.

Linked Controls

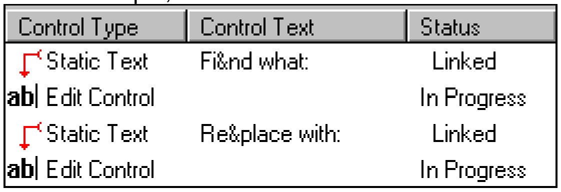
If a control shows with a red arrow in Control view, it is linked to another control. The control that it is linked to follows and is pointed to by the arrow. If a control is linked to another control, you do not need to write Help text for it as it will display the Help text for the control it is linked to.

Examples

Static text (simple link)
Group box (multiple links)

Linked Example 1

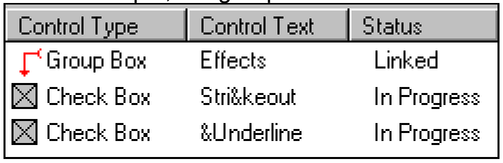
In this example, static text is linked to the control it describes.



Control Type	Control Text	Status
Static Text	Fi&nd what:	Linked
Edit Control		In Progress
Static Text	Re&place with:	Linked
Edit Control		In Progress

Linked Example 2

In this example, the group box is linked to the first control in the group.



Control Type	Control Text	Status
Group Box	Effects	Linked
☒ Check Box	Stri&keout	In Progress
☒ Check Box	&Underline	In Progress

List Box

A box that contains a list of selectable items. The user must select from the list (they cannot type in their selection.)

List View Control

A window that displays a list of selectable items. The user must select from the list (they cannot type in their selection.)

Owner Draw Button

A type of button that is created by the application at run time.

Program File

One of several files that comprise a working application.

What's This? Help Composer can generate Help for any of the following types of program files:

- executable (.EXE)
- dynamic link library (.DLL)
- OLE control (.OCX)

The main program file is the one that runs your application - usually an .EXE file.

Progress Bar Control

A control that is located in the status bar and shows the progress of an operation.

Project File

A file that is part of the *What's This? Help Composer* project.

Radio Button

A type of control that allows the user to select an option to use. In a group of radio buttons, the user can select only one option.

Rectangle

A type of control used to specify an area of a dialog box, such as a formatting area, a scroll area, or a client repaint area.

Resource File

A file in the application that stores the dialog box definitions (it stores other things but dialog boxes are what we're most interested in). *What’s This? Help Composer* reads the resource file to generate the What's This? Help topics.

Slider Control

A type of control that, when the user moves the slider left/right (or up/down), the value changes. For example, a slider control may be used to change the sound volume on your PC.

Sort Order

By default, the list of controls (in Control View) is sorted by topic ID. You can click a heading label to sort the list by that type. For example, click Status to list the controls in order of status. Note that linked controls always stay together.

This sort order also affects navigation - if you sort by status and then choose Next Control, the next control in order of status will display.

Static Text

Text that displays on the dialog box and cannot be changed by the user.

Most often static text is a label for another type of control. In this case it may be linked to the control it describes and you do not need to write Help for it. If static text is not linked to the control it describes, you can copy the text from that control to the static text control.

Tab Control

A type of control that has two or more "tabs." When the user selects a tab, the display changes. The *What’s This? Help Composer* Compile Settings dialog box contains a tab control.

Tab Order

The order in which controls are selected when the user Tabs through a dialog box. The tab order is set by the programmer. If a control is linked to another control, they appear as one in the tab order (the user cannot select the linked control). Linking is most often used for controls such as static text that do not require (or allow) any user interaction.

Tree View Control

A control that lists information in tree view, similar to Windows 95 Explorer. The Dialog Boxes list in *What’s This? Help Composer* is an example of a tree view control.

Unused Dialog Box

A dialog box that is included in your What's This? Help project, but that isn't used in the application. This type of dialog box can exist for several reasons, such as the development tool creates it, but the programmer chooses not to use it. You can remove these dialog boxes from your active project, storing them in the Recycle Bin. See Unused Dialog Boxes for further details.

