# Add Project Configuration

Use this dialog box to add a new project configuration based on an existing configuration to an existing project. After you added the configuration, you can use the Project Settings dialog box to alter the settings for this configuration.

**Configuration**
Type the name to identify the new configuration.

**Copy settings from**
Choose the existing configuration from which to copy initial settings.

**Platform**
Choose the platform to be used in the new configuration. This choice may alter some settings from the settings initially copied, and may specify a different tool set.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Add Tool

Use this dialog box to add a tool to the Tools menu.

**Command**
Specify the command that starts the tool you want to add to the Tools menu.

**Browse**
Opens the Browse dialog box, which you can use to select a tool.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Add to Source Control

Use this dialog box to put files under source-code control in the source-code control project associated with your project workspace, either when you initially create the project workspace or when you later add files to it.

**Files**
Lists all the files currently in your project, but not currently under source-code control, with a check against files selected in FileView. If this list does not contain the desired files, click the check boxes in the list to change the selection.

**Keep checked out**
Checks the files out to you after adding the master versions of the files to the source-code control project.

**Comment**
Adds a comment about the files that you are adding, if you desire.

**Advanced**
Accesses additional options. If your source-code control system does not support such options, this button is inactive.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Advanced Breakpoint

Use this subdialog to specify context information (some combination of function, source file, and DLL or EXE) for a location or variable reference used in a breakpoint.

**Location**
This field appears if the Advanced Breakpoints dialog was selected from the Breakpoints dialog Location tab. It contains the location you specified in the Location field of the Breakpoints dialog Location tab. If you edit the location in this field, any changes you make will be reflected in the Location field of the Breakpoints dialog when you click OK.

**Expression**
This field appears if the Advanced Breakpoints dialog was selected from the Breakpoints dialog Data tab. It contains the variable or expression you specified in the Expression field of the Breakpoints dialog Data tab. If you edit the location in this field, any changes you make will be reflected in the Expression field of the Breakpoints dialog when you click OK.

**Function**
Use this field to enter the name of the function where the variable or location can be found.

**Source File**
Use this field to enter the name of the source file where the variable or location can be found. If the source file is not in the current directory, you must include the drive and directory path as well.

**Note    Executable File**
Use this field to enter the name of the executable file or DLL where the variable or location can be found. If the executable file or DLL is not in the current directory, you must also include the drive and directory path.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Advanced Debug Options

This subdialog contains Advanced debug options for 680X0 Macintosh only. If you have selected a Macintosh project configuration, Advanced Debug Options appears when you select the Advanced button on the Debug tab of the Tools Options dialog.

**Stop at Debug Traps**
If this checkbox is cleared, the debugger does not stop at debug() traps.

**Print Code Load Messages**
If this checkbox is selected, the debugger prints messages in the output window whenever a Macintosh code segment loads.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Assertion

An Assertion dialog box appears when a condition enclosed within an assertion statement evaluates to false. In Visual C++, assertion statements can be based on any of the following constructs:

- The ANSI C/C++ assert function
- The C runtime library _ASSERT macro
- The MFC ASSERT macro

For help on debugging assertions, or information about the three types of assertion statements, use the See Also button at the top of this window.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Batch Build

Use this dialog box to build a project for multiple configurations. A configuration is a final binary output file you can create from a project.

**Project configurations**
Select the configuration(s) you want to build.

**Build**
Builds only the elements for each configuration that are out of date.

**Rebuild All**
Builds all elements for each configuration.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Bookmarks

Use this dialog box to add, delete and go to named bookmarks.

**Name**
Type or use the drop-down list to select the name of the bookmark.

**Add**
Add the bookmark listed in the Name field to the list of active bookmarks.

**Close**
Close the Bookmarks dialog box.

**Delete**
Delete the bookmark selected in the Name field from the list of active bookmarks.

**Goto**
Goto the bookmark listed in the Name field.

**File**
Displays the file location for the bookmark listed in the Name field.

**Line**
Displays the line number for the bookmark listed in the Name field.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Breakpoints

Use this dialog box to specify breakpoints. A breakpoint specifies when to halt program execution temporarily. At that point, you can examine values of variables, registers, and so on. A breakpoint can be:

- A location in the source file. Program execution halts immediately before the debugger executes the line marked with a breakpoint.
- A condition. Program execution halts whenever the condition becomes true or changes.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Location Tab

Using the Breakpoints dialog, you can set, clear, disable, enable, and view:

- Location breakpoints
- Data breakpoints
- Message breakpoints
- Conditional breakpoints

Use the Location tab to set, clear, disable, enable, or view location breakpoints. Use the Location tab and the <u>Breakpoint Condition dialog</u> to set conditional breakpoints.

**Break At:**
Use this field to enter the location (line number, memory address, function, or label) where you want to set the breakpoint.

Just to the right of this field is a dropdown list. This list contains the current location (source line number or memory address) for your program. Select this location to enter the current location into the Break At field. Select the Advanced... item in this list to open the <u>Advanced Breakpoints dialog</u>.

**Condition...**
Use this button to display the <u>Breakpoint Condition dialog</u>.

**Breakpoints**
This list shows all current breakpoints for your program. To the left of each breakpoint is a check box. If the checkbox is selected, the breakpoint is currently enabled. If the check box is cleared, the breakpoint is currently disabled. If the check box contains an asterisk (*), the breakpoint is not supported on the current platform. To temporarily disable a breakpoint, clear the corresponding check box. To reenable a breakpoint, select the check box.

**Edit Code**
Click on this button to display the source code containing the breakpoint and move the cursor to the breakpoint location.

**Remove**
Click this button to delete the selected breakpoint.

**Note**   If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Data Tab

Using the Breakpoints dialog, you can set, clear, disable, enable, and view:

- Location breakpoints
- Data breakpoints
- Message breakpoints
- Conditional breakpoints

Use the Data tab to set, clear, disable, enable, or view breakpoints on variables or expressions.

**Enter the expression to be evaluated:**
In this field, enter a variable or expression to determine when to break. If you specify a Boolean expression that evaluates to true or false, such as a==1 or c<d, the debugger will stop when the expression evaluates to true. If you specify a variable or a non-Boolean expression (one that does not evaluate to true or false, such as a+b/c), the debugger will stop when the value of the expression changes.

The text below this field tells whether the debugger will stop if the current expression is true or if the current expression changes. Because only one interpretation makes sense for any given expression, the debugger determines this for you automatically.

Just to the right of this field is a dropdown list. Select the Advanced... item in this list to open the <u>Advanced Breakpoints dialog</u>.

**Enter the number of elements to watch in an array or structure**
If you enter a variable, pointer, or memory address in Expression, you can use this field to indicate the number of elements that you want to monitor.

If you specified a dereferenced pointer, such as `*lptr`, in Expression, use this field to enter the length (in bytes) of the variable that the pointer addresses.

The length must always be a positive number.

**Breakpoints**
This list shows all current breakpoints for your program. To the left of each breakpoint is a check box. If the checkbox is selected, the breakpoint is currently enabled. If the check box is cleared, the breakpoint is currently disabled. If the check box contains an asterisk (*), the breakpoint is not supported on the current platform. To temporarily disable a breakpoint, clear the corresponding check box. To reenable a breakpoint, select the check box.

**Edit Code**
Click on this button to display the source code containing the breakpoint and move the cursor to the breakpoint location.

**Remove**
Click this button to delete the selected breakpoint.

**Note**　If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Messages Tab

Using the Breakpoints dialog, you can set, clear, disable, enable, and view:

- Location breakpoints
- Data breakpoints
- Message breakpoints
- Conditional breakpoints

Use the Messages tab to set, clear, disable, enable, or view a breakpoint on a **WndProc** message.

**Break at WndProc**
Use this textbox to enter the name of the **WndProc** you want to break on, or select a **WndProc** name from the dropdown list.

**Set one breakpoint for each message to watch**
Use this textbox to enter the message that you want to break on, or select a message from the dropdown list.

**Breakpoints**
This list shows all current breakpoints for your program. To the left of each breakpoint is a check box. If the checkbox is selected, the breakpoint is currently enabled. If the check box is cleared, the breakpoint is currently disabled. If the check box contains an asterisk (*), the breakpoint is not supported on the current platform. To temporarily disable a breakpoint, clear the corresponding check box. To reenable a breakpoint, select the check box.

**Edit Code**
Click on this button to display the source code containing the breakpoint and move the cursor to the breakpoint location.

**Remove**
Select this button to delete the selected breakpoints.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Breakpoint Condition

Use this subdialog to specify a condition that you want to attach to a location breakpoint.

**Enter the expression to be evaluated**
In this field, enter a variable or expression that you want the debugger to evaluate as a condition to determine whether to stop at the breakpoint. If you specify a Boolean expression that evaluates to true or false, such as a==1 or c<d, the debugger considers the condition to be met whenever the expression evaluates to true. If you specify a variable or a non-Boolean expression (one that does not evaluate to true or false, such as a+b/c), the debugger considers the condition to be met if the value of the expression changes.

**Enter the number of elements to watch in an array or structure**
If you enter a variable, pointer, or memory address in Expression, you can use this field to indicate the number of elements that you want to monitor. For example, if you enter a[0] in Expression, you can enter 100 in Number of Elements to monitor changes in the first 100 elements of the array.

If you specified a dereferenced pointer, such as `*lptr`, in Expression, use this field to enter the length (in bytes) of the variable that the pointer addresses.

The number of elements must always be a positive integer.

**Enter the number of times to skip before stopping**
Use this field if you do not want the debugger to stop every time the condition is met. The value in this field represents the number of times the debugger should skip this breakpoint. For example, if you enter 4 in this field, the debugger will stop at this breakpoint every fifth time that the condition is met. If you enter 9 in this field, the debugger will stop at this breakpoint every tenth time the condition is met.

You can also use this field, without specifying a condition, to stop at a breakpoint after a specified number of times. You cannot set a Skip Count, however, for a When Expression Changes breakpoint.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Browse

Use this dialog box to select the type of symbol relationship you would like to display.

**Identifier**
Name of the symbol whose relationship you want to display. If you type a pattern using the *
wildcard, symbols matching that pattern are displayed. If you are querying on a file, you must include
a file extension. You can use the * wildcard to search for a pattern in the name or extension.

**Select Query**
Types of symbol relationships you can display. You can choose from:

| Option | Description |
| --- | --- |
| Definitions and References | Indicates where symbols are defined and referenced. |
| File Outline | Displays all user-defined functions, classes, data, macros, and types. |
| Base Classes and Members | Displays all classes from which the selected class inherits attributes. |
| Derived Classes and Members | Displays all classes that inherit attributes from the selected class. |
| Call Graph | Displays relationships among all the functions that the selected function calls. |
| Callers Graph | Displays relationships among all the functions |

that call the
selected
function.

**Case-Sensitive**
Select this checkbox to make the results of your query case sensitive. Clear this checkbox to make
the results of your query case insensitive.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Browse

Use this dialog box to select a tool to add to Developer Studio. Under Windows 95, you get help on the following controls by selecting the control and then pressing F1:

**Look in**
**File name**
**Files of type**


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Browse Files

Use this dialog box to select a file to import into ClassWizard. Under Windows 95, you get help on the following controls by selecting the control and then pressing F1:

**Look in**
**File name**
**Files of type**


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Change Symbol

Use this dialog box to change a symbol's ID or to see which resources are using this symbol.

**Used By**
Resources that use the symbol. You can select one of the resources in the list box.

**View Use**
Displays the property page of the selected symbol. You can use the property page to change the symbol's name or value.

**Note**   If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Check in file(s)

Use this dialog box to check files in to the source-code control project.

**Files**
Lists all the files that you have checked out, with checks against the files selected in FileView. If this list does not contain the desired files, click the check boxes in the list to change the selection.

**Keep checked out**
Keeps the files checked out to you after updating the master versions of the files in the source-code control project.

**Comment**
Adds a comment about the reason for checking in the files, or about the changes made, if you desire.

**Advanced**
Accesses additional options. If your source-code control system does not support such options, this button is inactive.

**Differences**
Displays differences between your local copy and the master copy. If your source-code control system does not support this option, this button may not appear.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Check out file(s)

Use this dialog box to check out files maintained under your source-code control system.

**Files**
Lists all the files that you have checked in, with checks against the applicable files selected in FileView. If this list does not contain the desired files, click the check boxes in the list to change the selection.

**Comment**
Add a comment about the reason for checking out the files, if desired.

**Note**    This option may not be present on your dialog box, because not all source-code control systems support comments when checking out files.

**Advanced**
Accesses additional options. If your source-code control system does not support any advanced options, this button is inactive.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Choose Directory

Use this dialog box to specify the path for the specified file when Microsoft Developer Studio does not find a file. This may happen if a source file has been moved from the project directory before starting a debugging session, or if an executable file is not in the directory specified during Setup.

**Directory name**
Type the path for *filename*.

Under Windows 95, you get help on other controls by selecting the control and then pressing F1.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Configurations

Use this dialog box to add a new configuration to an existing project or remove an existing configuration from a project.

**Projects and Configurations**
Displays the existing projects and configurations in the open project workspace. Click the plus and minus boxes to expand and contract the view of configurations.

**Add**
Displays the Add Configuration dialog box to add a configuration to the project selected in the Projects and Configurations tree. You can select a project by selecting either the project node or a configuration in the project.

**Remove**
Removes the selected configuration from the project. If you remove the last remaining configuration in a project, the project is also removed.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Connection Password

To maintain security, the password to the data source is not cached. Use this dialog box to reestablish permission to connect to a data source.

**Data Source**
Enter the name of the data source to which you are trying to connect.

**Data Base**
Enter the name of the database to which you are trying to connect.

**Password**
Enter the password in this edit box.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Custom Color Selector

Use this dialog box to change the hue, luminance, saturation, or red, green, and blue values of a color selected from the Bitmap, Icon, or Cursor color palette.

Position the crosshair on the color you want to change. Then move the slider up or down to change the luminance or RGB values of the color. Or, specify HLS or RGB values.

**Color**
Displays the solid color closest to the dithered color selected on the palette.

**Hue**
The hue value of the color you are defining. Values range from 0 to 240.

**Sat**
The saturation value of the color you are defining. Values range from 0 to 240.

**Lum**
The luminance value of the color you are defining. Values range from 0 to 240.

**Red**
The red value of the color you are defining. Values range from 0 to 255.

**Green**
The green value of the color you are defining. Values range from 0 to 255.

**Blue**
The blue value of the color you are defining. Values range from 0 to 255.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Custom Image

Use this dialog to enter the width, height and colors for a custom image.

**Width**
Enter the width of the custom image.

**Height**
Enter the height for the custom image.

**Colors**
Choose the colors for the custom image: 2, 16 or 256.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Customize

Use this dialog box to:

- Assign shortcut keys to existing menu commands.
- Add toolbar buttons to existing toolbars.
- Add a command to the Tools menu.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Keyboard Tab

Use the Keyboard tab in the Customize dialog box to assign shortcut keys to existing menu commands.

**Editor**
Select the type of editor to which you want to assign shortcut keys. For example, select Text if you want to assign a shortcut key to text editing commands.

**Categories**
Select the category of commands. For example, select the Edit category for assigning text editor shortcut keys.

**Commands**
Shows commands on the menu selected in the Categories list box.

**Press New Shortcut Key**
To test a new shortcut key combination for the command selected in the Commands list box, press the keys you want to assign. If the key combination is currently assigned to another command, the command is displayed in the Currently Assigned To box. To assign the new shortcut key combination, choose the Assign button.

**Current Keys**
Displays the shortcut key combinations currently assigned to the command selected in the Commands list box.

**Assign**
Assigns the shortcut key combination displayed in the Current Keys list box to the command selected in the Commands list box.

**Remove**
Removes the shortcut key combination that is selected in the Current Keys list box.

**Reset All**
Restores the keystroke combinations for the selected editor to their default setting.

**Description**
Describes the command selected on the Commands list.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Toolbars Tab

Use the Toolbars tab in the Customize dialog box to add toolbar buttons to existing toolbars.

**Categories**
Lists the types of tools you can choose from. You can choose from File, Edit, View, Insert, Layout, Debug, Tools, Window, and Help.

**Buttons**
Shows the tools you can add to toolbars. Resting the pointer on a button gives you a title for the button; selecting it gives you a description in the Description box. To add a tool, drag it from the dialog box to a toolbar.

**Description**
Describes the tool you have selected.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Tools Tab

Use the Tools tab in the Customize dialog box to add a command to the Tools menu.

**Menu Contents**
Lists commands you have added to the Tools menu. Select the command you want to modify.

**Menu Text**
The command name that appears on the Tools menu. Placing an ampersand (&) before a letter makes the letter an accelerator key.

**Command**
The command associated with the command name on the Tools menu.

**Arguments**
Specifies additional arguments for the tool each time you start it. For example, you can use $ (TargetArgs) to specify the command-line arguments that are passed to the application you are developing. You can also specify additional arguments for each particular instance of the tool.

**Initial Directory**
The working directory of the tool.

**Prompt for Arguments**
Displays a dialog box prompting for command-line arguments each time you run the tool.

**Redirect To Output Window**
Specifies that a console application runs without a console window, and that the application's output is displayed in an Output window. Each tool uses a separate output window. You can switch among Output windows by selecting a tab at the base of the output window.

