

Introduction to Context-Sensitive (What's This?) Help

Context-sensitive Help is a unique type of Help. Users access context-sensitive Help while in a running application. It appears in popup windows that are activated by a mouse click or two. Context-sensitive Help usually displays concise information regarding a specific feature of a program.

Context-sensitive Help is intended to assist users in tasks by informing them about the components of the dialog they are using.

In most Windows 95 and Windows NT applications, there are several ways for a user to invoke context-sensitive Help. The user can do the following:

- Place the cursor over the point to be queried, click the right mouse button, then click What's This?.
- Click the ? button in the title bar of the dialog box; then click an item to be queried.
- Tab to the control to be queried and press F1.
- Click on a Help button if one is available on the dialog.

Tip

With RoboHELP's new What's This? Help Composer, it's easy to add context-sensitive help to any application without custom programming for each dialog.

See Also

{button ,JI('BH32SAMP.HLP>(w95sec)',`How_Context_sensitive_Help_Works')} [How context-sensitive Help works](#)
{button ,JI('BH32SAMP.HLP>(w95sec)',`Sample')} [Context-sensitive Help Sample](#)

This is the Help button. Its What's This? Context ID is 2.

You pressed the Done button. Its Context ID is 28443. This is determined by Windows because its resource ID is IDOK.

The Help button on the left calls WinHelp with a valid Context ID (9). The No Help button on the right calls WinHelp with a Context ID (103) that does not exist in the Help file.

The question mark button on the caption bar changes the cursor to the What's This cursor. You can then click on any control in this dialog to test What's This? context-sensitive Help. Using this cursor causes a WM_HELP message to be sent to the dialog.

Right-clicking on items in this dialog will also let you access What's This? Help, but no WM_HELP message is sent. Note, however, that right-clicking on edit boxes does not pop up the What's This? menu.

This edit box has a correct Context ID. It is 7. The other edit box has a Context ID (101) that does not appear in the [map] section of the HPJ file.

Notice that you can even make static controls have What's This Help. Note that controls must have the Tab Stop window style if you wish to author What's This Help for them.

You clicked the Help button

If you click the No Help button, you'll get a WinHelp error because the Context ID used when the No Help button is clicked (103) is invalid.

How Context-sensitive Help Works

The ? button

If there is a ? button at the top of the dialog, the user can click on this button to change the cursor to a What's This? cursor. Clicking with this cursor asks for Help on a dialog control rather than activating the control, so you can safely click on a button to get Help for that button--it won't cause the button to perform its normal function.

1. User clicks on ?.
2. Cursor changes to What's This? cursor.
3. User clicks on a dialog control.
4. Windows sends a WM_HELP (sometimes known as WM_HELPINFO) message to the control, and then to the dialog itself.
5. The dialog's WM_HELP message handler calls WinHelp with the HELP_WM_HELP command, passing it a table of Control IDs mapped to Context IDs. This must be set up by a programmer unless you use RoboHELP's What's This? Help Composer.
6. WinHelp calls its HELP_SETPOPUP_POS function to set the position for the popup window.
7. WinHelp calls its HELP_CONTEXTPOPUP function to display the topic in a popup window.

Right-clicking on controls

If a dialog has context-sensitive Help, the user can right-click on the dialog's controls to get What's This? Help. Right-clicking does not activate the control--it just brings up a *context menu* with "What's This?" as the top (usually only) menu entry.

1. User right-clicks on a control.
2. Windows sends a WM_CONTEXTMENU message to the control, and then to the dialog itself.
3. The dialog's WM_CONTEXTMENU message handler calls WinHelp with the HELP_CONTEXTMENU command, passing it a table of Control IDs mapped to Context IDs. This must be set up by a programmer unless you use RoboHELP's What's This? Help Composer.
4. WinHelp calls its HELP_SETPOPUP_POS function to set the position for the popup window.
5. WinHelp calls its HELP_CONTEXTPOPUP function to display the topic in a popup window.

Pressing F1

If a dialog has context-sensitive Help, the user can press F1 obtain Help for the dialog or its controls. In Windows 3.x, this is normally the only kind of context-sensitive Help available (unless there is a Help button on the dialog).

1. User tabs to control for which Help is desired (note that in Windows 3.1 context-sensitive Help is usually dialog-based, so it does not matter which control has focus).
2. User presses F1.
3. Windows sends a WM_HELP (sometimes known as WM_HELPINFO) message to the control, and then to the dialog itself.
4. The dialog's WM_HELP message handler calls WinHelp with the HELP_WM_HELP command, passing it a table of Control IDs mapped to Context IDs. This must be set up by a programmer unless you use RoboHELP's What's This? Help Composer.
5. WinHelp calls its HELP_SETPOPUP_POS function to set the position for the popup window.
6. WinHelp calls its HELP_CONTEXTPOPUP function to display the topic in a popup window.


Clicking Help button

The simplest form of context-sensitive Help is provided by a Help button on the dialog. The user simply clicks the button and Help is invoked.

1. User clicks the Help button.
2. Windows sends a WM_COMMAND message to the dialog with the BN_CLICKED notification.
3. The dialog's WM_COMMAND message handler determines that it was the Help button that was clicked and then calls WinHelp to display a Help topic, usually with the HELP_CONTEXT command, passing it the Context ID of the topic to display.
4. WinHelp displays the topic.

Sample

WinHelp BugHunter 4.0 includes a sample program that you can run to explore the world of context-sensitive Help.

1. Start WinHelp BugHunter 4.0 (if it is not already running).
2. Click here  to run the sample application.
3. Make sure that WinHelp BugHunter is enabled.
4. Use right-clicks and the ? button to explore the context-sensitive Help for the sample application.
5. Examine WinHelp BugHunter's output window.

Dummy topic

(New topic text goes here.)