Vertical Scroll Bar

A scroll bar that allows the user to move the contents of the window vertically (up and down).

New Project

File - New

Starts the New Project Wizard so you can create a new What's This? Help project.

Open Project

File - Open

Allows you to select an existing What's This? Help project to open in *What’s This? Help Composer*.

Save As RTF

File - Save As RTF

Saves the current What's This? Help project in RTF format. You can do this as a precaution but it isn't necessary - when you compile What's This? Help, the RTF document is automatically generated.

Note: Each time you move to a new control/topic, the previous control's Help text is automatically saved in the database - you do not need to Save to save your Help text.

Print

File - Print

Prints the current Help topic, including control and dialog box information as well as the topic text.

Reports

File - Reports

Opens the Reports dialog, allowing you to generate, view and print Help Author Reports and Project Status Reports.

List of Files

The previous four project files list on the File menu. You can click a project to open it, skipping the File-Open selection.

Exit What's This? Help Composer

File - Exit

Closes *What’s This? Help Composer*. All data is saved automatically as it is added so you are not prompted to save.

Undo

↶ Edit - Undo

Undoes (reverses) the last edit action.

Cut

Edit - Cut

Removes the selected text, saving it to the Windows clipboard.

Copy

Edit - Copy

Makes a copy of the selected text, saving it to the Windows clipboard.

Paste

Edit - Paste

Inserts the contents of the Windows clipboard.

Delete

Edit - Delete

Deletes the selected item or text.

If a dialog box is selected and the Dialog Boxes list is the active window, this option becomes Remove Dialog, indicating that the dialog box will be removed from the project and placed in the Recycle Bin.

If a file is selected and the Dialog Boxes list is the active window, this option becomes Remove File, indicating that the entire file (and all dialog boxes within it) will be removed from the project and placed in the Recycle Bin.

Select All

Edit - Select All

Selects all of the text/graphics in the Help Text window. Note that this option is only available if the insertion point is in the Help Text window, making it the active window.

Insert Graphic

Edit - Insert Graphic

Opens the Insert Graphic window so that you can add a graphic to your Help topic.

Replace Text

▭ Edit - Find and Replace

Opens the *What’s This? Help Composer* Find and Replace window so that you can replace text for the entire What's This? Help project.

Copy Help Text

Edit - Copy Help Text

Copies all of the text/graphics in the Help Text window of *What's This? Help Composer* to the clipboard. You can then paste the text into another topic using the Edit - Paste Help Text command.

This is most often used when you want more than one topic to contain the same text, such as for a static text field that serves as a label for an input field, but isn't linked (linked controls do not need their own Help text.)

Paste Help Text

Edit - Paste Help Text

Pastes the contents of the clipboard into the current Help topic. This command is only available if the Edit - Copy Help Text was previously used to copy the text of another topic to the clipboard.

This is most often used when you want more than one topic to contain the same text, such as for a static text field that serves as a label for an input field, but isn't linked (linked controls do not need their own Help text.)

Restore Recycled

Edit - Restore Recycled

Removes the selected dialog box from the Recycle Bin, placing it back into the active project. This option is only available if a dialog box or file in the Recycle Bin is selected in the Dialog Boxes list. If a file is selected, all dialog boxes in the file are restored to the active project.

Topic Status

Edit - Generated, In Progress, For Review, Complete

The check mark indicates the status of the Help text for the current control. You can change the status of a topic by selecting the new status. Note that Generated is not selectable as it means that what Note that *What's This? Help Composer* generated the topic text but it has not been edited. As soon as you make a change, the topic must at least be In Progress.

Toolbars

View - Toolbars

Toggles display of the Standard and Formatting toolbars. A check mark indicates the toolbar is displayed.

Status Bar

'View - Status Bar

Toggles display of the *What's This? Help Composer* status bar, located at the bottom of the screen. A check mark indicates the status bar is displayed.

Dialog View

View - Dialog View

Changes the display to dialog view, showing an image of the dialog box being documented.