**Close Window On Exiting**
Specifies that the window associated with the tool will be closed when you close the tool.

**Add**
Displays the Add Tool dialog box, which you use to add a tool to Developer Studio.

**Remove**
Deletes the command selected in the Menu Contents box.

**Move Up**
Moves the command selected in the Menu Contents box up the Tools menu one position.

**Move Down**
Moves the command selected in the Menu Contents box down the Tools menu one position.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Data Sources

Use this dialog box to connect to a Microsoft or an Oracle ODBC data source. The listed data sources are those that have been set up using the 32-Bit ODBC Manager.

**Note**      To set up a server so that it is visible from the Data Sources dialog box, use the 32-bit ODBC Manager from the Control Panel.

After you select a data source from the Select Data Source list and click the OK button, a dialog box that allows you to log onto a server will appear.

After you successfully log on, DataView connects to the data source and displays the contained stored procedures, tables, views, and user-defined data types.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Exceptions

Use this dialog box to specify debug actions for system and user-defined exceptions in your program. Exceptions can occur inside or outside exception handlers (sections of code containing structured exception handling statements). The debugger is always called when an exception occurs.

**Number**
The unique number of the exception. System exceptions are defined in WINBASE.H with the prefix of EXCEPTION (for example EXCEPTION_ACCESS_VIOLATION).

**Name**
Optional name to be displayed in the Exception list for the exception.

**Action**
Action to take when the debugger is notified of an exception. It can take two actions: Stop Always or Stop If Not Handled.

**Exceptions**
The list of system exceptions that you want to handle. You can modify this list, deleting system exceptions or adding your own. This information is saved in the *project*.MDP file (where *project* represents your project name) and persists with the project. You can select multiple exceptions.

**Add**
Adds the exception, as specified in the Number and Name text boxes, to the Exceptions list, along with an optional action. The default action, if none is specified, is Stop If Not Handled.

**Remove**
Deletes the selected exception(s) from the Exceptions list. The debugger still handles deleted exceptions, with the action Stop If Not Handled.

**Change**
Accepts changes made to the highlighted exception(s) in the Exceptions list. A change might include a different action, for instance.

**Reset**
Restores all default system exceptions to the exceptions list without disturbing any of the user-defined exceptions that have been added.

Note    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Executable For Debug Session

Use this dialog box to specify the executable file you want to debug.

**Executable For Debug Session**
The path and name of the executable file that you want to debug.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Export Resource

Use this dialog box to save a bitmap, icon, cursor or custom resource as a separate file. Under Windows 95, you get help on the following controls by selecting the control and then pressing F1:

**Note**　If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Find

Use this dialog box to:

- Find a symbol or a sequence of symbols in a file.
- Find all symbols or sequences of symbols that match a given pattern.
- Set bookmarks (tags) on lines containing a symbol or sequence of symbols.

**Find What**
Specifies the search text or the regular expression to match. You can use the menu button to the right of the combo box to display a list of regular search expressions. When you select an expression in this list, the expression is substituted as search text in the Find What text box. If you use regular expressions, be sure the Regular Expression option is checked.

You can also use the drop-down list to select from a list of up to 16 previous search strings.

**Match Whole Word Only**
Matches text strings only if they are preceded and followed by a space, tab, or punctuation character, or the start or end of a line. Otherwise, the command matches any string, whether it is a fragment of a larger string or not.

**Match Case**
Finds only text strings that match the case of the characters in the Find What string exactly. Otherwise, the command finds strings with either uppercase or lowercase characters that match the characters in the Find What string.

**Regular Expression**
Check this option if you use regular expressions in the Find What text box.

**Direction**
The options have the following meanings:

Up    Search from the current cursor position toward the beginning of the file.

Down    Search from the current cursor position toward the end of the file.

**Find Next**
Repeats the most recent find operation.

**Mark All**
Places a bookmark on all lines containing the string or regular expression found. The lines are highlighted in your source file.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Find in Files

Use this dialog box to search for a sequence of characters in one or more files. You specify the files to search by indicating the type of file and the folders to search in. You can use regular expressions to match character patterns in the selected files. The search results are displayed in the Output window. When the search is complete, you can open a file containing a match by double-clicking the entry in the Output window.

**Find What**
Specifies the search text or the regular expression to match. You can use the menu button to the right of the drop-down list to display a list of regular search expressions. When you select an expression in this list, the expression is substituted as search text in the Find What text box. If you use regular expressions, be sure the Regular Expression option is checked.

You can also use the drop-down list to select from a list of up to 16 previous search strings.

**In Files of Type**
Enter the type of file you want to search or select from the drop-down list. The filename extension determines the file type.

**In Folder**
Select the primary folder that you want to search. You can use the browse button ( . . . ) to display the Choose Directory dialog box to change drives and directories.

**Look in Subfolders**
Select this option to extend the search to subfolders.

**Regular Expression**
Check this option if you use regular expressions in the Find What text box.

**Match Case**
Finds only text strings that match the case of the characters in the Find What string exactly. Otherwise, the command finds strings with either uppercase or lowercase characters that match the characters in the Find What string.

**Match Whole Word Only**
Matches text strings only if they are preceded and followed by a space, tab, or punctuation character, or the start or end of a line. Otherwise, the command matches any string, whether it is a fragment of a larger string or not.

**Advanced**
Choose the Advanced button to select additional folders to search. This button displays an extended dialog with the option to look in project source and include file folders.

**Look in Project Folders for Source Files**
Select this option to extend the search to project source file folders.

**Look in Project Folders for Include Files**
Select this option to extend the search to project include file folders.

**Note**    These project source and include file folders are the same as the project's directory paths. For more information on how to view and change these directory paths, select Setting Directories from the See Also dropdown list.

**Look in Additional Folders**
To add a folder to this list, double-click the empty selection. Then type the path and filename or use the browse button ( . . . ) to display the Choose Directory dialog box to change drives and directories.

To remove a folder from this list, select and delete the text entry.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Browse for Local Module

Use this dialog box to find a local (PC host) copy of a program module (DLL or EXE) that you are debugging on a remote machine. To specify a local DLL pathname before debugging begins, choose the ... button on the Debug tab (Additional DLLs category) of the Project Settings dialog. If you have not specified a module name before debugging begins, the Find Local Module dialog appears. You can then choose the Browse button to access this dialog.

Under Windows 95, you get help on the following controls by selecting the control and then pressing F1:

**Look in**
**File name**
**Files of type**


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Copy

Use this dialog box to specify the destination directory of a sample copy operation.

**Copy files**
This edit box contains the list of files that will be copied when you click OK. Initially, this will be either the currently selected file on the Sample Application dialog box when you clicked the Copy button, or it will be the list of all the files in the sample project if you clicked the Copy All button. You can also edit this list manually.

**To directory**
This combination edit box and directory list tree is used to select the destination directory for the copy operation. It will initially suggest a directory based on where samples have been copied in the past. You can change the destination by typing a new path into the edit box or selecting a new path in the directory tree.

**Drive:**
This drop-down list displays all the drives currently available to your machine. This includes floppy drives, hard drives, CD-ROM drives, and network drives. Use this list to select the destination drive for the copy operation.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Default Project Configuration

Use this dialog box to choose the default project configuration to display and build. The default project configuration is displayed in the workspace window. Build commands apply to this project configuration. If you build a project containing subprojects, the same configuration is built in all the subprojects, if they are out of date.

**Project Configurations**
Select a project configuration from the list of existing projects and their configurations.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Find Local Module

Use this dialog box to find a local (PC host) copy of a program module (DLL or EXE) that you are debugging on a remote machine. This dialog box appears during remote debugging if a pathname for the local module has not been specified on the Debug tab of the Project Settings dialog.

**Local file name for remote file**
Use this field to specify a local module name (or use the Browse button).

**Try to locate other DLLs**
Select this check box if you want the Find Local Module to ask you to locate additional DLLs after the current DLL or EXE.

**Browse**
Use this button to find local modules using the Browse for Local Module dialog to find an EXE or DLL.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Find Source

Use this dialog box to specify the path for the specified file when Microsoft Developer Studio does not find a file. This may happen if a source file has been moved from the project directory before starting a debugging session, or if an executable file is not in the directory specified during Setup.

**Please enter the path for FILENAME**
Type the path for *filename*.

**Drives**
Click the arrow to display a list of all attached drives. Click a drive from that list to display directories and filenames from the selected drive.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Get Latest Version

Use this dialog box to update your local copies of files with the latest versions from the master source-code control project files.

**Files**
Lists the current source file or all the files that you have selected in the project window. If this list does not contain the desired files, click the check boxes in the list to change the selection.

**Advanced**
Accesses additional options for getting files, such as getting a specific version of a file, for instance. If your source-code control system does not support such options, this button is inactive.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Go To

Use this dialog box to move around in source files.

**Go to What**
Select from this list of go to items: Address, Bookmark, Definition, Error/Tag, InfoViewer Annotations, InfoViewer Bookmarks, Line, Offset, Reference.

For example, Address jumps to any valid debugger expression in the Memory or Disassembly window. In the binary data editor, Offset jumps to the specified decimal or hexadecimal offset in the file.

**Additional Selection Criteria**
Depending on the Go to What selection, the additional selection criteria changes to either an edit control or a drop-down list. Help text is displayed defining the additional selection criteria. For example, if you select Bookmark, the additional selection criteria requests you to enter the bookmark name.

**Go To**
Use this navigation button to go directly to the selection.

**Previous**
Use this navigation button to go move backward from the current selection.

**Next**
Use this navigation button to move forward from the current selection.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Guide Settings

Use this dialog box to:

- Turn Snap to Grid on or off for all dialog box editor windows.
- Turn Rulers and Guides on or off for all dialog box editor windows.
- Work without Guides or Grid (none).
- Change the size of the layout grid.

**Rulers & Guides**
Adds rulers to the layout tools; guides can be placed in the rulers. The default guides are the two margins, which can be moved by drag and drop. Click in the rulers to place a guide.

Controls "snap" to guides when the controls are moved over or next to them. Controls are also moved with a guide, once attached to a guide. When a control is attached to a guide on each side, and a guide is moved, the control is resized.

**Snap to Grid**
Automatically aligns controls on the layout grid.

**None**
No layout tools selected.

**Width**
Width of the layout grid, in dialog box units (DLUs). A horizontal DLU is the average width of the dialog box font divided by four.

**Height**
Height of the layout grid, in DLUs. A vertical DLU is the average height of the dialog box font divided by eight.

**Note**   If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Import from OLE TypeLib

Use this dialog box to create a class definition from an existing OLE type library. Under Windows 95, you get help on the following controls by selecting the control and then pressing F1:

**Look in**
**File name**
**Files of type**


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Import Resource

Use this dialog box to add a bitmap, icon, cursor or custom resource to the current resource file.

**Open As**
Click either Auto or Custom to import the resource as a certain type. Auto, the default setting, imports the resource by matching the type to any existing types. Custom imports the resource as the type you specify.

Under Windows 95, you get help on the following controls by selecting the control and then pressing F1:

**Look in**
**File name**
**Files of type**

**Note**   If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Insert File

Inserts a file into the currently open source editor window at the insertion point location.   Under Windows 95, you get help on the following controls by selecting the control and then pressing F1:

**Look in**
**File name**
**Files of type**


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Insert Files into Project

Inserts a file or set of files into a project. The target project must be in the current project workspace.

**Add to Project**
Click the arrow to display a list of projects to which you can add files. Only projects in your workspace appear in this list. (Most workspaces have only one project.)

Under Windows 95, you get help on the following controls by selecting the control and then pressing F1:

**Look in**
**File name**
**Files of type**


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Insert OLE Control

This dialog displays a list of OLE controls you can insert into a dialog in the dialog editor. Inserting a control from this dialog does not generate a wrapper class. If you need a wrapper class you may use ClassWizard or Component Gallery to create one.

If an OLE control does not appear in this dialog, try installing the control according to the vendor's instructions, or add the control to Component Gallery.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Insert Project

Use this dialog box to insert a new project into the current project workspace. The project has configurations for Debug and Release for every platform selected. It creates a subdirectory for the project files in the project workspace directory.

**Name**
Enter the name for the new project. Microsoft Developer Studio also creates a subdirectory for the project using this same name. This name is appended to the existing project workspace directory to form the fully qualified path for the new project directory, as shown in the Location text box.

**Type**
Select the type for the project. This choice sets some default options for the build.

**Top-level project**
Choose this option if the newly inserted project is not a subproject of another project—that is, if no other project depends on this project.

**Subproject of**
Choose this option and select a project from the list if you want the newly inserted project to be a subproject of the selected project—that is, if the selected project depends on the newly inserted project.

**Platform**
Select the platforms for which you want configurations. Each platform choice specifies a set of build tools and some default options for the build.

**Location**
Displays the path for the newly created project's subdirectory. You cannot edit this field.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

## Insert Resource

Use this dialog box to choose the type of resource you want to create. Select the resource you want in the Resource Type box and choose OK.

**Resource Type**
The kind of resource you want to create. The following resources are displayed: accelerators, bitmaps, cursors, dialogs, icons, menus, string tables, toolbars, version information and any custom resources.

**Custom**
Click this button to add a new custom resource to the current resource file.

**Import**
Click this button to add a bitmap, icon, cursor or custom resource to the current resource file.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

## Insert Resource

Use this dialog box to choose the type of resource you want to create. Select the resource you want in the Resource Type box and choose OK.

**Resource Type**
The kind of resource you want to create. The following resources are displayed: accelerators, bitmaps, cursors, dialogs, icons, menus, string tables, toolbars, version information and any custom resources.

**Custom**
Click this button to add a new custom resource to the current resource file.

**Import**
Click this button to add a bitmap, icon, cursor or custom resource to the current resource file.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Insert Resource Copy

Use this dialog box to create a copy of a resource in a resource collection.

**Language**
Lists all the available languages in which you can create resources.

**Condition**
Enter a defined symbol if you want the resource to be built only when that symbol is defined for your configuration.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Create Trigger

Use this dialog box to provide a name for your trigger and specify the conditions under which it is activated. You can specify that a trigger be fired when data is inserted into, deleted from, and/or updated in the currently selected table.

If you type a name into the edit box, specify activiation conditions, and click OK, the trigger's name will be visible in DataView under the Table folder and an appropriately-named skeletal procedure will be opened in the editor.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Load Palette Colors

Use this dialog box to load a previously saved custom color palette into the Developer Studio image editor. Under Windows 95, you get help on the following controls by selecting the control and then pressing F1:

**Look in**
**File name**
**Files of type**


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Macintosh Network (AppleTalk) Settings

Use this dialog box to set the communication settings to use while debugging a Macintosh application across an AppleTalk network.

**Remote Machine Name**
Specify the name of the target Macintosh computer.

**Appletalk Zone**
Specify the AppleTalk zone containing the target Macintosh computer.

**Debug Monitor Password**
Specify the password for debugging. This password must match the password set in the debug monitor running on the target Macintosh computer.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Macintosh Network (TCP/IP) Settings

Use this dialog box to set up a connection for remote debugging of a Macintosh application across a network using TCP/IP protocol.

**Remote Machine Name**
Type the identifier for the machine running the program you want to debug. The identifier can be the machine name or the machine's TCP/IP address. Under some circumstances (such as when debugging a program running on a Macintosh), TCP/IP name resolution may not be possible. In those cicumstances, you must use the TCP/IP address instead of the machine name.

A TCP/IP address is a series of four non-negative integers, separated by periods (for example, 125.12.0.1). Each integer can have up to three digits. Leading zeroes are not required. If you do not know the TCP/IP address for the machine you want to connect to, contact your network administrator.

**Debug Monitor Password**
Type the debug password for the target machine running the program you want to debug. This password must match the password set in the Visual C++ Debug Monitor on the target machine.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Macintosh Serial Settings

Use this dialog box to set the communication settings to use while debugging a Motorola 68000-series Macintosh application across a serial connection. You must set the same settings on the Macintosh and reboot the Macintosh before debugging.

**Connector**
Specify the communications port your modem uses.

**Baud Rate**
Specify the transfer rate for the port. Some modems can transmit at more than one baud rate, so check your hardware manual and select one that both systems can handle.

**Data Bits**
Specify the number of data bits for the port. Check your hardware manual and select a number both systems can handle.

**Parity**
Specify the type of parity for the port. Check your hardware manual and select a parity both systems can handle.

**Stop Bits**
Specify the number of stop bits for the port. Check your hardware manual and select a number both systems can handle.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Microsoft Developer Studio

This dialog box indicates that the browse information is out of date or does not exist. Choose Yes to build your project, and generate the .SBR and .BSC files.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Name Stored Procedure

Use this dialog box to provide a name for your stored procedure.

If you type a name into the edit box and click OK, the name will be visible in DataView under the Stored Procedures folder and an appropriately-named skeletal procedure will be opened in the editor.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# New

Use this dialog box to choose what you want to create.

**Text File**
Creates a text file.

**Project Workspace**
Opens the New Project Workspace dialog box, which you use to create a new project workspace and the initial project in the workspace.

**Resource Script**
Creates an empty script file for resources. You add resources to this file with the Insert Resource command.

**Resource Template**
Creates a new file for resource templates. You add resources to this file with the Insert Resource command, modify them, and then save the file in the Templates directory under your Developer Studio installation directory. Subsequently, the resources that you have created appear in the list of resources in the Insert Resource dialog.

**Binary File**
Opens an empty raw-data editor. You can use this editor to create and manipulate arbitrary files.

**Bitmap File**
Opens the bitmap resource editor.

**Icon File**
Opens the icon resource editor.

**Cursor File**
Opens the cursor resource editor.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# New Custom Resource

You can use this dialog box to input a custom resource type.

**Resource Type**
Enter a custom resource type. The resource type should be all capital letters.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# New Cursor Image

Use this dialog box to add a new cursor image to the list of devices available in the cursor editor.

**Target Device**
Select the device on which you intend users to display the cursor.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# New Icon Image

Use this dialog box to add a new cursor image to the list of devices available in the cursor editor.

**Target Device**
Select the device on which you intend users to display the cursor.

**Custom**
Click this button to create your own custom icon image.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# New Project Workspace

Use this dialog box to create a new project workspace and the initial project in the workspace. The project workspace consists of a directory and files in that directory describing individual projects that you add to the workspace. The files for the initial project also reside in the project workspace directory.

A project consists of a single set of files and a set of configurations. Each configuration specifies the set of tools required to build a version of the output file using the set of files, and the settings for those tools. The project workspace that you create consists of information about the initial project, such as the files it contains and the configurations it includes. Once you begin using the workspace, it records information about your preferences for the display of information while working with this project. You can later add other projects or configurations to the workspace.

With this dialog box, you specify a type for the initial project, one or more platforms for which to create project configurations, and a subdirectory for the workspace and its associated files. Initially, for the given project type, you create project configurations with Debug and Release versions for each platform selected.

**Name**
Enter the name for the new project workspace. Microsoft Developer Studio also creates a subdirectory for the workspace using this same name. This subdirectory is appended to the directory that appears in the Location box, and the fully qualified path appears below the "Project will be placed in" label.

**Type**
Select the type for the project that you will initially create. Click the See Also button and select the reference to the types installed for your product or products.

**Platform**
Select the platforms for which you want configurations. Each platform choice specifies a set of build tools, and some default options for the build.

**Location**
Enter the directory in which you want to create the new project workspace subdirectory, if necessary. The initial default choice is the Projects subdirectory in your Developer Studio installation directory. If you change the default location, Developer Studio retains that as the default location for subsequent project workspaces that you create. Developer Studio creates subdirectories in the project workspace directory for all projects that you subsequently add.

**Browse**
Displays a dialog box for selecting a new drive and/or directory for the location.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# New Project Workspace

Use this dialog box to create a new project workspace and the initial project in the workspace. The project workspace consists of a directory and files in that directory describing individual projects that you add to the workspace. The files for the initial project also reside in the project workspace directory.

A project consists of a single set of files and a set of configurations. Each configuration specifies the set of tools required to build a version of the output file using the set of files, and the settings for those tools. The project workspace that you create consists of information about the initial project, such as the files it contains and the configurations it includes. Once you begin using the workspace, it records information about your preferences for the display of information while working with this project. You can later add other projects or configurations to the workspace.

With this dialog box, you specify a type for the initial project, one or more platforms for which to create project configurations, and a subdirectory for the workspace and its associated files. Initially, for the given project type, you create project configurations with Debug and Release versions for each platform selected.

**Name**
Enter the name for the new project workspace. Microsoft Developer Studio also creates a subdirectory for the workspace using this same name. This subdirectory is appended to the directory that appears in the Location box, and the fully qualified path appears below the "Project will be placed in" label.

**Type**
Select the type for the project that you will initially create. Click the See Also button and select the reference to the types installed for your product or products.

**Location**
Enter the directory in which you want to create the new project workspace subdirectory, if necessary. The initial default choice is the Projects subdirectory in your Developer Studio installation directory. If you change the default location, Developer Studio retains that as the default location for subsequent project workspaces that you create. Developer Studio creates subdirectories in the project workspace directory for all projects that you subsequently add.

**Browse**
Displays a dialog box for selecting a new drive and/or directory for the location.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# New or Change Symbol

Use this dialog box to define the name and, if necessary, a value for a new symbolic resource identifier. This same dialog box appears when you ask to change the name or value of an unassigned symbol.

**Name**
The name of the symbol you want to create or change. All symbol names must be unique within the scope of the application. This prevents conflicting symbol definitions in the header files. Legal characters for a symbol name include A–Z, a–z, 0–9, and the underscore (_). Symbol names cannot begin with a number and are limited to 247 characters. Symbol names are case insensitive, but the case of the first symbol definition is preserved.

**Value**
The value of the symbol. A symbol value can be any integer expressed in the normal manner for **#define** preprocessor directives. Symbol values for resources can be decimal values in the range from 0 to 32767. Symbol values for parts of objects (such as dialog box controls or individual strings in the string table) can be in the range from 0 to 65534 or from –32768 to 32767.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# New Toolbar

Use this dialog box to create a new toolbar and name it.

**Toolbar Name**
Type the name of the toolbar you want to create. Developer Studio creates an empty toolbar. To place buttons on the toolbar, drag them from the Toolbars tab to the new toolbar.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# New Toolbar Resource

Use this dialog to convert a bitmap resource to a toolbar resource. The images are cropped to the width and height specified, and the colors are adjusted to use standard toolbar colors.

**Button Width**
Enter the width for the toolbar buttons.

**Button Height**
Enter the height for the toolbar buttons.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

## Next Key Typed

The next key that you type becomes the accelerator key. For example, if you type the ESC key, instead of dismissing the dialog box, the ESC key is assigned as the accelerator.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Open

Use this dialog box to open any file.

**Open As**
Specifies the editor that Developer Studio uses to open a file. If you specify "Auto," Developer Studio determines the editor based on filename extension.

Under Windows 95, you get help on the following controls by selecting the control and then pressing F1:

**Look in**
**File name**
**Files of type**
**Open as read-only**


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Open

Use this dialog box to select a DAO data source to be attached to a record set. Under Windows 95, you get help on the following controls by selecting the control and then pressing F1:

**Look in**
**File name**
**Files of type**


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Open Cursor Image

Use this dialog box to change the device definition currently being used by the cursor image editor.

**Current Images**
Lists available cursor images. Select the image you want to open.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Open Icon Image

Use this dialog box to change the device definition currently being used by the icon image editor.

**Current Images**
Lists available icon images. Select the image you want to open.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Open Project Workspace

Use this dialog box to open a project workspace.

**Source Control**
Opens your source-code control dialog box for accessing source-code control projects. If you have not installed a source-code control system that conforms to the Microsoft Source Code Control API Specification or if your source control system does not require this button, this button is not available.

Under Windows 95, you get help on the following controls by selecting the control and then pressing F1:

**Look in**
**File name**
**Files of type**


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Open Sample CD

Use the Open Sample CD dialog box to tell InfoViewer where to find the source files for the sample projects so it can copy them to your local installation.

**Location**
Use the Location edit box to specify the location of the samples. This should be the directory that contains the SAMPLES directory.   For example, if your CD-ROM drive is D:, the path in the Location edit box should be D:\MSDEV\ because the SAMPLES directory resides in the MSDEV directory on the Visual C++ CD.

This could be a different location if, for example, you installed Visual C++ from a network drive.

**Browse**
Use the Browse button to display an Open dialog box, which allows you to navigate to the location of the source files rather than typing it in the Location edit box.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Options

Use this dialog box to:

- Configure the Micorsoft Developer Studio debugger.
- View or change the directory paths of executable, include, or library files.
- Specify the configuration of the Developer Studio editor.
- Specify fonts and colors for various Developer Studio window and text elements.
- Specify which help files to search when using F1 help.
- Specify the order in which the files are searched.
- Specify how your workspace is configured when you start Developer Studio.
- Specify information and options used by your source-code control system.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Debug Tab

Use the Debug tab in the Options dialog box to configure the Developer Studio debugger. Most options on this tab are grouped according to the window they are associated with: Disassembly, Registers, Call Stack, or Memory. A fifth group of options is not associated with any particular window; this group has no heading.

**Hexadecimal**
Shows values in hexadecimal format.

**Disassembly Window: Source Annotation**
Annotates disassembled code with source code.

**Disassembly Window: Code Bytes**
Displays code bytes when disassembling code.

**Disassembly Window: Symbols**
Displays symbols when disassembling code.

**Call Stack Window: Parameter Values**
Displays parameter values on the call stack.

**Call Stack Window: Parameter Types**
Displays parameter types on the call stack.

**Variables Window: Return Value**
Determines whether the Variables window Auto tab displays function return values.

**Registers Window: Floating Point**
Displays current values of the floating-point registers and flags.

**Memory Window: Address**
Type the starting address for the memory block you want to display. The address can be a numeric value, a register, or a variable address.

**Memory Window: Format**
Specifies the formats in which memory contents are displayed. By default, memory is displayed in byte (8-bit hex) format, with the equivalent ASCII characters shown in the Memory window's right column.

**Memory Window: Re-evaluate Expression**
Causes the Memory window to re-evaluate the memory location, from the expression in the Address text box, every time the Memory window is opened.

**Memory Window: Show Data Bytes**
Displays data in byte (8-bit hex) format along with the display format you choose using the Format option.

**Memory Window: Fixed Width**
Specifies the number of formatted values on each line of the Memory window. By default, this field is Var (variable), which means that the number of formatted values depends on the dimensions of the window. To hard-code the number of value columns, specify an integer in this field.

**Just-In-Time Debugging**
Enables the debugger to be invoked when a fault occurs in an executable launched outside the Microsoft Developer Studio environment.

**OLE RPC Debugging**
Enables the debugger to step into OLE Remote Procedure Calls.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

## Compatibility Tab

Use the Compatibility Tab to establish your choice of text editor behavior.

**Recommended Options For**
You can choose the standard Developer Studio text editor, the Visual C++ 2.0 editor, or emulation of either the BRIEF® or Epsilon™ editors. When you choose an editor from this list, the text editor emulates the key bindings, text selection characteristics, and most editing commands of the selected editor. The default editor is Developer Studio.

**Options**
Displays the compatibility options and their settings for the selected editor. You can modify the editor's behavior by selecting the following options:

Disable Backspace At Start Of Line    Prevents joining of lines by using the BACKSPACE key.

Enable Copy Without Selection    Enables the copy command to copy the entire current line (even if there is no selection).

Enable Line-mode Pastes    Pastes previously copied lines into the text above the current line (and not at the cursor position).

Enable Virtual Space    Allows the cursor into a locations that do not currently contain text.

Include Caret Positioning In Undo Buffer    Retraces previous caret positions using the Undo command.

Indent Separate Paragraphs    Paragraphs normally are defined as lines of text between white space. When you are using paragraph commands such as ParaDown and ParaUp, this option treats any line beginning with a tab character as a new paragraph.

Use BRIEF's Regular Expression Syntax    Applies BRIEF regular expression syntax to your selected text editor. If you have selected BRIEF emulation, then BRIEF regular expression syntax is already available.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Directories Tab

Use the Directories tab in the Options dialog box to view or change the directory paths of executable, include, library, or source files.

**Platform**
Specifies the platform on which the directory paths exist.

**Show Directories For**
Determines the type of directories displayed in the Directories list box. You can choose to display directories that contain:

- Executable files used by build utilities such as NMAKE, CL, LINK, and BSCMAKE
- Default include files used by build utilities
- Default library files used by build utilities
- MFC and MSVC CRT source files used by the debugger

**Directories**
Displays a list of directories that contain the type of files chosen in Show Directories For. The directories searched first are shown at the top of the list. To change the order in which directories are searched, drag one of the directories up or down the list and drop it in a new location. To delete a directory from the list, select the directory and press the delete key. To add a directory to the list, select the blank entry (marked by an empty box) at the bottom of the list and type in the directory.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Editor Tab

Use the Editor tab in the Options dialog box to specify the configuration of the editor.

**Vertical Scroll Bar**
Specifies that editor windows include vertical scroll bars.

**Automatic Window Recycling**
Allows reuse of a single document window when the process of debugging, browsing, or traversing the list of build errors would otherwise create multiple windows.

**Drag-and-Drop Text Editing**
Enables drag-and-drop text editing so you can move or copy selected text with the mouse.

**Horizontal Scroll Bar**
Specifies that editor windows include horizontal scroll bars.

**Selection Margin**
Specifies that Microsoft Developer Studio display a margin to the left of each line of text. You can use this margin to select text and display information about source lines, including breakpoints, instruction points, and tag pointers.

**Save Before Running Tools**
Specifies that Microsoft Developer Studio save files before you build a project or run a build utility such as NMAKE.

**Prompt Before Saving Files**
Specifies that Microsoft Developer Studio confirm that you want to save files.

**Automatic Reload of Externally Modified Files**
Specifies that Microsoft Developer Studio will automatically reload externally modified files that have been loaded (but not yet changed) by the editor.

**Note**     If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Format Tab

Use the Format tab in the Options dialog box to specify fonts for various text elements of Microsoft Developer Studio. You can also use this tab to specify colors for various window and text elements of Microsoft Developer Studio.

**Category**
Select the window for which you want to set formatting options. If you select All Windows, the formatting options are applied to all windows in Developer Studio. Categories in bold type specify all windows in that category. Categories in normal type specify windows included in the categories in bold type. For instance, you can set options for all the windows related to the debugger, and then set different options for individual types of debugger windows, such as the Calls Window.

**Reset All**
Resets the formatting options for a selected Category to the default setting.

**Font**
Select a font from the list of available fonts.

**Size**
Type the font size you want or select from the list of available font sizes.

**Colors**
Select the window element for which you want to specify a foreground or background color. The list of elements displayed changes depending on the window(s) selected in the Category list.

The Text element in the Source Windows category sets the default for all the other Text elements in all other windows. The selection of Automatic in the Foreground or Background lists for the Source Windows category uses the system-wide choices of colors in the Control Panel.

**Foreground**
Select a color for text in the selected element in the Code Colors list. Click the arrow to display the standard 16 colors or Automatic. Automatic specifies the use of the color for the same element in the Source Window category.

**Background**
Select a background color for the selected element in the Code Colors list. Click the arrow to display the standard 16 colors or Automatic. Automatic specifies the use of the color for the same element in the Source Window category.

**Sample**
Displays an example of the selected format options.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# InfoViewer Tab

Use the InfoViewer Tab in the Options dialog to configure the InfoViewer online help system.

**Topic Window:**
**Show toolbar**
Toggles display of the toolbar at the top of the Topic window.

**Show title**
Toggles display of the title of the current topic, in a non-scrolling region at the top of the Topic window.

**Underline jumps**

Toggles underlining of jumps to other topics.

**Track topic in InfoView**

When checked, the Table of Contents display in the InfoView pane is automatically synchronized to show the position of the current topic.

**Load information title at startup**
When checked, Microsoft Developer Studio opens the most recently used information title at startup. You can turn this off if you're using Developer Studio without an information title and you don't want to be prompted for the title at startup.

**Zoom:**
**Topic title**
Determines the degree of enlargement used when displaying the topic title.

**Topic text**
Determines the degree of enlargement used when displaying topic text.

**Query:**
**Display Highlights**
When displaying a topic as a result of a full-text search, highlights each occurrence of the word or phrase that was searched for.

**Jump to First Highlight**
When displaying a topic as a result of a full-text search, automatically scrolls the topic to display the first occurrence of the word or phrase that was searched for.

**NEAR means within**
Specifies how close two words must be to satisfy a query, if you specified the NEAR operator when performing a full-text search.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Source Control Tab

Use this tab to set the options used with your source-code control system.

**Get files when opening the workspace**
Displays a dialog box to copy the latest versions of the master copies of files under source-code control to your workspace when you open the project workspace.

**Check out source file(s) when edited**
Displays a dialog box to check out a file from source-code control if you attempt to edit the file while it is checked in.

**Check in files when closing the workspace**
Displays a dialog box to check files in if they are checked out when you close the workspace.

**Add files during creation**
Displays a dialog box to add files to source-code control when you add them to a project.

**Advanced**
Accesses additional options. If your source-code control system does not support such options, this button is inactive.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Tabs Tab

Use the Tabs tab in the Options dialog box to specify the configuration of tab and indent settings for the editor. You also use the Tabs tab to control auto and smart indenting options.

**File Type**
Select the file type for subsequent choices in the dialog box. Your choice determines the default tab and indent settings. Default specifies all files not otherwise identified.

**Tab Size**
Defines the number of space characters that equal one tab character. The default is four spaces.

**Indent Size**
Defines the column width in spaces between indent locations. The default is four spaces. Pressing the TAB key once moves the cursor to the next indent location.

Tab Size and Indent Size work together in this way: If you have an Indent Size of seven characters, a Tab Size of three characters, and the insertion point in a text file is at an indent location, pressing the TAB key once inserts two tab characters (each three spaces wide) plus one space character (one space wide) to move the seven character positions to the next indent location.

**Insert Spaces**
If selected when you save a file, Microsoft Developer Studio converts tabs to the number of spaces specified in the Tab Size box.

**Keep Tabs**
If selected when you save a file, Microsoft Developer Studio saves tabs as tab characters.

**Auto Indent**
Specifies how Microsoft Developer Studio will indent source code in the specifed file type.

| | |
|---|---|
| None | Do not indent the next line |
| Default | Indent to the most recently used |
| Smart | Indent based on the specified program language element |

**Smart Indent Options**
Specifies how to determine indents, if the Smart option is chosen in the Auto Indent group.

Indent Open Brace    Indent open brace.