See Also

Working in Dialog View

Control View

▭ View - Control View

Changes the display to control view, showing a list of the controls in the dialog box.

See Also

Working in Control View

Linked Controls

View - Linked Controls

Toggles display of linked controls on/off. Since you don't need to write Help text for linked controls, it can make it easier to work with their display turned off.

Expand Dialog

View - Expand Dialog

Expands the selected dialog box. This option is only available if the currently selected dialog box is expandable (for example, it may contain a "More" button that shows more selections).

Next Control

View - Next Control

Displays the Help text for the next control in the sort order.

Previous Control

View - Previous Control

Displays the Help text for the previous control in the sort order.

Next Dialog

View - Next Dialog

Control View - Displays the list of controls for the next dialog box in the Dialog Boxes list.

Dialog View - Displays an image of the next dialog box in the Dialog Boxes list.

Previous Dialog

View - Previous Dialog

Control View - Displays the list of controls for the previous dialog box in the Dialog Boxes list.

Dialog View - Displays an image of the previous dialog box in the Dialog Boxes list.

Expand All

View - Expand All

Expands the Dialog Boxes list, listing all dialog boxes.

Collapse All

View - Collapse All

Collapses the Dialog Boxes list, displaying only the module names.

Normal

Format - Normal

Formats the selected text as normal (no bold, italics, or underlining).

Bold

B Format - Bold

Formats the selected text as bold.

Italic

Format - Italic

Formats the selected text as italic.

Underline

U Format - Underline

Formats the selected text as underlined.

Project Help Text Font

Format - Project Help Text Font

Opens the Help Text Font dialog box so you can set the font type and size to be used for all Help text in your What's This? project.

Alignment

Format - Alignment

Displays the alignment menu so you can choose the alignment for the currently selected paragraph.

Left

Format - Alignment - Left

Left aligns the selected paragraph.

Center

Format - Alignment - Center

Centers the selected paragraph.

Right

Format - Alignment - Right

Right aligns the selected paragraph.

Compile

Project - Compile

Generates the source files and runs the WinHelp compiler for the current project.

Run Application

Project - Run Application

Runs the application (main program file).

Note: If you want to test What's This? Help, you can use ActiveTest - this is the recommended way to test as you can edit the Help text (using ActiveEdit) as you run the application.

Run Help

Project - Run Help

Runs the current Help project (.HLP file). If you are running *What's This? Help Composer* from within RoboHELP, the RoboHELP project displays.

Compile Settings

Project - Compile Settings

Displays the Compile Settings dialog box so that you can specify some of the compile options directly from within *What's This? Help Composer*.

Configuration

Project - Configuration

Displays the Configuration dialog box so that you can specify setup options for the What's This? Help project.

WinHelp Mode

Project - WinHelp Mode

Lets you select between Normal and Help Author modes for viewing your Help project. When you view the Help system in Help Author mode, the topic number displays in the title bar, and you can see the jump text.

ActiveTest

Project - ActiveTest

Runs the application in ActiveTest mode, allowing you to test your What's This? Help before the programmer has linked it to the application. You can also use ActiveTest to find dialog boxes and controls for which What's This? Help was not automatically generated (special cases, such as dynamic dialog boxes or controls).

Update Project

Project - Update Project

Displays the Update Project dialog box, allowing you to select files to be updated in your project.

Add File

Project - Add File

Displays the Add Program File dialog box, so you can select a file to be added to the current project. Use this button/option if you find that a file was not included when the project was originally generated. You can add an additional DLL, OCX, or EXE file.

Relocate File

Project - Relocate File

Opens the Locate File dialog box so that you can select a new location for the program file. This option is only available if a file is selected in the Dialog Boxes list.

Empty Recycle Bin

Project - Empty Recycle Bin

Deletes all items in the Recycle Bin.

Note: **Be careful about deleting items from the Recycle Bin!** If you remove a dialog box and then update your project, it may be added into your project as "New." See Using the Recycle Bin for more information.