Indent Closing Brace     Indent closing brace.

Previous Lines Used For Context     The number of lines used as context for determining the Smart indent option.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Workspace Tab

Use the Workspace tab in the Options dialog box to specify how your workspace is configured when you start Developer Studio.

**Docking Views**
Select the check box associated with a window to enable the window to be docked along the edges of Developer Studio.

**Display Status Bar**
Select this option to display the status bar.

**Display Clock On Status Bar**
Select this check box to display the status bar.

**Reload Documents When Opening Project**
If you select this option, each time a project is opened the project facility restores document windows to the positions they last occupied in the project's workspace.

**Reload Last Project At Startup**
Select this option to automatically load the project you last worked on.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

## SQL Tab

Use the SQL tab in the Options dialog box to specify how the SQL editor and the SQL Debugger are configured when you start Developer Studio.

**Limit SQL output**
Select the check box to limit the number of rows displayed in the Output window when you use the Dump Table command. This command is on the pop-up menu associated with individual tables in DataView. You specify the limit in the edit box that is associated with this check box.

**Automatically commit to data source**
Select the check box to specify that the stored procedures and triggers that you have touched be automatically commited to the data source under the following conditions:

- When you choose Save or Save As from the File menu.
- When you choose the Work Offline command from either the Build menu or the pop-up menu that is associated with a data source in DataView.

If this check box is not selected, you must use the Commit to Server command from either the Build menu or the pop-up menu that is associated with a data source in DataView.

**SQL Server debugging**
Select the check box to enable SQL Server debugging.

**Verbose SQL output**
Select the check box to specify that all output from the SQL Server driver is displayed to the Debug pane of the Output window during debugging.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Page Setup

Use this dialog box to:

- Set the margins on printed pages.
- Specify the text for running headers and running footers on each page.

**Header**
The text printed at the top of every printed page. The header is always centered and prints 0.25" from the top of the page. You can use the dropdown list to insert the appropriate macro for an item in the list.

**Footer**
The text printed at the bottom of every printed page. The footer is always centered and prints 0.5" from the bottom of the page. You can use the dropdown list to insert the appropriate macro for an item in the list.

**Margins**
The width, in inches, of each margin on every printed page.

**Tips and Troubleshooting**    You can use the following special character sequences in headers and footers to insert the corresponding information automatically:

| To Print | Use |
| --- | --- |
| Filename | &f |
| Page number of current page | &p |
| Current system time | &t |
| Current system date | &d |
| Left aligned | &l |
| Centered | &c |
| Right aligned | &r |

**Note**    With some printers, the minimum margin is determined by the physical characteristics of the printer. For instance, some laser printers cannot print on the top 0.25" or on the left 0.25" of any page.

The base page size for the printer is set when you install the printer or modify the installation. You can use the Printer Setup command on the File menu to determine the page size or modify it.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Platforms

Use this dialog box to specify platforms for external projects when you create a new project workspace after opening an existing makefile not created by Visual C++ version 4. The dialog box appears only if you have installed additional platforms. By default, Microsoft Developer Studio selects all installed platforms.

**Platform**
Select any of the installed platforms to create external projects for those platforms.

**Note**   If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Power Macintosh Network (AppleTalk) Settings

Use this dialog box to set the communication settings to use while debugging a Power Macintosh application across an AppleTalk network.

**Remote Machine Name**
Specify the name of the target Power Macintosh computer.

**Appletalk Zone**
Specify the AppleTalk zone containing the target Power Macintosh computer.

**Debug Monitor Password**
Specify the password for debugging. This password must match the password set in the debug monitor running on the target Power Macintosh computer.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Power Macintosh Network (TCP/IP) Settings

Use this dialog box to set up a connection for remote debugging of a Power Macintosh application across a network using TCP/IP protocol.

**Remote Machine Name**
Type the identifier for the machine running the program you want to debug. The identifier can be the machine name or the machine's TCP/IP address. Under some circumstances (such as when debugging a program running on a Macintosh), TCP/IP name resolution may not be possible. In those circumstances, you must use the TCP/IP address instead of the machine name.

A TCP/IP address is a series of four non-negative integers, separated by periods (for example, 125.12.0.1). Each integer can have up to three digits. Leading zeroes are not required. If you do not know the TCP/IP address for the machine you want to connect to, contact your network administrator.

**Debug Monitor Password**
Type the debug password for the target machine running the program you want to debug. This password must match the password set in the Visual C++ Debug Monitor on the target machine.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Power Macintosh Serial Settings

Use this dialog box to set the communication settings to use while debugging a Power Macintosh application across a serial connection. You must set the same parameter on the Power Macintosh computer you are using for debugging. You do not need to reboot the host workstation or the Power Macintosh after setting this parameter.

**Connector**
Specify the communications port to which you have attached the cable for debugging.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Print

Use this dialog box to print the current file to any printer that you have installed for Windows.

**Printer**
Use the drop-down list box to display a list of installed printers. Select a printer from the list to use that printer.

**Print Range**
Specify printing of the entire source file or just the selected text.

**Setup**
Opens the Document Properties dialog box.

**Tip**    To print a range of selected text, highlight the text before choosing Print from the File menu. Then, under Print Range, select the Selection option button.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Profile

Use this dialog box to examine the behavior of an application when it runs. You can determine which parts of the code it executes, which parts it never executes, the absolute amount of time that it spends executing any part of the code, or the relative amount of time it spends executing any part of the code. You can use this information to determine where to optimize the performance of your code, and to determine what you gained after your optimizations.

**Function Timing**
Lists the amount of time spent in each function.

**Function Coverage**
Lists which functions were executed or not executed.

**Line Coverage**
Lists which lines were executed or not executed.

**Custom**
Enables the controls under Custom Settings so that you can select a profiler command file.

**Merge**
Incorporates the output from the current run into previous runs of the profiler.

**Browse**
Type the name of a batch or command file into the associated text box or use the Browse button to browse for a batch or command file.

**Advanced Settings**
Type profiler options, in their command-line form, in this text box for greater control over your profiling session.

**Note**     If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Project Settings

Use this dialog box to:

- Define the targets of your program.
- Specify how your program is linked to MFC.
- Specify where files are placed.
- Control the behavior of the debugger.
- Control the behavior of the compiler.
- Control the behavior of the linker.
- Control how resources are compiled into .RES files.
- Control how OLE Type Library files (.TLB) are generated from Object Description Language Files (.ODL).
- Specify how browse information is built and managed.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# General Tab

Use the General tab in the Project Settings dialog box to define how your program is linked to MFC, and where files are placed.

**Settings For**
The platform for which you want to set options. You can select one or more platforms.

**Microsoft Foundations Classes**
Specifies how your program is linked to MFC. Choose one of the following options from the drop down list:

Not Using MFC     Your program is not linked with MFC. If you have created an application using one of the wizards, this choice is not available.

Use MFC in a Static Library     Your program is linked with MFC as a static library.

Use MFC in a Shared DLL     Your program is linked with MFC as a shared DLL. If you choose this option, the MFC DLL must be present for your program to work.

**Intermediate Files**
Specifies where files used to build your program, such as object and source files, are located.

**Output Files**
Specifies where final output files are located.

**Note**     If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Custom Build Tab

Use the Custom Build tab in the Project Settings dialog box to specify custom tools for use in project builds. You can specify tools to run on the output file of a project configuration, or you can specify tools for files which have no default tools. For example, in Visual C++, you cannot specify a custom build tool for .CPP files because the build system already specifies the compiler as the default tool for .CPP files. If you have a .L file, however, which is the input for a lexical analysis generator, you can specify how to process it and what output it generates.

You select the topic in the drop-down list under the See Also button to see an example for using custom tools.

**Input File**
Identifies files for which you want to specify custom tools. In the Settings For pane of the Project Settings dialog, you can select:

- a single file from an expanded project configuration
- more than one file from one or more expanded project configurations
- the output file for one or more project configurations by selecting one or more top-level project configuration nodes.

If you select multiple files, the text for this field indicates multiple selections. In the case of a single .L file, the text would show the name of the file relative to the project directory (.\MYLEX.L, for instance).

**Note**    The custom build commands apply only in builds of the project configurations in which you selected the input file or files.

**Description**
Specifies the build step, and is displayed on the Build tab of the Output window during the build.

**Build Commands**
Enter the commands to run. If you enter more than one command, the build system runs them in order from top to bottom. You can use the directory and file macros in these commands.

**Note**    You must enter the complete command to run, with all its required options including the input file and the output file or files. You may want to use a directory macro to specify the location for the output file.

**Output Files**
Enter the names of the output files that you create with the commands in the Build Commands list. The build system checks these files during a build to determine whether they are out of date with respect to the input file. If so, the build system builds them. If the output file or files are subsequently used as input files for the build system, you must also add them to project.

**Note**    You must specify an output file or files. If you do not, the build system has no way to determine that a file is out of date, and thus will never run the custom build tool.

**Directory**
Lists <u>directory macros</u> that you can insert at the current insertion point in the Build Command or Output Files list. Selecting from the list inserts the macro into the command or name that you are currently entering.

**Files**
Lists <u>filename macros</u> that you can insert at the current insertion point in the Build Command or Output Files list. Selecting from the list inserts the macro into the command or name that you are currently entering.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Directory Macros

| Label | Macro | Description |
| --- | --- | --- |
| Intermediate | $(IntDir) | Path to the directory specified for intermediate files, relative to the Project directory |
| Output | $(OutDir) | Path to the directory specified for output files, relative to the Project directory |
| Target | $(TargetDir) | Fully qualified path to the directory specified to output files |
| Input | $(InputDir) | Fully qualified path to the Project directory |
| Project | $(ProjDir) | Fully qualified path to the Project directory |
| Workspace | $(WkspDir) | Fully qualified path to the Project Workspace directory |
| Microsoft Developer | $(MSDevDir) | Fully qualified path to the installation directory for Microsoft Developer Studio |
| Re | $ | Fully |

| mot e Tar get | (Re mot eDir ) | qualified path to the remote output file |
|---|---|---|

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# File Macros

| Label | Macro | Description |
| --- | --- | --- |
| Target Path | $(TargetPath) | Fully qualified name for the project output file |
| Target Name | $(TargetName) | Base name for the output file |
| Input Path | $(InputPath) | Fully qualified name for the input file |
| Input Name | $(InputName) | Base name for the input file |
| Workspace Name | $(WkspName) | Name of the Project Workspace |
| Remote Target Path | $(RemoteTargetPath) | Fully qualified name for the remote output file |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# General Tab (External Project)

Use the General tab for external projects in the Project Settings dialog box to define how an external project is built.

**Settings For**
The platform for which you want to set options. You can select one or more platforms.

**Build Command Line**
The command line that the operating system executes for this project when you choose Build from the Build menu. By default, the system executes Microsoft NMAKE with the /F option followed by the name of the external makefile in the project subdirectory that you have just created. You can revise this field to execute any command, such as any batch file. You could create a batch file, for example, that first changes to the directory with the external project files and then runs NMAKE.

**Rebuild All Options**
The options added to the command line when you choose Rebuild All from the Build menu. By default, /A for Microsoft NMAKE is added.

**Output File Name**
The name of the file that is created when you build a project.

**Browse Info File Name**
The name of the browse information file that is created when you build this project. It must have the file extension .BSC.

**ClassWizard File Name**
The name of the ClassWizard file that has MFC information for this project.

**Build Working Directory**
The working directory for files during the build.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Debug Tab: General Category

The Debug tab contains two categories of information:

- General
- <u>Additional DLLs</u>

You can change the category of information currently displayed using the dropdown list labeled Category.

**Settings For**
A list of debugging targets for which you can view or modify settings. Clicking on one of the folders causes the target executable to appear in the Executable for Debug Session field.

**Category**
Use this dropdown list to change the category of settings displayed on the Debug tab.

**Executable For Debug Session**
Edit this field to specify the executable file you want to debug.

**Working Directory**
Edit this field to specify the directory in which debugging occurs. If you do not specify a directory, debugging occurs in the directory where the executable is located.

**Program Arguments**
Edit this field to specify command-line arguments you want to pass to the program at startup. The program receives these arguments when started with the Go or Restart command. You can use I/O redirection in this field using CMD.EXE format.

**Note**    This text box does not appear for DLL project types.

**Remote Executable Path And File Name**
Edit this field to specify the location of an application on a remote target computer.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Debug Tab: Additional DLLs Category

The Debug tab contains two categories of information:

- <u>General</u>
- Additional DLLs

You can change the category of information currently displayed using the dropdown list labeled Category.

Use the Additional DLLs Category to supply information for debugging a DLL called by an executable program or another DLL. This information enables the debugger to load symbolic information for the DLL.

You do not need to use this dialog for DLLs loaded implicitly with IMPLIB.

**Settings For**
A list of debugging targets whose settings you can view or change. Click on one of the folders to choose the target whose settings you want to modify or view.

**Category**
Use this dropdown list to change the category of settings displayed on the Debug tab.

**Modules**
This field becomes active when you select one of the targets in the Settings For list. It contains an unlabeled checkbox column, a Local Name column, and a third column, Remote Name, that appears only when you are remote debugging. To make a row active, select any cell in that row by double-clicking. When a row is active, a button with an ellipsis (...) appears in the rightmost cell. Once you have selected a cell, you can use TAB to move to the next cell or SHIFT+TAB to move back to the previous cell.

Select the checkbox in the first (leftmost) column to load a DLL prior to the debugging session. Preloading is useful if you want to set breakpoints in a DLL that is not loaded implicitly using IMPLIB or at startup. You can use this option to preload symbols even after debugging has begun.

**Local Name**
Edit this field to specify the name of a DLL you want to debug.

**Remote Name**
Edit this field, which appears only during remote debugging, to map a local DLL to a file on a remote machine.

**...**
Select this button to use the <u>Find Local DLLs dialog</u> to add a DLL to the list.

**Try to locate other DLLs**
If this check box is selected, the debugger asks for additional DLLs when debugging begins.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

## C/C++ Tab

Use the C/C++ tab in the Project Settings dialog box to specify:

- The most frequently used compiler options.
- The CPU, run-time library, calling convention, and structure alignment.
- Options that speed compile time. They allow you to precompile any C or C++ code (including inline code).
- Options that determine how the compiler fine tunes the performance of your program.
- Options that turn off Microsoft language extensions, enable function-level linking, eliminate duplicate strings, enable minimal rebuild, enable incremental compilation, and suppress displaying the startup banner and informational messages.
- An inheritance representation for the C++ pointers to class members in your application, control exception handling, enable RTTI (run-time type information), and control the generation of hidden virtual constructor/destructor displacement fields in classes with virtual bases.
- Options that control symbols, macros, and include paths used by the preprocessor.
- Options that generate output files for browse information files and code listing files.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# C/C++ Tab: General Category

These are the most frequently used compiler options.

**Settings For**
A list of targets and files whose project settings you can define. Choose a target or file to view or modify current settings.

**Reset**
Resets the project settings of a target or a file back to the settings that existed when the target or file was created.

**Warning Level**
Specifies the severity of warning for which the compiler generates messages.

**Warnings As Errors**
Select this check box to have warning messages emitted as error messages.

**Generate Browse Info**
Select this check box to generate intermediary (.SBR) browse files when you build your program. To create the .BSC file required for browsing, select the Build Browse Info File check box on the Browse Info tab as well.

**Debug Info**
Specify the type of debugging information generated by the compiler.

**Optimizations**
Specify one of four predetermined optimization categories: Default, Disable, Maximize Speed, and Minimize Size. If you specify Customize from this, the General category, you can use a list box in the Optimizations category to specify a custom set of optimizations.

**Preprocessor Definitions**
Create one or more preprocessor macros.

**Project, Source File, or Common Options**
Named "Project Options" if a single project is selected in the left pane. Displays and allows editing of project options.

Named "Source File Options" if a single source file is selected in the left pane. Displays and allows editing of source-file options.

Named "Common Options" if multiple projects or files are selected in the left pane. Displays the options that are common to the selections.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# C/C++ Tab: Code Generation Category

These options specify the CPU, run-time library, calling convention, and structure alignment.

**Settings For**
A list of targets and files whose project settings you can define. Choose a target or file to view or modify current settings.

**Reset**
Resets the project settings of a target or a file back to the settings that existed when the target or file was created.

**Processor**
Specify code generation that compliments the 80386, 80486, or the Pentium processor. The default setting is Blended, which uses instructions that work best for all processors. Each selection specifies a different value for the **_M_IX86** preprocessor macro.

**Use Run-Time Library**
Specify a run-time library for singlethread or multithread programs.

**Calling Convention**
Specify the calling convention applied to functions not marked as **__cdecl**, **__stdcall**, or **__fastcall**.

**Struct Member Alignment**
Specify whether the members of a structure are packed into memory on the smaller of the size of the member type or on 1-, 2-, 4-, 8-, or 16-byte boundaries.

**Project, Source File, or Common Options**
Named "Project Options" if a single project is selected in the left pane. Displays and allows editing of project options.

Named "Source File Options" if a single source file is selected in the left pane. Displays and allows editing of source-file options.

Named "Common Options" if multiple projects or files are selected in the left pane. Displays the options that are common to the selections.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# C/C++ Tab: Code Generation Category

These options specify the CPU, run-time library, calling convention, and structure alignment for a 680x0 Macintosh application.

**Memory Model**
Select a memory model from this list to determine the maximum size of code and data segments used by your Macintosh application.

**Instruction Type**
Select from this list to generate code optimized for a specific processor: 68020, 68030, or 68040.

**Calling Convention**
Use this list to select a calling convention for your application: **__cdecl**, **__fastcall**, **__stdcall**, or **__pascal**.

**Global Threshold**
Type a value in this box to set the global threshold for your application. The compiler allocates data items that are smaller than the global threshold in near data. All other data items are allocated in far data.

**Generate Swapper Prolog/Epilog**
Select this option to generate swappable code segments. When this option is set, Visual C++ for Macintosh generates special function prolog and epilog sequences for designated functions.

**Generate Macsbug Symbols**
Select this checkbox to generate symbols for use with Apple's Macsbug low-level debugger.

**Order Bitfields Low to High**
Select this checkbox to reverse bit-field ordering from bid endian to little endian.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# C/C++ Tab: Code Generation Category

These options specify the CPU, run-time library, calling convention, and structure alignment for a Power Macintosh application.

**Processor**
Select from this list to generate code optimized for one of these specific PowerPC processors: 601, 603, 604, or 620.

**Use Runtime Library**
Use this list to specify the run-time library to use for the application. The options include static libraries and DLLs, with debug and non-debug versions of each.

**Struct Member Alignment**
Use this list to specify how the compiler is to align struct members to memory-address boundaries (4 bytes, 8 bytes, or 16 bytes). Aligning struct members to 8- or 16-byte boundaries results in better optimized memory accesses and faster code.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# C/C++ Tab: Precompiled Headers Category

These options speed compile time. They allow you to precompile any C or C++ code (including inline code).

**Settings For**
A list of configurations and files whose project settings you can define. Choose a configuration or file to view or modify current settings.

**Reset**
Resets the project settings of a configuration or a file back to the settings that existed when the configuration or file was created.

**Not Using Precompiled Headers**
Disables precompiled headers.

**Automatic Use of Precompiled Headers**
Creates and uses precompiled header files using the Automatic Use of Precompiled Headers system.

**Through Header**
The compiler compiles all code up to and including the header file (.H) specified in this text box and places it in a file with a .PCH extension.

**Create .PCH File**
Creates a precompiled header file (.PCH). Select Use .PCH File to use the created .PCH file.

**Through Header**
The compiler compiles all code up to and including the header file (.H) specified in this text box and places it in a file with a .PCH extension.

**Use .PCH File**
Uses a precompiled header file (PCH).

**Through Header**
The compiler assumes that all code through the header file specified in this text box is precompiled.

**Project, Source File, or Common Options**
Named "Project Options" if a single project is selected in the left pane. Displays and allows editing of project options.

Named "Source File Options" if a single source file is selected in the left pane. Displays and allows editing of source-file options.

Named "Common Options" if multiple projects or files are selected in the left pane. Displays the options that are common to the selections.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# C/C++ Tab: Optimizations Category

These options determine how the compiler fine tunes the performance of your program.

**Settings For**
A list of configurations and files whose project settings you can define. Choose a configuration or file to view or modify current settings.

**Reset**
Resets the project settings of a configuration or a file back to the settings that existed when the configuration or file was created.

**Optimizations**
Specify one of four predetermined optimization categories (Default, Disable, Maximize Speed, and Minimize Size) or specify Customize to select individual optimizations from the enabled list box.

**Inline-Function Expansion**
Suggest to the compiler the type of functions to expand inline.

**Project, Source File, or Common Options**
Named "Project Options" if a single project is selected in the left pane. Displays and allows editing of project options.

Named "Source File Options" if a single source file is selected in the left pane. Displays and allows editing of source-file options.

Named "Common Options" if multiple projects or files are selected in the left pane. Displays the options that are common to the selections.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# C/C++ Tab: Customize Category

These options turn off Microsoft language extensions, enable function-level linking, eliminate duplicate strings, enable minimal rebuild, enable incremental compilation, and suppress displaying the startup banner and informational messages.

**Settings For**
A list of configurations and files whose project settings you can define. Choose a configuration or file to view or modify current settings.

**Reset**
Resets the project settings of a configuration or a file back to the settings that existed when the configuration or file was created.

**Disable Language Extensions**
The compiler uses ANSI C rules. If this check box is cleared, the compiler allows use of the Microsoft C language extensions.

**Enable Function-Level Linking**
Use the linker to exclude and/or order individual functions in a DLL or executable file.

**Eliminate Duplicate Strings**
Enables the compiler to place a single copy of identical strings into the executable file. This feature is also called "string pooling."

**Enable Minimal Rebuild**
Enables detection of changes to C++ class definitions and whether the changes require recompilation of source files.

**Enable Incremental Compilation**
Enables function-level recompilation.

**Suppress Startup Banner and Informational Messages**
Suppresses display of the sign-on banner and informational messages.

**Project, Source File, or Common Options**
Named "Project Options" if a single project is selected in the left pane. Displays and allows editing of project options.

Named "Source File Options" if a single source file is selected in the left pane. Displays and allows editing of source-file options.

Named "Common Options" if multiple projects or files are selected in the left pane. Displays the options that are common to the selections.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# C/C++ Tab: C++ Language Category

These options specify an inheritance representation for the C++ pointers to class members in your application, control exception handling, and control the generation of hidden virtual constructor/destructor displacement fields in classes with virtual bases.

**Settings For**
A list of configurations and files whose project settings you can define. Choose a configuration or file to view or modify current settings.

**Reset**
Resets the project settings of a configuration or a file back to the settings that existed when the configuration or file was created.

**Representation Method**
Specify the method used by the compiler to represent pointers to class members.

**General-Purpose Representation**
When the representation method is General-Purpose Always, you must also specify the inheritance model of classes.

**Enable Exception Handling**
Destructors are called for automatic objects during a stack unwind caused by either a Windows NT-based structured exception or a C++ exception.

**Enable RTTI**
Causes the compiler to add code to check object types at runtime (run-time type information).

**Disable Construction Displacements**
Suppress the vtordisp constructor/destructor displacement member if you are certain that all class constructors and destructors call virtual functions virtually.

**Project, Source File, or Common Options**
Named "Project Options" if a single project is selected in the left pane. Displays and allows editing of project options.

Named "Source File Options" if a single source file is selected in the left pane. Displays and allows editing of source-file options.

Named "Common Options" if multiple projects or files are selected in the left pane. Displays the options that are common to the selections.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# C/C++ Tab: Preprocessor Category

These options control symbols, macros, and include paths used by the preprocessor.

**Settings For**
A list of configurations and files whose project settings you can define. Choose a configuration or file to view or modify current settings.

**Reset**
Resets the project settings of a configuration or a file back to the settings that existed when the configuration or file was created.

**Preprocessor Definitions**
Creates one or more preprocessor macros.

**Undefined Symbols**
Undefines a previously defined macro.

**Undefine All Symbols**
Undefines every previously defined macro.

**Additional Include Directories**
Adds one or more directories to the list of directories searched for include files.

**Ignore Standard Include Paths**
Prevents the compiler from searching for include files in directories specified in the PATH and INCLUDE environment variables.

**Project, Source File, or Common Options**
Named "Project Options" if a single project is selected in the left pane. Displays and allows editing of project options.

Named "Source File Options" if a single source file is selected in the left pane. Displays and allows editing of source-file options.

Named "Common Options" if multiple projects or files are selected in the left pane. Displays the options that are common to the selections.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# C/C++ Tab: Listing Files Category

These options generate output files for browse information files and code listing files.

**Settings For**
A list of configurations and files whose project settings you can define. Choose a configuration or file to view or modify current settings.

**Reset**
Resets the project settings of a configuration or a file back to the settings that existed when the configuration or file was created.

**Generate Browse Info**
Select this check box to generate intermediary (.SBR) browse files when you build your program. To create the .BSC file required for browsing, select the Build Browse Info File check box on the Browse Info tab as well.

**Intermediate Browse Info File Name**
Specify a directory and/or filename for the .SBR and .BSC files generated by selecting the Generate Browse Info option.

**Exclude Local Variables**
Generates browse information files containing complete symbolic information, minus information on local variables, that you can examine in browse windows.

**Listing File Type**
Specify generation of a listing file.

**Listing File Name**
Specify a directory and/or filename for the listing file selected from the Listing File Type list box.

**Project, Source File, or Common Options**
Named "Project Options" if a single project is selected in the left pane. Displays and allows editing of project options.

Named "Source File Options" if a single source file is selected in the left pane. Displays and allows editing of source-file options.

Named "Common Options" if multiple projects or files are selected in the left pane. Displays the options that are common to the selections.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

## Link Tab

Use the Link tab in the project settings dialog box to specify:

- The most frequently used linker options.
- Options that control the linker when producing a Win32 configuration.
- Custom link options.
- Options that control generation of debugging information and mapfile output.
- Input files to the linker.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

## Link Tab: General Category

These are the most frequently used linker options.

**Settings For**
A list of configurations and files whose project settings you can define. Choose a configuration or file to view or modify current settings.

**Reset**
Resets the project settings of a configuration or a file back to the settings that existed when the configuration or file was created.

**Output File Name**
Overrides the default name and location of the program that the linker creates.

**Object/Library Modules**
Specifies an object file or standard library (either static or import) to pass to the linker.

**Generate Debug Info**
Generates debugging information for the executable file or DLL.

**Link Incrementally**
Enables incremental linking.

**Enable Profiling**
Creates an output file that can be used with the Microsoft 32-Bit Source Profiler.

**Ignore All Default Libraries**
Removes all default libraries from the list of libraries the linker searches when resolving external references.

**Generate Mapfile**
Creates a mapfile.

**Project or Common Options**
Named "Project Options" if a single project is selected in the left pane. Displays and allows editing of project options.

Named "Common Options" if multiple projects or files are selected in the left pane. Displays the options that are common to the selections.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Link Tab: Output Category

These options control the linker when producing a Win32 configuration.

**Settings For**
A list of configurations and files whose project settings you can define. Choose a configuration or file to view or modify current settings.

**Reset**
Resets the project settings of a configuration or a file back to the settings that existed when the configuration or file was created.

**Base Address**
Specifies a base address for a program. The specified address overrides the default location for an executable file (at 0x400000) or a DLL (at 0x10000000).

**Entry-Point Symbol**
Specifies a starting address for an executable file or DLL.

**Reserve**
Specifies the total stack allocation, in bytes, of virtual memory. The default stack size is 1 MB.

**Commit**
Specifies the amount, in bytes, of physical memory to allocate from the reserve memory.

**Major**
Specifies the major part of a version number, that which precedes the decimal point, to insert into the header of an executable file or DLL.

**Minor**
Specifies the minor part of a version number, that which follows the decimal point, to insert into the header of an executable file or DLL.

**Project or Common Options**
Named "Project Options" if a single project is selected in the left pane. Displays and allows editing of project options.

Named "Common Options" if multiple projects or files are selected in the left pane. Displays the options that are common to the selections.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Link Tab: Customize Category

These options allow you to specify use of a program database, force the creation of an output file, control incremental linking, specify the name for the output file, print progress messages, and prevent display of copyright and version information at startup.

**Settings For**
A list of configurations and files whose project settings you can define. Choose a configuration or file to view or modify current settings.

**Reset**
Resets the project settings of a configuration or a file back to the settings that existed when the configuration or file was created.

**Link Incrementally**
Enables incremental linking.

**Use Program Database**
Places debugging information in a program database (PDB).

**Program Database Name**
Specifies the filename for the program database (PDB).

**Force File Output**
Generates a valid executable file or DLL even if a symbol is referenced but not defined or is multiply defined.

**Output File Name**
Overrides the default name and location of the program that the linker creates.

**Print Progress Messages**
Displays details about the linking process during a build.

**Suppress Startup Banner**
Prevents display of the copyright message and version number during a build.

**Project or Common Options**
Named "Project Options" if a single project is selected in the left pane. Displays and allows editing of project options.

Named "Common Options" if multiple projects or files are selected in the left pane. Displays the options that are common to the selections.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Link Tab: Output Category

These options control the linker when producing a 680x0 Macintosh configuration.

**File Creator**
Use this box to set the Macintosh file creator attribute. The default file creator is **????** (four question marks).

**File Type**
Use this box to set the Macintosh file type attribute. The default file type is **APPL** (application).

**Macintosh Data Fork File Name**
Use this box to set the path to a file on the Windows workstation to be copied to the data fork of the Macintosh file. Use this box only if you need to copy a separate file directly to the data fork.

**Set Bundle Bit**
Select this checkbox to enable the bundle bit, which tells the Finder that the application has a bundle resource designating which icons to use for the application and associated documents.

**Update Remote File Only on Demand**
Select this checkbox to update the remote file only when you choose the Update Remote Output File command from the Tools menu.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Link Tab: Output Category

These options control the linker when producing a Power Macintosh configuration.

**File Creator**
Use this box to set the Macintosh file creator attribute. The default file creator is **????** (four question marks).

**File Type**
Use this box to set the Macintosh file type attribute. The default file type is **APPL** (application).

**Macintosh Data Fork File Name**
Use this box to set the path to a file on the Windows workstation to be copied to the data fork of the Macintosh file. Use this box only if you need to copy a separate file directly to the data fork.

**Init Routine**
Use this box to specify a routine to be called by the Macintosh operating system when the application or DLL is loaded.

**Term Routine**
Use this box to specify a routine to be called by the Macintosh operating system when the application or DLL is unloaded.

**Set Bundle Bit**
Select this checkbox to enable the bundle bit, which tells the Finder that the application has a bundle resource designating which icons to use for the application and associated documents.

**Update Remote File Only on Demand**
Select this checkbox to update the remote file only when you choose the Update Remote Output File command from the Tools menu.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Link Tab: Debug Category

These options control generation of debugging information and mapfile output.

**Settings For**
A list of configurations and files whose project settings you can define. Choose a configuration or file to view or modify current settings.

**Reset**
Resets the project settings of a configuration or a file back to the settings that existed when the configuration or file was created.

**Mapfile Name**
Specifies a name for a mapfile other than the default.

**Generate Mapfile**
Creates a mapfile.

**Generate Debug Info**
Generates debugging information for the executable file or DLL.

**Microsoft Format**
Generates new-style Microsoft debugging information.

**COFF Format**
Generates COFF-style debugging information.

**Both Formats**
Generates both COFF debugging information and old-style Microsoft debugging information.

**Project or Common Options**
Named "Project Options" if a single project is selected in the left pane. Displays and allows editing of project options.

Named "Common Options" if multiple projects or files are selected in the left pane. Displays the options that are common to the selections.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Link Tab: Input Category

These options allow you to specify modules to pass to the linker, specify that libraries be ignored at link time, force linking, and use an MS-DOS stub program.

**Settings For**
A list of configurations and files whose project settings you can define. Choose a configuration or file to view or modify current settings.

**Reset**
Resets the project settings of a configuration or a file back to the settings that existed when the configuration or file was created.

**Object/Library Modules**
Specifies an object file or standard library (either static or import) to pass to the linker.

**Ignore Libraries**
Specifies one or more default libraries to remove from the list of libraries the linker searches when resolving external references.

**Ignore All Default Libraries**
Removes all default libraries from the list of libraries the linker searches when resolving external references.

**Force Symbol References**
Specifies one or more symbols to add to the symbol table. Forces linking with the object that contains the symbol definition(s).

**MS-DOS Stub File Name**
Specifies the name of an MS-DOS stub program to attach to a Win32 program.

**Project or Common Options**
Named "Project Options" if a single project is selected in the left pane. Displays and allows editing of project options.

Named "Common Options" if multiple projects or files are selected in the left pane. Displays the options that are common to the selections.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Resources Tab

Use the Resources tab in the Project Settings dialog box to control how resources are compiled from .RC files into .RES files.

**Settings For**
A list of configurations and files whose project settings you can define. Choose a configuration or file to view or modify current settings.

**Reset**
Resets the project settings of a configuration or a file back to the settings that existed when the configuration or file was created.

**Resource File Name**
Names the output file (.RES). Developer Studio assumes a path relative to the project directory.

**Language**
Specifies the default language for all resources.

**Additional Include Directories**
Adds one or more directories to the list of directories that are searched for include files.

**Ignore Standard Include Paths**
Prevents the compiler from searching for include files in directories specified by the INCLUDE environment variable.

**Preprocessor Definitions**
Creates one or more preprocessor macros for use with the **#ifdef** directive.

**Project, Common, or Source File Options**
Named "Project Options" when a single configuration is selected in the left pane. This command displays and allows editing of the options set for the selected configuration.

Named "Common Options" when multiple projects are selected in the left pane. This command displays the options common to the selected projects.

Named "Source File Options" when a single resource file (.RC) is selected in the left pane. This command displays the options selected for the file.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Macintosh Resources Tab

Use the Macintosh Resources tab to view and set Macintosh Resource Compiler options.

**Resource File Name**
Use this box to specify a file name and path for the Macintosh resource corresponding to the selected .R file. By default, the Macintosh resource has the same base name as the .R script file, but a different extension. If the script file is **pooh.r**, for example, the resource file is **pooh.rsc**.

**Read Only Output**
Select this checkbox to make the generated resource read-only. Read-only resources cannot be accidentally overwritten by subsequent build operations.