Spelling

Tools - Spelling

Opens the Project Spell Check dialog box so you can spell check the current control, current dialog box, or the entire What's This? project.

Spell Check Current Control

Tools - Spell Check Current Control

Spell checks the Help text for the current control. Note that this option is only available if the insertion point is in the Help Text window, making it the active window.

Help Topics

Displays the Help Topics screen of the *What's This? Help Composer* Help system. From the Contents you can access any area of the Help system.

About

Displays version information for *What’s This? Help Composer*.

Closes this dialog.

This section displays a preview of the selected graphic.

Select the file name from the list or type it into this field.

Select the file name from the list or type it into this field.

Select the file to be opened.

Select the folder where the file is located.

To open a What's This? Help Composer file, this type must always be set to .CHJ.

To open a What's This? Help Composer file, this type must always be set to .CHJ.

Click the down arrow to the right of the list box to display a list of filename extensions identifying the file types available.

Select the drive (local or on the network) where the file is located.

Select the drive (local or on the network) where the file is located.

Closes this dialog, saving any changes you have made.

Closes this dialog without saving any changes you have made.

Displays context-sensitive Help for this dialog box.

If the file you wish to open is located on a network, click this button to select the network drive.

This section displays a preview of the selected graphic.

Select the file name from the list or type it into this field.

Select the file name from the list or type it into this field.

Select the file to be opened.

Select the folder where the file is located.

To open a What's This? Help Composer file, this type must always be set to .CHJ.

Select the file type to be opened.

Click the down arrow to the right of the list box to display a list of filename extensions identifying the file types available.

Select the drive (local or on the network) where the file is located.

Select the drive (local or on the network) where the file is located.

Closes this dialog, saving any changes you have made.

Closes this dialog without saving any changes you have made.

Displays context-sensitive Help for this dialog box.

Select this option to open the file as read only (you can't make changes to it).

If the file you wish to open is located on a network, click this button to select the network drive.

Choose the default font for all context-sensitive Help topics.

Choose the default font size for all context-sensitive Help topics.

Use the arrows to select a font size.

Closes this dialog, saving any changes you have made.

Closes this dialog without saving any changes you have made.

Select to spell check all controls in all dialogs in the project.

Spell check only the current dialog.

Spell check only the current control.

Starts the selected spell check.

Closes this dialog and cancels the spell check.

Select an option to specify the extent of the spell check.

Rescans the selected files and updates the database.

Closes this dialog without saving any changes you have made.

Selects all files listed.

Lists the program files that are included in your project. You can select any or all of these files to be updated.

Check this option if you don't want to be reminded to update again in this session.
Note: This option will be reset next time you start What's This? Help Composer.

Lists the path and name of the file, as currently defined.

Enter the new path and name of the file. You can use Browse to search for the location.

To browse your system for the new location, click Browse.

Closes this dialog, saving any changes you have made.

Closes this dialog without saving any changes you have made.

Lists the path and name of the file, as currently defined.

Displays the results of the project update.

Closes this dialog, saving any changes that the update has made.

Closes this dialog without saving any changes that the update has made.

Copies the report to the clipboard.

Displays the Print dialog box so that you can print the report.

Displays the results of the project update

Choose the default font for all What's This? Help topics.

Choose the default font for all What's This? Help topics.

Choose the default font size for all What's This? Help topics.

Choose the default font size for all What's This? Help topics.

Closes this dialog, saving any changes you have made.

Closes this dialog without saving any changes you have made.

Applies the current font style and size to the selected text.

Displays context-sensitive Help for this dialog box.

Formats the selected text as strikeout.

Underlines the selected text.

Shows how the text will appear in the selected font and size.

Enter the text that you want to find.

When the Find What text is found, it will be replaced with the text entered here. To remove the Find What text, leave this field blank.

Select this option to search for occurrences that are whole words, not part of a larger word.

Select this option to distinguish between upper and lower case characters when performing the search.

Search for the next occurrence of the text entered in Find What.

Replace the found text with the text in the Replace With field.