**Preprocessor Definitions**
Use this box to specify symbols you want to define. Use commas to separate multiple symbols.

**Language**
Use this box to specify the natural language (such as US English, Australian English, or Mexican Spanish) used in the resource.

**Additional Include Paths**
Use this box to specify the path for text files specified by a **#include** directive in the resource script file.

**Resource Files Path**
Use this box to specify the path for binary Macintosh files specified by **#include** statements in the resource script file. Files specified with the include statements must not be in the paths defined in the Directories tab for the project.

**Alignment**
Use this box to set the alignment of resources on two-byte or four-byte boundaries.

**Project Options**
Use this box to view and set command-line options for the Macintosh Resource Compiler.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Macintosh Resources Tab

Use the Macintosh Resources tab to view and set Macintosh Resource Compiler options.

**Resource File Name**
Use this box to specify a file name and path for the Macintosh resource corresponding to the selected .R file. By default, the Macintosh resource has the same base name as the .R script file, but a different extension. If the script file is **pooh.r**, for example, the resource file is **pooh.rsc**.

**Read Only Output**
Select this checkbox to make the generated resource read-only. Read-only resources cannot be accidentally overwritten by subsequent build operations.

**Preprocessor Definitions**
Use this box to specify symbols you want to define. Use commas to separate multiple symbols.

**Language**
Use this box to specify the natural language (such as US English, Australian English, or Mexican Spanish) used in the resource.

**Additional Include Paths**
Use this box to specify the path for text files specified by a **#include** directive in the resource script file.

**Resource Files Path**
Use this box to specify the path for binary Macintosh files specified by **#include** statements in the resource script file. Files specified with the include statements must not be in the paths defined in the Directories tab for the project.

**Alignment**
Use this box to set the alignment of resources on two-byte or four-byte boundaries.

**Project Options**
Use this box to view and set command-line options for the Macintosh Resource Compiler.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# OLE Types Tab

Use the OLE Types tab in the Project Settings dialog box to control the generation of a Type Library file (.TLB) from an Object Description Language file (.ODL).

**Settings For**
A list of configurations and files whose project settings you can define. Choose a configuration or file to view or modify current settings.

**Reset**
Resets the project settings of a configuration or a file back to the settings that existed when the configuration or file was created.

**Output File Name**
Specifies a name and/or location for the Object Description Language file (.ODL) other than the default. This option is available only if you select an .ODL file in the Source Files folder of the Settings For list box.

**Output Header File Name**
Generates an include file (.H) that contains the types definitions from the Object Description Language file (.ODL). The generated file can be included in a C or a C++ file.This option is available only if you select an .ODL file in the Source Files folder of the Settings For list box.

**Additional Include Directories**
Specifies additional directories in which to search for include files.

**Preprocessor Definitions**
Define one or more preprocessor macros for use with the **#ifdef** directive.

**Suppress Display Banner**
Prevents the display of the MKTYPELIB startup banner.

**Project, Source File, or Common Options**
Named "Project Options" if a single project is selected in the left pane. Displays and allows editing of project options.

Named "Source File Options" if a single source file is selected in the left pane. Displays source-file options.

Named "Common Options" if multiple projects or files are selected in the left pane. Displays the options that are common to the selections.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Browse Info Tab

Use the Browse Info tab in the Project Settings dialog box to specify how browse information is built and managed.

**Settings For**
A list of configurations and files whose project settings you can define. Choose a configuration or file to view or modify current settings.

**Reset**
Resets the project settings of a configuration or a file back to the settings that existed when the configuration or file was created.

**Browse Info File Name**
Specifies the name and location of the browse information (.BSC) file that Developer Studio will build.

**Build Browse Info File**
Select this checkbox to build a browse information file (BSC) each time you build a program. Make sure the Generate Browse Information checkbox on the appropriate language tab is selected also, so that for each source file, the compiler generates .SBR files used to create the browse information file. Click the See Also button for a list of topics that include information about browse information settings.

**Suppress Startup Banner**
Prevents the display of the startup banner.

**Project or Common Options**
Named "Project Options" if a single project is selected in the left pane. Displays and allows editing of project options.

Named "Common Options" if multiple projects or files are selected in the left pane. Displays the options that are common to the selections.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# QuickWatch

Use this dialog box to quickly view or modify variables or expressions or to add variables or expressions to the Watch window.

To evaluate a variable or expression in QuickWatch, type it into the Expression text box, then press ENTER or click on the Recalc button. The expression or variable name and its current value appear in the spreadsheet-like Current Value field.

If the Current Value field contains an array, object, or structure variable, a button appears next to the expression or variable name. By clicking on the button, you can expand or contract your view of the variable. The button displays a plus sign (+) when the variable is displayed in contracted form, a minus sign when it is displayed in expanded form.

To modify the value of a variable, place the variable into the Current Value field as described above. Then, use the TAB key to select the spreadsheet cell containing the value or double-click on the cell. Type the new value and press ENTER.

To add a variable or expression to the Watch window, type it into the Expression text box, then click on the Add Watch button.

When you evaluate an expression or variable, or add it to the Watch window using the Add Watch button, QuickWatch adds the variable to a dropdown list that appears below the Expression text box. You can recall the expression or variable at a later time by selecting it from the list. You can then use the Recalc button to recalculate it or the Add Watch button to add it to the Watch window.

The QuickWatch dialog supports editing functions. You can paste information into QuickWatch from another window or copy information from the Current Value field and paste it into a window.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Row and Column Behavior

To autosize a column to fit its contents, double-click on the vertical divider at the column edge. To size a column manually, drag the vertical divider left or right.

**Note**    Rows are sized to fit the current font and cannot be resized manually. To change the font size, use the Fonts and Colors tab of the Options command from the Tools menu.

**Expression**
The variable or expression to display or modify.

**Current Value**
Displays the expression from the Expression text box under Name its value (or values for an array, object, or structure) under Value.

**Recalc**
Calculates the value of an expression in the Expression text box, placing the result into the Current Value text box.

**Note    Add Watch**
Adds the variable or expression to the Watch window. If the Watch window is not displayed, it also displays the window.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Remote Connection

Use this dialog box to establish a connection to another computer.

**Platform**
Specifies the type of computer you want to connect to.

**Connection**
Specifies the type of communication transport you want to use to connect your computers.

**Settings**
Opens the dialog box you use to specify communications settings for the transport you have chosen.

**Note    Attach Now**
This button is enabled only if you select Power MacIntosh as the platform. Select this button to debug an active process.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Remote Executable Path and File Name

This dialog box appears if you try to debug a remote executable without specifying the path and file name in the General category on the Debug tab of the Project Settings dialog. Remote debugging cannot begin until you specify the path and file name, either in the Project Settings or in this dialog.

**Remote Executable Path and File Name**
Specifies the path and file of the remote executable to debug.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Remove from Source Code Control

Use this dialog box to remove files from your source-code control project.

**Files**
Lists all the files that you have checked in, with a check next to every checked-in file   that you have selected in FileView. Click the check boxes to change your selection.

**Comment**
Text describing the removal: reasons for removing the file from source code control, the person removing the files, and so on.

**Note**    This option may not be present on your dialog box, because not all source-code control systems support comments when removing files.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Replace

Use this dialog box to:

- Find and replace a symbol or a sequence of symbols in a file.
- Find and replace all symbols or sequences of symbols that match a given pattern.

**Find What**
Specifies the search text or the regular expression to match. You can use the menu button to the right of the drop-down list to display a list of regular search expressions. When you select an expression in this list, the expression is substituted as search text in the Find What text box. If you use regular expressions, be sure the Regular Expression option is checked.

You can also use the drop-down list to select from a list of up to 16 previous search strings.

**Replace With**
The string of characters to replace the characters found. You cannot use regular expressions in this string.

**Match Whole Word Only**
Matches text strings only if they are preceded and followed by a space, tab, or punctuation character, or the start or end of a line. Otherwise, the command matches any string, whether it is a fragment of a larger string or not.

**Match Case**
Finds only text strings that match the case of the characters in the Find What string exactly. Otherwise, the command finds strings with either uppercase or lowercase characters that match the characters in the Find What string.

**Regular Expression**
Check this option if you use regular expressions in the Find What text box.

**Replace In**
Specifies whether to replace the strings in the current selection or the entire file.

**Find Next**
Repeats the most recent find operation.

**Replace**
Replaces the currently selected string that matches the string specified in the Find What text box.

**Replace All**
Replaces all strings that match the Find What string automatically, without requiring confirmation for each replacement.

**Tips and Troubleshooting**    It is a good idea to test regular expressions using only the Find command before using the Replace command, because they may match expressions that you don't intend to match. You should test them especially if you intend to use the Replace All option, because you can only Undo the last replacement if your regular expression replaces unintended matches.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Resolve Ambiguity

Use this dialog box to select an instance of a symbol you want to examine. This dialog box is displayed when there are multiple instances of an identical or overloaded symbol.

**Symbols**
Lists the instances in which an identical symbol occurs. Choose one of the instances to examine.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Resolve Source Line Ambiguity

Use this dialog box to select an instance of a source line when code you are debugging includes multiple instances of an identical source line.

**Instances**
Lists the instances in which an identical source line occurs. Choose one of the instances.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Sample Application

Use the Sample Application dialog box to browse and copy the source files that make up a sample project. You can browse the source files (header files, resource script files, make files, graphics files, and so on) associated with the sample programs. Microsoft Developer Studio displays ASCII text files in a source file window. If the source file has a non-ASCII format, Microsoft Developer Studio uses an appropriate browser.

**Files**
This box displays a list of all the files that make up the sample project. If you select a source file (.CPP, .H, .RC, and so on) the View button will be enabled. If you select an executable file (.EXE), the View button will change to a Run button.

**View**
Choose this button to view the file currently highlighted in the Files box.

**Run**
This button will only be available if the current selection in the Files box is an executable file. Choose this button to run the selected application.

**Copy**
Choose this button to copy the file currently selected in the Files box to another location, usually your local hard disk.

**Copy All**
Choose this button to copy all the files listed in the Files box to another location, usually your local hard disk.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Save As

Use this dialog box to save the contents of the active file in a different file, or the displayed text in a non-editor window in a text file. You can use this feature to make a template file with **#include** and **#define** directives, for instance, and then quickly copy this information to other files, while keeping the original open for further revisions.

**Save In**
Select a folder to save the file in. You can select network paths, computers, drives, directories, and folders.

**File Name**
Type the new filename for this file.

**Note**   If you type a pattern using a wildcard (* or ?) in the File Name box and press ENTER, the list box displays files matching that pattern in the selected folder.

**Save As Type**
Select a file type from the drop-down list box. The filename extension determines the file type of the saved file.

**Save**
Saves the designated file.

**Tips and Troubleshooting**   If you open the Save As dialog box while an Output, Locals, Watch, Registers, or Browse window is active, the Save As dialog box proposes the name of the window followed by the extension .TXT for the filename in which to save the text displayed in the window. In the case of the Browse window, it does not save the .BSC file in .BSC format under a different name.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Save Palette Colors

Use this dialog box to save the current custom color palette as a .PAL file. Under Windows 95, you get help on the following controls by selecting the control and then pressing F1:

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Select Command File

Use this dialog box to change drives and directories to select a profiler command file. A profiler command file is a batch file containing commands to run the software components of the profiler with customized switch settings. Command files allow much greater versatility in using the profiler.

Under Windows 95, you get help on the following controls by selecting the control and then pressing F1:

**Look in**
**File name**
**Files of type**

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Select Dialog Font

Use this dialog box to select the font to be used when displaying your dialog box.

**Font**
Select a font from the list of available fonts.

**Size**
Type the font size you want or select from the list of available font sizes.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Subprojects

Use this dialog box to add a subproject to an existing project in your project workspace, or to remove a subproject from a project. You can also create a new project and add it as a subproject.

**Select project to modify**
Select a project from the list of existing projects. You add the subprojects that you select or create to this project, and/or remove subprojects from this project.

**Select subprojects to include**
Select the subproject or subprojects to add to or delete from the project. If the project has a check next to it, it will be added; if it has no check, it will be removed. This list includes all of the projects in the project workspace which can become a subproject of the project that you have selected to modify. The list does not display projects that would establish a circular subproject relationship if Project_A has Project_B as a subproject, and Project_B has Project_C as a subproject, when you choose Project_C as the project to modify, neither Project_A nor Project_B appear in this list, because adding either of them as a subproject of Project_C would create a circular subproject relationship.

**New**
Displays the Insert Project dialog box to create a new project, with the subproject option selected and the project chosen to modify selected in the dropdown list.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Resource Includes

Use this dialog box to:

- Change the name of the symbols header file.
- Include files containing shared (read-only) or calculated symbols.
- Include files containing resources that are added to your project at compile time.

**Symbol Header File**
Type the name of the symbols file you want to use.

**Read-Only Symbol Directives**
Use this box to include .H files containing shared (read-only) or calculated symbols.

**Compile-Time Directives**
Use this box to include resource files that are added to your project at compile time.

**Tips and Troubleshooting**    What you type in the Read-Only Symbol Directives box or the Compile-Time Directives box is included in the resource file exactly as you type it. Make sure what you type does not contain any spelling or syntax errors.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Resource Symbols (Symbol Browser)

Use this dialog box to:

- Examine the symbols in your current resource file: their names, their values, and whether they are in use.
- Add or change symbols.
- Move to the appropriate editor for the resource that uses the symbol.

**Name**
The name of the symbol.

**Value**
The numeric value of the symbol.

**In Use**
A check mark in this column indicates that the symbol is being used by one or more resources. The resource or resources are listed in the Used By box.

**Show Read-Only Symbols**
Normally the Symbol Browser displays only the modifiable resources in your resource script file. Select this option to display read-only resources as well. Modifiable resources appear in bold text; read-only resources appear in plain text.

**Used By**
Shows the resource or resources using the symbol selected in the symbols list. To move to the editor for a given resource, select the resource in the Used By box and choose View Use.

**New**
Opens the New Symbol dialog box, which you use to define the name and, if necessary, a value for the symbolic resource identifier.

**Change**
Opens the Change Symbol dialog box, which you use to change the name or value of a symbol.

**Delete**
Deletes the symbol selected in the Name/Value/In Use list box.

**View Use**
Opens a source document window that contains code associated with the resource or identifier selected in the Used By box.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Text Tool Font

Use this dialog box to select the font to be with the text tool in the image editor.

**Font**
Select a font from the list of available fonts.

**Font Style**
Select the font style you want from the list of available font styles.

**Size**
Type the font size you want or select from the list of available font sizes.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Threads

Use this dialog box to select a single thread to debug, if your program has multiple threads. While debugging programs with multiple threads, you can select a thread to debug, suspend a thread, resume running a suspended thread, or terminate a thread.

**Thread List**
The Thread List displays status information on each thread as follows:

Thread ID    This is the DWORD that uniquely identifies the thread. When you set focus on a thread, it is displayed in the Thread List with an asterisk next to its identifier.

Suspend    This can be a value from 0 to 127.

Priority    This can be one of seven priorities: Idle, Lowest, Below Normal, Normal, Above Normal, Highest, or Time Critical.

Location    This is the current address of the thread, displayed either as a function name or as an address, depending on the state of the Current Location Display option at the bottom of the dialog box.

**Name**
Displays the current function name, if known by the debugger. If no function is known, the address is displayed.

**Address**
Displays the current address.

**Suspend**
Increments the suspend count of the thread selected in the Thread List. A thread is suspended when its count goes from 0 to 1.

**Resume**
Decrements the suspend count of the suspended thread selected in the Thread List. A thread resumes when its count goes from 1 to 0.

**Set Focus**
Switches the focus to the thread selected in the Threads List.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Tool Arguments

Use this dialog box to specify the arguments you are prompted for when you run a custom tool.

**Arguments**
Specify additional arguments for the tool each time you start it. In addition to the arguments that are automatically used every time the tool starts, you can type arguments for each particular instance of the tool.

**Redirect to Output Window**
The output from the tool is displayed in an Output window. Each tool uses a separate Output window. You can switch among Output windows by selecting a tab at the base of the Output window.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Toolbars

Use this dialog box to display, change, or create toolbars.

**Toolbars**
Displays a list of toolbars you can choose to display.

**Toolbar Name**
Displays the name of the toolbar selected on the toolbars list. If this is a customized toolbar, you can edit the name by typing in this text box.

**New**
Displays the New Toolbar dialog box, which you use to name the toolbar you want to create.

**Customize**
Displays the Toolbars tab in the Customize dialog box, which you use to add toolbar buttons to existing toolbars.

**Reset**
Replaces all toolbars in the development environment with the standard toolbars. This option is available only when a standard toolbar is selected on the toolbar list.

**Delete**
Deletes a customized toolbar. This option is available only when a customized toolbar is selected on the toolbar list.

**Show ToolTips**
When this box is checked, Developer Studio displays the name of a tool when you place the cursor on the tool button.

**With Shortcut Keys**
When this box is checked, Developer Studio displays in the ToolTip the shortcut keys associated with the tool button.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

## Undo Check Out from Source Code Control

Use this dialog box to return checked out files to the control of the source-code control project without any local modifications.

**Files**
Lists the current source file or all the files that you have selected in the project window. If this list does not contain the desired files, click the files in the list to change the selection.