Replace all occurrence of the Find What text with the text in the Replace With field. What's This? Help Composer will not ask for confirmation before replacing text.

Closes this dialog without saving any changes you have made.

Displays context-sensitive Help for this dialog box.

Closes this dialog, saving the check box setting

Closes this dialog. If you have changed the check box setting, the change will not be saved.

Dislays context-sensitive Help for this dialog box.

Select this option to hide this dialog box in the future, preventing this reminder from displaying.

Choose an item from the list.

Choose an item from the list.

Select the type of report you wish to generate. Help Author Report gives Control Type, Control Text, Status and Help Topic text of each control within the selected Report Range and Status parameters. Project Status Report gives statistics on the status of all controls within the selected Report Range parameter.

Select the range within the current project for which you wish to generate a report.

Generates a report on the currently selected dialog only.

Generates a report on the currently selected file only.

Generates a report on the whole project.

Select controls by specific status to be included in the Help Author report. (These selections are not available in Project Status Report.)

Include all controls regardless of the status.

Include only generated controls in the report.

Include only controls marked "In Progress" in the report.

Include only controls marked "For Review" in the report.

Include only controls marked "Complete" in the report.

Click here to close the report dialog.

Click here to generate a report.

Copy the report to the clipboard. (You can then paste it into other documents.)

Send the report to the printer.

When you generate a report, it is displayed here.

Quits the print job and closes the dialog.

Enter your input in this field.

Click this button to make your choice.

Click this button to make your choice.

Closes this dialog without saving any changes you have made.

Previews the selected report.

Prints the selected report.

Select the desired report from this list.

Continues with the spell check without chaning the word.

Bypasses all instances of the word.

Displays a possible error, such as a misspelled word, duplicate word, or incorrect capitalization.

Type corrections here or choose the correct word from the suggested list below.

Changes the word to the word listed in the Change To box.

If the Change To box is empty, this button name changes to Delete. Use Delete to delete the selected word.

Changes all instances of the word, replacing them with the word in the Change To box.

If the Change To box is empty, the button name changes to Delete All. Use Delete All to delete all instances of the selected word from the document.

Adds the word to the dictionary.

Displays a list of proposed alternatives for a misspelled word.

When you click Suggest, lists possible replacement words from the main dictionary and open custom dictionaries.

Closes this dialog without saving any changes you have made.

Displays the name of the Help file to be created.

The copyright statement to display in your Help system's About box.

Click this button to insert a copyright (c) symbol into the copyright statement.

The copyright statement to be included when Help text is copied to the clipboard or printed.

Click this button to insert a copyright (c) symbol into the copyright statement.

Select this option to see messages on the screen during the Help compile.

Select this option to display informational notes during the compile. Notes describe events that will not prevent the Help compiler from running or affect the integrity of compiled Help file. Displaying notes can slow down the compiler.

Select this option to automatically close HCW (if you selected to display the window) and return to *What's This? Help Composer* after compiling.

Select this option to display the WinHelp compiler window during the compile. If you select this option, you can also choose to close the window when the compilation has finished.

Select from the list of possible compression levels. You can also choose **Custom** to select a customized version of the compression.

Click this button to select a custom compression setting.

The location of the Help compiler (HCW.EXE). To specify a new location, enter the location or select it using **Browse**.

Click this button to look though your system's folders and select a location for the Help compiler.

The location of the Help compiler (HCRTF.EXE). To specify a new location, enter the location or select it using **Browse**.

Click this button to look though your system's folders and select a location for the Help compiler.

The location of the TMP folder used to store temporary files during the compile. To specify a new location, enter the location or select it using **Browse**.

Click this button to look though your system's folders and select a location for the TMP folder.

The location of the file used to store error messages during the compile. To specify a new location, enter the location or select it using **Browse**.

Click this button to look though your system's folders and select a location for the error log.

Click this button to look though your system's folders.

To remove a folder from the list, select it and then click this button.

Lists the folders that contain graphics used in the Help system.

Click this button to make your choice.

Click this button to make your choice.

Choose an item from the list.