**Advanced**
Accesses additional options, such as adding comments to the source-code control project log. If your source-code control system does not support such options, this button is inactive.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Update All Dependencies

Use this dialog box to update the dependencies for one or more projects in your project workspace.

**Project configurations**
Select the project configurations for which you want to update dependencies. The Default Configuration is selected by default.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Win32 Network (TCP/IP) Settings

Use this dialog box to set up a connection for remote debugging of a Win32 application across a network using TCP/IP protocol.

**Remote Machine Name**
Type the identifier for the machine running the program you want to debug. The identifier can be the machine name or the machine's TCP/IP address. Under some circumstances (such as when debugging a program running on a Macintosh), TCP/IP name resolution may not be possible. In those cicumstances, you must use the TCP/IP address instead of the machine name.

A TCP/IP address is a series of four non-negative integers, separated by periods (for example, 125.12.0.1). Each integer can have up to three digits. Leading zeroes are not required. If you do not know the TCP/IP address for the machine you want to connect to, contact your network administrator.

**Note    Debug Monitor Password**
Type the debug password for the target machine running the program you want to debug. This password must match the password set in the Debug Monitor on the target machine.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Win32 Serial Settings

Use this dialog box to set the communication settings to use while debugging a Win32 application across a serial connection.

**Connector**
Specifies the communications port your modem uses.

**Baud Rate**
Specifies the transfer rate of your modem. Some modems can transmit at more than one baud rate, so check your hardware manual and select one that both systems can handle.

**Flow Control**
Indicates what Developer Studio should do if the communications buffer becomes too full to receive more data from the remote computer.

Select Hardware if the remote computer uses the hardware method. Select XON/XOFF if you don't know which flow-control method is used.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Add Bookmark

Use this dialog box to tag the current topic with a bookmark.

**Bookmark Name:**
The default bookmark name is the title of the topic. You can type in another name if you want to use a different name to identify the topic.

Once a topic is identified with a bookmark, you can quickly return to the topic by opening the InfoViewer Boomarks dialog (from the Edit menu, choose InfoViewer Bookmarks...) and double-clicking on the bookmark name. You can also browse among all topics identified by bookmarks by displaying the InfoViewer toolbar (from the View menu, choose Toolbars... and check InfoViewer) and pressing the following buttons:

{ewc msdncd, EWGraphic, dev148a 0 /a "MSDEV40.BMP"}

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Define Subset

Use this dialog box to define a new subset.

If you are interested in only a portion of the information available in the help system, you can define a "subset" that contains only the information you're interested in. There are several ways you can use subsets:

- You can view a subset in the Contents window, which allows you to browse only those topics in the subset.
- You can specify a subset to be searched when performing a query (full-text search).
- You can also specify a subset to be searched when you press F1 on a keyword.

You can select which subset you want for each of these uses by choosing Set Default Subsets from the Help menu. Note that these uses of subsets are all independent from one another.

**Subset Name:**
Displays the name of the subset whose contents are currently being edited. You can create a new subset by typing in a name, or you can edit an existing subset by selecting a name from the drop-down list.

**Available Books:**
Displays the nodes remaining outside the subset. You can drag and drop nodes between this list and the Books in Subset list. You can also select multiple nodes in this list.

**Add**
Adds the selected node to the subset. Note that when a node is added to the subset, it is removed from the Books to Choose From list.

**Add All**
Adds all nodes to the subset.

**Remove**
Removes the selected node from the subset. Note that when a node is removed from the subset, it reappears in the Books to Choose From list.

**Remove All**
Removes all nodes from the subset.

**Books in Subset:**
Displays the nodes currently contained within the subset. You can drag and drop nodes between this list and the Books To Choose From list. You can also select multiple nodes in this list.

**Include New Topics Only**
Includes in the subset only those topics that are flagged as being new since the last release.

**Save**
Saves the changes made to the current subset.

**Delete**
Deletes the current subset from the list of defined subsets.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Query Tab

Use the Query Tab of the Search dialog box to do a full-text search for a word or phrase. This will produce a list of all topics in the current information title that contain the specified word or phrase anywhere in their text.

**1. Type the words you want to find.**
Type in the word or phrase you want to search for. Phrases must be delimited by quotes; you can also use operators such as AND and OR. The drop-down list displays previous queries.

**2. Where would you like to search?**
**Entire Contents**
If selected, the entire Table of Contents for the current information title is searched.

**Current Topic Only**
If selected, only the current topic is searched. By default, occurrences of the word are highlighted in the text. Use CTRL+H to toggle highlighting.

**Last Topics Found**
If selected, only the topics found during the previous query are searched. You can select this to narrow your search in successive steps.

**Subset of Contents**
If selected, only the specified subset is searched. Choose a subset from the drop-down list. This lets you specify which portion of the documentation you're interested in searching. To define a new subset, press the Subset button.

**Subset...**
Press this button to open the Define Subset dialog, which lets you create a custom subset of the documentation.

**3. How thorough would you like the search?**
**Title and Text**
If selected, both the topic titles and the topic text are searched.

**Title Only**
If selected, only the topic titles are searched.

**4. Click the Query button to start the search.**
**Query**
Press this button to execute the query; that is, search for all occurences of the word or phrase you specified. The results of the search are displayed in the Query Results dialog box.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Index Tab

Use the Index Tab of the Search dialog box to look in the index for the current information title. This will produce a list of all topics that are indexed under the specified keyword.

**1. Type the first few letters of the word you're looking for.**
Type in the keyword you want to search for. The list box will scroll to the portion of the index corresponding to what you've typed.

**2. Click the Index entry you want, then choose List Books.**
The index entry that's the closest match to what you've typed is automatically highlighted. If that's the entry you're interested in, press the List Books button, or hit ENTER. If one of the adjacent entries in the index is what you're interested in, click on that entry and then press List Books.

You can also use the scroll bar to browse among the index listing, or use the TAB key to move focus into the list box and use the arrow keys to browse.

**List Books**
Press this button to list the topics indexed under the word or phrase highlighted in the index list.

**Click a topic you're interested in, then choose Display.**
This box displays a list of the topics indexed under the highlighted word or phrase. The list contains the name of each topic and of the book in which each topic is located. You can sort this list by either the topic name or the book name by clicking on the button at the top of either column. Click on the topic you want to see and then press the Display button.

**Display**
Press this button to display the topic selected in the topic list described above. The topic is displayed in a Help Topic window.

**Note**    If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# InfoViewer Bookmarks

Use the InfoViewer Bookmarks dialog to edit the list of bookmarks currently defined.

**Name:**
Use this control to add a new bookmark. The default bookmark name is the title of the help topic currently being viewed (or, if the InfoViewer Topic window is closed, the last one viewed). You can type in another name if you want to use a different name to identify the topic.

The list box below displays all the bookmarks currently defined, and the books in which those bookmarks are located. You can sort this list by the name of the bookmark or the name of the book by clicking on the button at the top of the appropriate column.

**Book:**
Displays the name of the book containing the topic currently being viewed, or the bookmark currently selected in the list box.

**Add**
Press this button to add a bookmark to the topic currently being viewed (or, if the InfoViewer Topic window is closed, the last one viewed), using the name specified in the Name: control.

**Delete**
Press this button to delete the bookmark currently selected in the list box.

**Display**
Press this button to display the topic identified by the bookmark currently highlighted in the list box. The topic is displayed in an InfoViewer Topic window.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Select Reference

Use the Select Reference dialog box to pick which help topic you want to view, choosing from the topics indexed under the word you requested help about.

If there are too many references to your keyword, you can select or define a subset that excludes certain portions of the documentation.

**Keyword:**
Displays the word your cursor was on when you hit the F1 key.

The list box below this displays all the topics which are indexed under the specified keyword, and the books in which those topics are located. You can sort this list by either the topic name or the book name by clicking on the button at the top of the appropriate column.

**Display references from subset:**
Displays the subset currently in use, that is, the selections of books (within the current information title) that were searched for information on the keyword you specified. You can choose another subset from the drop-down list, or press the Subset button to define your own subset. When you choose another subset, only the topics in that subset matching the keyword are displayed in the list box.

**Titles which have index entries for: <keyword>**
Displays the information titles that have the specified keyword in their indexes.

**Display**
Press this button to display the topic highlighted in the list box. The topic is displayed in an InfoViewer Topic window.

**Subset...**
Press this button to open the Define Subset dialog, which lets you create a custom subset of the documentation.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Keyword files not found

The index used by InfoViewer is stored in a .KWD file. This file wasn't found in the expected location.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Print

Use the Print dialog box to print out a help topic, a group of help topics, or the Table of Contents listing for help.

To print an individual help topic, you can also click the right mouse button and choose the Print Topic menu item, or you can hit CTRL+P from within a topic window.

**Printer**
Displays the printer to be used for printing.

**Copies**
Specifies the number of copies to be printed.

**Print Range**
**Current Topic**
If selected, only the current topic will be printed.

**Selected Topics or Books**
Print all the topics that are selected in the Table of Contents; if one or more books is selected, all the topics in those books are printed.

You can select a continuous range of topics by holding down the SHIFT key and using the cursor keys. You can select non-adjacent topics by holding down the CTRL key and either using the mouse or using the cursor keys and space bar.

**Content Listing**
Print the Table of Contents as it is currently displayed.

**Print to File**
Check this box to print the selection to a file.

**Print Annotations**
Check this box to include any annotations that you've added to the topic(s).

**Setup**
Press this button to open the Printer Setup dialog box.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Microsoft Developer Help

You are viewing an environment-help topic window, used for displaying context-sensitive help about the user interface of the Microsoft Developer Studio.

Help about the user interface is often invoked from within a dialog box (either by hitting F1 or pressing the Help button); however, because most dialog boxes are modal, it is impossible to open an ordinary MDI child window until the dialog box is dismissed. As a result, help about the user interface cannot be presented in normal Help Topic windows (which are MDI child windows). Instead, a window such as this is used, which floats above all the other windows within Developer Studio.

Because of this constraint, this window lacks much of the functionality that normal Help Topic windows have, including the Previous Topic, Next Topic, and Sync with Contents toolbar buttons. You can move to other topics by clicking on a jump or using the See Also button; you can return to a previously viewed topic by pressing the Go Back button.

If you want the navigation features found in a normal Help Topic window, you must dismiss the modal dialog box from which you invoked help. Then open a normal Topic window by pulling down the View menu and choosing Help Topic. The keyboard shortcut is ALT+1.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Query Results/History List

This window acts as both the Query Results window and the History List window.

**Query Results Tab**
Use the Query Results Tab to view the results of a query (full-text search) made from the Query tab in the Help Search dialog.

The list box displays all the help topics which contain the word or phrase that was searched for.The list box displays the titles of the topics and the books in which those topics are located, in order from most occurrences of the target word to fewest occurrences. You can sort this list by order, book, or topic by clicking on the button at the top of the appropriate column.

The query that was executed and the scope in which it was executed are displayed at the top of the window. You can also use the following buttons:

{ewc msdncd, EWGraphic, dev157a 0 /a "MSDEV40.BMP"}

Press this button to display the previous topic in the results list in a topic window.

{ewc msdncd, EWGraphic, dev157a 1 /a "MSDEV40.BMP"}

Press this button to display the next topic in the results list in a topic window.

{ewc msdncd, EWGraphic, dev157a 2 /a "MSDEV40.BMP"}

Press this button to return to the Search dialog box.

**History List Tab**

The list box displays the last 50 help topics you have viewed. The list box displays the order in which the topics were viewed (starting with the previous topic), the titles of the topics, and the books in which those topics are located. You can sort this list by order, book, or topic by clicking on the button at the top of the appropriate column.

**Note**   If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Set Default Subsets

Use the Set Default Subsets dialog to select which subset of the documentation you would like to use for various InfoViewer operations.

A subset is selection of the books available within an information title. There are predefined subsets which you can choose from; the Entire Contents set contains everything in an information title. You can also define your own custom subsets. For example, you could define a subset that excludes the samples but includes the rest of the documentation.

There are three different ways a subset can be used:

**InfoView Subset**
This subset is used for the InfoView pane of the Project Workspace window. As long as a subset is in effect for the InfoView pane, only the books and topics in that subset are displayed in the Table of Contents.

To see the name of the subset currently displayed in the Table of Contents, display the InfoViewer Contents toolbar; from the View menu, choose Toolbars... and check the InfoViewer Contents checkbox.

**Query Subset**
This subset is used to narrow a full-text search. Only the books within that subset will be searched during the query. You can also select a subset for a query from the Query Tab of the Search dialog box.

**Context-Sensitive Help (F1) Subset**
This subset is used to narrow the scope of F1 help. Only the books within that subset will be used in response to hitting F1 on a keyword.

To define a new subset, choose Define Subset from the Help menu, or click the right mouse button in the Table of Contents and choose Define Subset... from the pop-up menu. You can also reach the Define Subset dialog by pressing the Subset button on the Query Tab of the Search dialog box, or the Subset button on the Select Reference dialog box.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Open Information Title

Use the Open Information Title dialog to select which information title (or help file) you would like InfoViewer to use. Once you have selected an information title, that title will be used for the Table of Contents displayed in the InfoView pane of the Project Workspace window, for finding information in response to hitting F1 on a keyword, and for both index lookups and full-text searches from the Search dialog.

You can see which information title is currently open by displaying the InfoView toolbar; from the View menu, choose Toolbars... and check the InfoView checkbox.

You can also open an information title by specifying the name of its .MVB file in the File Open dialog.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Windows

Use this dialog box to manage open files in Microsoft Developer Studio.

**Select Window**
Lists all of the currently open windows. You can Activate a window by either double-clicking your selection in the Select Window list box or by selecting a window and then choosing the Activate button. If you select a single window, you can choose Activate, Save, or Close Window. If you select multiple windows, you can choose Save or Close Window.

**Activate**
Activates the currently selected file in the Windows list.

**Cancel**
Closes this dialog box.

**Save**
Saves the currently selected files in the Windows list.

**Close Window**
Closes the window and file of each file currently selected in the Windows list.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Find

Use this text box to take actions on text. Key commands that you can apply to text in other contexts, such as keyword searches, searches in text editors, and going to references or definitions also take effect on the text in this box.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Web Favorites

Use this dialog box to visit your favorite World Wide Web sites from Microsoft Developer Studio. To choose a Web site, select the site's name in the Name box. The Universal Resource Locator (URL) for the selected site appears beneath the Name box. To visit a site, double-click on the site name or select the Name and choose the Go To button.

If you attempt to access a Web site and InfoViewer cannot find a Web browser, a dialog box appears asking if you would like to specify a Web browser as an InfoViewer command. If you choose Yes, the Specify InfoViewer Command dialog box appears. Be sure to test the command using the Test button on the Specify InfoViewer Command dialog box. If you accept a command without testing it, and the command is in error, the command cannot be removed without editing the Registry.

**Name**
This box contains the names of favorite Web sites. Visual C++ ships with this box preloaded with a list of Web sites that contain information of interest to Visual C++ developers. You can change the contents of this list using the New, Edit, and Delete buttons.

**New**
Opens the Web Favorite Details dialog, where you can add a new Web site.

**Edit**
Opens the Web Favorite Details dialog, where you can edit the name and URL of the Web site currently selected.

**Delete**
Deletes the name and URL of the Web site currently selected.

**Go To**
Connects you to the Web site currently selected.

**Close**
Closes the window.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Web Favorite Details

Use this dialog box to enter a World Wide Web site to be displayed in the list of sites in the <u>Web Favorites</u> dialog box. If you accessed this dialog box by selecting the Edit button, the new site will replace the site selected in the Web Favorites dialog bix. If you accessed this dialog box by selecting the New button, it will added to the list of Web sites.

**Friendly Name**
Your name for the Web site.

**Location**
The Universal Resource Locator (URL) for the web site.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Specify InfoViewer Command

Use this dialog box to register a World Wide Web browser as an InfoViewer command. This dialog box appears when you try to access a Web site using Web Favorites and InfoViewer is unable to launch a Web browser. This occurs most often under Windows NT 3.51 or under Windows 95 when a browser is not properly specified and installed.

To specify a Web browser, type the full path name of the browser, followed by a space followed by %s, in the Command box. When you select a Web site in the Web Favorites dialog box, Developer Studio replaces %s with the Universal Resource Locator for the site and executes the command.

The initial contents of the Command box show the bad command that caused this dialog box to appear. Do not accept this command by choosing OK until you have edited the command string and verified it by using the Test button. If you accept a command without testing it, and the command is in error, the command cannot be removed without editing the Registry.

**Command**
Contains the command to launch a Web browser, consisting of the browser path name followed by a space, followed by %s. You can also specify a command for ftp or WAIS.

**Test**
Executes the command specified in the Command box. Always use the Test button before accepting before choosing OK to accept a command.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Work Offline

Use this dialog box to specify how closely your working copy of the currently-selected data source(s) matches the actual data source(s). This dialog box is useful if you carry your development environment on a portable hard disk or if you know that a server will be down for maintenance.

**Disconnect only**
Disconnects from the currently-selected data sources; no refresh occurs.

**Disconnect with Refresh**
Disconnects from the selected data source(s) but refreshs the names of all stored procedures, triggers, tables, views, columns, and user-defined types.

If you choose this option, you can choose the associated check box to also update the textual content of the stored procedures and triggers.

The refresh time varies with the size of the data source.