Choose an item from the list.

Click this button to make your choice.

Enter your input in this field.

Choose an item from the list.

Click this button to make your choice.

Choose an item from the list.

Click this button to make your choice.

Enter your input in this field.

Click this button to make your choice.

Click this button to make your choice.

Choose an item from the list.

Check your desired options.

Choose an item from the list.

Check your desired options.

Check your desired options.

Choose an item from the list.

Check your desired options.

Check your desired options.

Enter your input in this field.

Choose an item from the list.

Click this button to make your choice.

Enter your input in this field.

Click this button to make your choice.

Closes this dialog, saving any changes you have made.

Closes this dialog without saving any changes you have made.

Select this option to use Phrase compression when compiling the Help system. This option is recommended for files under 100K. It may not result in maximum compression.

Select this option to reuse the Phrase (.PH) file rather than generating it for each compile. This can speed up the compile but may not provide maximum compression.

Select this option to use Hall compression when compiling the Help system. This option is recommended for files larger than 100K, and is most effective when combined with Zeck compression.

Select this option to use Zeck compression when compiling the Help system. This option is most effective when combined with Hall compression.

The name of the main program file. This is the file that *What's This? Help Composer* reads to generate the Help topics. This name cannot be changed once the project is created.

The name of the file to run when testing the program. In most cases this will be the same as the Main Program File. You may change this if you are writing What's This? Help for a .DLL or .OCX that is attached to an .EXE which you have not included in your Help project.

Click this button to look though your system's folders and select a program file.

The name of the compiled Help file.

Click this button to look though your system's folders and select a Help file.

What's This? Help Composer generates a map number for each topic. You can specify a different starting map number.

The highest map number currently existing in the project.

If you want all of your topic IDs to have a specific prefix, enter it here. The default is IDH_.

If this option is selected, *What's This? Help Composer* opens the last opened project at startup. If this option is not selected, *What's This? Help Composer* will starts up with a blank screen.

Select this option if you want *What's This? Help Composer* to prompt for confirmation before removing a dialog box from the project (placing it in the Recycle Bin).

Closes this dialog, saving any changes you have made.

Closes this dialog without saving any changes you have made.

If you want *What's This? Help Composer* to generate default Help text, select this check box. The generated Help text is generic, and you can edit the text for each topic after you complete the Wizard (or update).

The name of the main program file. This is the file that *What's This? Help Composer* reads to generate the Help topics. This name cannot be changed once the project is created.

The highest map number currently existing in the project

Select this option if you want *What's This? Help Composer* to use the Help ID's created by Microsoft Visual C++. These Help ID's are stored in the .HM file and created by Microsoft Visual C++ for controls that has the "Help ID" check box selected in the **Control Properties** dialog.
NOTE: Be aware that if this option is selected all controls with the same ID might use the same Context Help text.

This message is just a reminder that the programmer needs to link the What's This? Help into the application. You can test your Help project but if the program is shipped without the links, the users won't be able to access your Help.

Closes this dialog, saving the check box setting.

Closes this dialog. If you have changed the check box setting, the change will not be saved.

Displays context-sensitive Help for this dialog box.

Select this option to hide this dialog box in the future, preventing this reminder from displaying.

Enter the name of the program file (or select using Browse) for which you want to generate What's This? Help. This can be an executable (.EXE), a dynamic link library (.DLL), or an ActiveX (OLE) control (.OCX).

Click this button to look though your system's folders and find the program file.

You can choose any name and location for the What's This? document. The default filename is Context, and the default location is the folder where your RoboHELP project is stored.

Enter a name for the file that will contain the What's This? Help. What's This? Help Composer will generate several files, all using this name, with different extensions. The default filename is Context.

Enter the location (or select using Browse) for the What's This? Help files. If you enter a folder name that doesn't exist, *What's This? Help Composer* creates a new folder.

Click this button to look though your system's folders and select a location for the What's This? Help files.

If you want to create a new folder, type in the full path and folder name.

Select this option to create a What's This? Help project that is not part of a RoboHELP project.

Select this option to create What's This? Help as part of an existing RoboHELP project.

Enter the name (or select using Browse) of the existing RoboHELP project.

Click this button to look though your system's folders and select a RoboHELP project.

Enter a name for the file that will contain the What's This? Help. What's This? Help Composer will generate several files, all using this name, with different extensions. The default filename is Context.

Enter the location (or select using Browse) for the What's This? Help files. If you enter a folder name that doesn't exist, What's This? Help Composer will create the folder.

Click this button to look though your system's folders and select a location for the What's This? Help files.

You can create What's This? Help as part of a RoboHELP project, resulting in one HLP file that contains all the Help text. Or you can create a separate HLP file that contains the What's This? Help.

If you want to create a new folder, type in the full path and folder name.

The name of the main program file for the application you are documenting.

Lists files associated with the program file you selected. Files listed are those that are either located in the same folder as the program file, or called by the program file. The files in this list will be included in the Help project.

Click this button to add a file to the list.

To remove a file from the list, select the file then click this button.

Enter your input in this field.

Any files that you include will have What's This? Help created for them. You can remove the files from your project later, so if you're not sure, it's better to add them. (You can also add them later if needed.)

If you want *What's This? Help Composer* to generate default Help text, select this check box. The generated Help text is generic, and you can edit the text for each topic after you complete the Wizard.

Select this check box if you want *What's This? Help Composer* to automatically change all Forms in the project to display the What's This? Help icon in the upper right corner. Selecting this option will also change all Forms to be Fixed Size.

Select this check box if you want *What's This? Help Composer* to automatically change all Forms in the project to display the What's This? Help icon in the upper right corner. Selecting this option will also change all Forms to be Fixed Size.

Choose an item from the list.

Will continue to update all accessible forms.

Closes this dialog canceling the update.

eHelp Corporation headquarters

eHelp Corporation is the worldwide leader in Help authoring solutions. Software developers, Help authors, technical writers, and documentation experts worldwide turn to eHelp Corporation for state-of-the-art Help authoring and Internet/intranet solutions.

Corporate Headquarters

Address:	eHelp Corporation 7777 Fay Avenue, Suite 201 La Jolla, CA 92037 U.S.A.
Telephone:	1-858-551-2485
Fax:	1-858-551-2486
World Wide Web:	www.blue-sky.com
Office Hours:	Monday to Friday 6:00 A.M. to 5:00 P.M. Pacific Time

Sales

eHelp Corporation's professional sales staff is ready to answer your sales questions Monday to Friday from 6:00 A.M. to 5:00 P.M. Pacific Time.

Contact your sales representative to hear about the latest products, the Subscription Plan, training, upgrade options and prices, and more. If you have a technical question or problem, please contact Technical Support.

Contacting your eHelp Corporation Sales Representative

Sales Phone:	1-800-677-4946
International (including Canada):	1-858-459-6365
Sales Fax:	1-858-459-6366
World Wide Web:	www.blue-sky.com
Sales Email:	sales@blue-sky.com

When sending email, please include your company name, city, and state to ensure your request is handled promptly.

eHelp Corporation Web site

{ewl RoboEx32.dll, WinHelp2000, HREF="www.blue-sky.com"
ALTERNATE="Blue_Sky_Software_WinHelp2000_Alternate"}

Technical Support

At eHelp Corporation, we work hard to provide you with intuitive technical products. Additionally, we try to provide superior online and print documentation to enable you to work independently. If you have a technical question that you can't answer with the provided tools, please contact our Technical Support department. All of our Technical Support representatives are eager to answer your questions and help you create the best Help systems possible.

Note: You must be a registered user to access eHelp Corporation support services. eHelp Corporation support services are subject to the prices, terms, and conditions in place at the time the service is used.

eHelp Corporation's Technical Support services includes email, telephone, and Web-based support on our corporate Web site.

Help Us Help You

When contacting Technical Support via phone or email, please provide the following information for the fastest possible service:

- Product serial number (registration number)
- Your name, company name, and phone number
- Exact product name and version number
- Type of operating system (e.g., Windows 95)
- Version of Microsoft Word (if using RoboHELP for WinHelp)
- Browser and version (if creating HTML-based Help)
- Complete description of the issue, including steps to reproduce it
- Exact wording of any messages displayed when you encountered the problem
- Steps taken to resolve the problem
- All previous email threads with eHelp Corporation about the issue, if any

For current hours of operation and details about all support offerings, please:

- Visit our Web site.
- Email us at **supportnews@blue-sky.com** to receive an automated reply listing all of your support options.
- Call us at 858-551-2194.

eHelp Corporation Knowledge Base

{ewl roboex32.dll, ActiveWinHelp, CLSID="E9B53FC3-761E-11D0-AF02-00009290C4DB"
ALTERNATE="Blue_Sky_Software_HelpDesk_Alternate" CODEBASE="HelpDesk.ocx"
PROPERTIES="00000100562f0000c80b0000000000008044524956455 23d7b506f7765722053514c20496e74657
26e6574204472697665727d3b5053514c5352563d7777772e7673716c2e636f6d3b5053514c504f52543d3435343
53b5053514c44534e3d54656368536f757263653b5053514c454e43523d505343525950542e5043523b5053514c
544f3d303b5549443d757365723b177777772e7673716c2e636f6d5c54656368536f75726365030004757365720 4
7573657200083c204e6f6e65203e"}

Documentation feedback

At eHelp Corporation, we strive to produce the highest quality documentation products and welcome your feedback. If you have comments or suggestions about our online Help or printed guides, you can email us.

Please include the following information with your feedback:

- Product name and version number
- Printed manual or online Help
- Topic title (for online Help)
- Page number (for printed manual)
- Brief description of content (for example, are you reviewing step-by-step instructions that are inaccurate, grammatical errors in a specific paragraph, information that requires clarification or more details, etc.)
- Your suggestion for how to correct/improve documentation

Please send email messages to:

documentation@blue-sky.com

Keep in mind, this email address is only for documentation feedback; you will not receive a reply. If you have a technical question, please contact Technical Support.

Other helpful resources

A good list of resources can make all the difference in terms of support, guidance, and inspiration. As the worldwide leader in Help authoring solutions, eHelp Corporation leads the way by offering a list of helpful resources on our Web site.

You’ll find the latest Help authoring resource information – including books, periodicals, online forums, helpful Internet links, training, and more.

WinHelp 2000

We are unable to detect Internet Explorer or an Internet connection on your system, so we are unable to display eHelp Corporation’s Web site.

This Help file uses WinHelp 2000, which extends the WinHelp viewer to provide full HTML and Web functionality without leaving WinHelp. WinHelp 2000 gives you all the benefits of HTML without the risk of switching to a new Help architecture.

WinHelp 2000 uses the same display engine as Microsoft’s Internet Explorer and HTML Help, so all of the rich content of the World Wide Web can be displayed and integrated into any standard Windows Help system.

If you use a browser other than Internet Explorer, use this alternative link:
▶ eHelp Corporation’s Web site

PC HelpDesk

We are unable to detect ODBC with an Internet connection on your system, so we are unable to display eHelp Corporation's Knowledge Base.

Our Knowledge Base was created using PC HelpDesk – an eHelp Corporation tool included in RoboHELP Office. PC HelpDesk packages your product knowledge base into a customized HelpDesk you design, create, and maintain. Customers simply click on the HelpDesk link from your system's Help file, and they have instant access to valuable information about your products. A well designed HelpDesk can lower your technical support costs and increase customer satisfaction by providing an easy way for your customers to get answers to their most common questions.

Premium Support

Premium Support entitles plan members to priority Technical Support access through a dedicated telephone line and a dedicated email address. Plan members receive direct access to eHelp Corporation’s seasoned Senior Technical Support Engineers.

Premium Support plans are available for current and previous versions of eHelp Corporation products.

{ewl RoboEx32.dll, WinHelp2000, }