**Note**    If your local copy and the server copy of a stored procedure or a trigger have both changed, your local copy is renamed. The renaming algorithm merely appends an integer to the local procedure's name.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# About Property Pages

The property page is one of the most important features of Developer Studio. Property pages allow you to control the appearance and behavior of resources you create. It also allows you to set some source code options and display information concerning the sources and targets of your project.

**Note**    Whenever you make a change on a property page, it is made immediately. You cannot cancel any changes made on a property page.

You can use any of the editing keyboard shortcut keys to cut, copy, and paste text. In general these shortcut keys can be used in any edit control on the property page.

You can control the behavior of the Properties window to suit your working style or the nature of the resource editing task. Use the "pushpin" command button in the upper-left corner of the Properties window:

| | Button Position | Result |
|---|---|---|
| | {ew c ms dnc d, EW Grap hic, dev 0c 0 / a "M SD EV 40. BM P"} | When the button is in the down position, the Properties window stays visible even when you are working in another window. This is convenient if, during an editing session, you want to move back and forth frequently between setting properties and editing objects. Pressing ENTER after you change a value in the Properties window returns you to the editing window but leaves the Properties window |

visible.

{ewc msdncd, EWGraphic, dev0c1 / a "MSDEV40.BMP"} When the button is in the up position, you can dismiss the active Properties window by pressing ENTER or ESC. This is useful if you want to concentrate on working in an editing window but need to bring up the Properties window briefly to change one or two values.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Accelerator Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Modifiers**
Indicates whether the accelerator is a combination formed with CTRL, ALT, or SHIFT. When the key is an ASCII value, Ctrl and Shift are not available. Type: Bool. Defaults: Ctrl is True, Alt and Shift are False.

**Key**
The accelerator key. It can be one of the following:

- Integer: Range 0 to 255. It is interpreted as an ASCII or virtual-key value, depending on the Type property. A single digit is interpreted as a key value. To enter an ASCII value from 0 to 9, precede the number with two zeros (for example, 006).
- Character: Single character optionally preceded by ^ to signify a control character.
- Virtual key identifier: Any one of the virtual-key identifiers in the drop-down list.

**Type**
Specifies whether the Key property is an ASCII value or a virtual key (VirtKey) value.

**Next Key Typed**
When you choose this command, the next key combination typed changes the Key and Modifiers values appropriately. The key is always interpreted as a virtual key if possible.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Accelerator Properties: Resource

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Language**
This drop-down list box contains the language to be used for this resource.

**Condition**
Determines the inclusion of the resource. For example, if the condition is _DEBUG, this resource would only be included in debug builds.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Animate Properties: Extended Styles

**Client Edge**
Creates a border with a sunken edge around the animate control. Type: Bool. Default: False.

**Static Edge**
Creates a border around the animate control. Type: Bool. Default: False.

**Modal Frame**
Creates a control that has a double border. Type: Bool. Default: False.

**Transparent**
Specifies that a control created with this style is to be transparent. Any controls that are beneath this control are not obscured by this control. A control with this style receives WM_PAINT messages only after all sibling controls beneath it have been updated. Type: Bool. Default: False.

**Accept Files**
A control with this style accepts drag-drop files. If a user drops a file on this control, WM_DROPFILES messages will be sent to the control.Type: Bool. Default: False.

**No Parent Notify**
Specifies that the child window does not send the WM_PARENTNOTIFY message to its parent window. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Animate Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: False.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: True.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.

**Center**
Centers the animation in the animation control's window. Type: Bool. Default: False.

**Transparent**
Draws the animation using a transparent background rather than the background color specified in the animation clip. Type: Bool. Default: False.

**Autoplay**
Starts playing the animation as soon as the animation clip is opened. Type: Bool. Default: False.

**Border**
Creates a border around the animate control. Type: Bool. Default: True.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Bitmap Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Width**
Image width in pixels. Type: Integer. Default: 48.

**Height**
Image height in pixels. Type: Integer. Default: 48.

**Colors**
Monochrome (2), 16, or 256. The number of colors in a bitmap is determined by the current display device.

**File Name**
The name of the file containing the bitmap resources.

**Save Compressed**
Saves the image in compressed format to save space. This option is useful for large bitmaps. Only color bitmaps can be compressed. Type: Bool. Default: False.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

## Bitmap/Icon Properties: Palette

Shows available palette colors. If you have selected 16-color images, you have sixteen choices; if you have selected 256-color images, you have 256 choices.

Click a color on the Palette tab to set the color indicator on the Colors palette of the graphics editor to the selected color.

Double-click a color to create a custom color using the Custom Color Selector dialog box and set the color indicator to that custom color.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Bitmap Properties: Resource

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Language**
This drop-down list box contains the language to be used for this resource.

**Condition**
Determines the inclusion of the resource. For example, if the condition is _DEBUG, this resource would only be included in debug builds.

**File Name**
The name of the file containing the resource.

**Preview**
A box showing what the bitmap looks like. It is useful for browsing through graphics resources without opening them.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Block Header Properties: General

**Language ID**
This drop-down list box contains the language to be used for this project.

**Code Page**
This drop-down list box contains the code page to be used for this project.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Check Box Properties: Extended Styles

**Client Edge**
Creates a border with a sunken edge around the check box control. Type: Bool. Default: False.

**Static Edge**
Creates a border around the check box control. Type: Bool. Default: False.

**Modal Frame**
Creates a control that has a double border. Type: Bool. Default: False.

**Transparent**
Specifies that a control created with this style is to be transparent. Any controls that are beneath this control are not obscured by this control. A control with this style receives WM_PAINT messages only after all sibling controls beneath it have been updated. Type: Bool. Default: False.

**Accept Files**
A control with this style accepts drag-drop files. If a user drops a file on this control, WM_DROPFILES messages will be sent to the control.Type: Bool. Default: False.

**Right-to-Left Reading Order**
The control text is displayed using Right to Left reading order properties. Type: Bool. Default: False.

**No Parent Notify**
Specifies that the child window does not send the WM_PARENTNOTIFY message to its parent window. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Check Box Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Caption**
The text that appears as part of the control to label it. To make one of the letters in the caption of a control the mnemonic key, precede it with an ampersand (&). Default: A name based on the type of control (for example, "Check") plus a number based on the resource identifier assigned by Visual C++.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: False.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: True.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Check Box Properties: Styles

**Auto**
Creates a check box that, when selected, automatically toggles between checked and unchecked states. You must set this property to True if you are using a group of check boxes with Dialog Data Exchange. Type: Bool. Default: True.

**Tri-State**
Creates a three-state check box. A three-state check box can be grayed as well as checked or not checked. A grayed check box indicates that the state represented by the control is undetermined. Type: Bool. Default: False.

**Left Text**
Positions the check box's caption text to the left instead of to the right. Type: Bool. Default: False.

**Pushlike**
Makes a button (such as a check box, three-state check box, or radio button) look and act like a push button. The button looks raised when it isn't pushed or checked, and sunken when it is pushed or checked. Type: Bool. Default: False.

**Multiline**
Wraps the button text to multiple lines if the text string is too long to fit on a single line in the button rectangle.Type: Bool. Default: False.

**Notify**
Notifies the parent window if a check box has been clicked or double-clicked. Type: Bool. Default: True.

**Flat**
Makes a button look flat, not three-dimensional. Type: Bool. Default: False.

**Icon**
Specifies that the button displays an icon.Type: Bool. Default: False.

**Bitmap**
Specifies that the button displays a bitmap. Type: Bool. Default: False.

**Horizontal Alignment**
Positions the control's caption text to the left, center, right or default position in the control.

**Vertical Alignment**
Positions the control's caption text to the top, bottom, center or default position in the control.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Class Properties: General

**Name**
Identifies the class.

**Base Class**
Identifies the base class from which this class is derived.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Column Properties

**Name**
The name of the column and whether the column can contain **NULL** values.

**Type**
The data type of the data in the column.

For information on the Precision, Length, Scale, Radix, and Data Type fields, see the topic, "Precision, Scale, Length, and Display Size" in the ODBC SDK documentation found in Books Online. You can use the Search command from the Help menu to quickly find this topic.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Combo Box Properties: Extended Styles

**Client Edge**
Creates a border with a sunken edge around the combo box control. Type: Bool. Default: False.

**Static Edge**
Creates a border around the combo box control. Type: Bool. Default: False.

**Modal Frame**
Creates a control that has a double border. Type: Bool. Default: False.

**Transparent**
Specifies that a control created with this style is to be transparent. Any controls that are beneath this control are not obscured by this control. A control with this style receives WM_PAINT messages only after all sibling controls beneath it have been updated. Type: Bool. Default: False.

**Accept Files**
A control with this style accepts drag-drop files. If a user drops a file on this control, WM_DROPFILES messages will be sent to the control.Type: Bool. Default: False.

**Left Scrollbar**
Vertical scroll bar (if present) is to the left of the client area. Type: Bool. Default: False.

**Right-to-Left Reading Order**
The control text is displayed using Right to Left reading order properties. Type: Bool. Default: False.

**Right Aligned Text**
Specifies that text is right-aligned within the control. Type: Bool. Default: False.

**No Parent Notify**
Specifies that the child window does not send the WM_PARENTNOTIFY message to its parent window. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Combo Box Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: False.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: True.

**Enter list choices**
Contains the choices you want to appear in the combo box when the dialog box is created. Press CTRL+ENTER at the end of each item in the list to move to the next line. This property is only available in resource files with Microsoft Foundation Class Library support.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Combo Box Properties: Styles

**Type**
Specifies the type of combo box. This property can have one of the following values:

- Simple: Creates a simple combo box that combines an edit-box control which takes user input with a list control. The list is displayed at all times, and the current selection in the list is displayed in the edit-box control.

- Dropdown (Default): Creates a drop-down combo box. This type is the same as a simple combo box, except the list is not displayed unless the user clicks a drop-down arrow at the right of the edit-box control portion of the combo box.

- Drop List: This type is the similar to the drop-down style, but the edit-box control is replaced by a static-text item which does not take user input that displays the current selection in the list.

**Owner Draw**
Controls the owner-draw characteristics of the combo box. This property can be one of the following values:

- No (default): Turns off the owner-draw style. The combo box contains strings.

- Fixed: Specifies that the owner of the combo box is responsible for drawing its contents and that the items in the combo box are the same height.

  **CWnd::OnMeasureItem** is called when the combo box is created and **CWnd::OnDrawItem** is called when a visual aspect of the combo box has changed.

- Variable: Specifies that the owner of the combo box is responsible for drawing its contents and that the items in the combo box are variable in height.

  **CWnd::OnMeasureItem** is called for each item in the list when the combo box is created and **CWnd::OnDrawItem** is called when a visual aspect of the combo box has changed.

**Has Strings**
Specifies that an owner-draw combo box contains items consisting of strings. The combo box maintains the memory and pointers for the strings so the application can use the **LB_GETTEXT** message to retrieve the text for a particular item. By default, all combo boxes except owner-draw combo boxes have this style. An application can create an owner-draw combo box either with or without this style.

This style is only available if the Owner Draw property is set to either Fixed or Variable. If Owner Draw is set to No, the combo box contains strings by default. Type: Bool. Default: False.

**Sort**
Sorts the contents of the combo box alphabetically. Type: Bool. Default: True.

**Vertical Scroll**
Creates a combo box with a vertical scroll bar. Type: Bool. Default: True.

**No Integral Height**
Specifies that the size of the combo box is exactly the size specified by the application when it creates the combo box. Normally, Windows sizes a combo box so that the combo box does not display partial items. Type: Bool. Default: False.

**Auto HScroll**
Automatically scrolls text to the right when the user types a character at the end of the line. Type: Bool. Default: False.

**Disable No Scroll**
Shows a disabled vertical scroll bar in the combo box when the box does not contain enough items to scroll. Without this style, the scroll bar is hidden when the combo box does not contain enough items to scroll. Type: Bool. Default: False.

**OEM Convert**
Converts text entered in the combo-box control from the Windows character set to the OEM character set and then back to the Windows set. This ensures proper character conversion when the application calls the **AnsiToOem** function to convert a Windows string in the edit-box control to OEM characters. This style is most useful for combo-box controls that contain filenames. Type: Bool. Default: False.

**Uppercase**
Converts all text to uppercase in both the selection field and the list.Type: Bool. Default: False.

**Lowercase**
Converts all text to lowercase in both the selection field and the list. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Cursor Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Width**
Image width in pixels. This property is determined by the target-device definition selected by the user. (This property is displayed only and cannot be modified on the property page).

**Height**
Image height in pixels. This property is determined by the target-device definition selected by the user. (This property is displayed only and cannot be modified on the property page).

**Hot Spot**
Location of the cursor's active area. This property is specified in pixels, relative to the upper-left corner (0,0). It is set with the Set Hot Spot button on the toolbar of the cursor editor. (This property is displayed only and cannot be modified on the property page).

**File Name**
The name of the file containing the cursor resource.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Cursor Properties: Resource

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Language**
This drop-down list box contains the language to be used for this resource.

**Condition**
Determines the inclusion of the resource. For example, if the condition is _DEBUG, this resource would only be included in debug builds.

**File Name**
The name of the file containing the resource.

**Preview**
A box showing what the bitmap looks like. It is useful for browsing through graphics resources without opening them.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Custom Control Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Caption**
The text that appears as part of the control to label it. To make one of the letters in the caption of a control the mnemonic key, precede it with an ampersand (&). Default: A name based on the type of control (for example, "Check") plus a number based on the resource identifier.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: False.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: True.

**Class**
The name of the control's Windows class. This class must be registered before the dialog box containing the control is created.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.

**Style**
A 32-bit hexadecimal value specifying the control's style, primarily used to edit the lower 16 bits that make up a user control's sub-style.

**ExStyle**
A 32-bit hexadecimal value specifying the control's extended style.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Custom Resource Properties: Resource

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Language**
This drop-down list box contains the language to be used for this resource.

**Condition**
Determines the inclusion of the resource. For example, if the condition is _DEBUG, this resource would only be included in debug builds.

**File Name**
The name of the file containing the resource (if the resource is external).

**External File**
If checked, the resource contents are stored in an external file listed in File Name. Type: Bool. Default: True.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Data Properties: General

**Name**
The member variable name.

**Type**
The member variable type.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Database Properties

**Driver Name**
The filename of the ODBC driver.

**Data Source**
The name of the data source as it was entered in the ODBC Manager's ODBC SQL Server Setup dialog box.

**Server**
The name of the server on which the database resides. This is the name that was entered in the ODBC Manager's ODBC SQL Server Setup dialog box.

**Database**
The name of the database. This is the name that was entered in the ODBC Manager's ODBC SQL Server Setup dialog box.

**User**
The user name as reported by the ODBC driver.

**Logon**
The logon name as reported by the ODBC driver.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Dialog Properties: Extended Styles

**Tool Window**
Creates a tool window; that is, a window intended to be used as a floating toolbar. A tool window has a title bar that is shorter than a normal title bar, and the window title is drawn using a smaller font. Type: Bool. Default: False.

**Client Edge**
Creates a border with a sunken edge around the dialog box. Type: Bool. Default: False.

**Static Edge**
Creates a border around the dialog box. Type: Bool. Default: False.

**Transparent**
A window using this style is to be transparent. Any windows that are beneath this window are not obscured by this window. A window with this style receives WM_PAINT messages only after all sibling windows beneath it have been updated. Type: Bool. Default: False.

**Accept Files**
A dialog box with this style accepts drag-drop files. If a user drops a file on this dialog box, WM_DROPFILES messages will be sent to the control.Type: Bool. Default: False.

**Control Parent**
Allows the user to navigate among the child windows of the dialog by using the TAB key.

**Context Help**
Includes a question mark in the title bar of the window. When the user clicks the question mark, the cursor changes to a question mark with a pointer. If the user then clicks a child window, the child receives a WM_HELP message. The child window should pass the message to the parent window procedure, which should call the WinHelp function using the HELP_WM_HELP command. The Help application displays a pop-up window that typically contains help for the child window.

**No Parent Notify**
Specifies that the child window does not send the WM_PARENTNOTIFY message to its parent window. Type: Bool. Default: False.

**Right-to-Left Reading Order**
The dialog box text is displayed using Right to Left reading order properties. Type: Bool. Default: False.

**Right Aligned Text**
Specifies that text is right-aligned within the dialog box. Type: Bool. Default: False.

**Left Scrollbar**
Vertical scroll bar (if present) is to the left of the client area. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Dialog Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Caption**
The text that appears as part of the dialog box to label it. Default: A name based on the type of resource (in this case, "Dialog") plus a number based on the resource identifier assigned by Visual C++.

**Font Name**
The typeface of the font that will be used in all the controls in the dialog box. The bold version of the typeface is always used. Change this value by choosing the Font command in the lower-left corner of the property page. Default: MS Sans Serif.

**Menu**
Contains the resource identifier of the menu used in the dialog box, if any. Type: Resource identifier. Default: None.

**Font Size**
The point size of the font that will be used in all the controls in the dialog box. Default: 8 points. Change this value by choosing the Font command in the lower-left corner of the property page.

**Font**
Choose the Font command to change the typeface or size of the dialog-box font.

**X Pos**
The x-coordinate, in dialog box units (DLUs), of the upper-left corner of the dialog box. Type: Integer.

**Y Pos**
The y-coordinate, in DLUs, of the upper-left corner of the dialog box. Type: Integer.

**Class Name**
Identifier of a registered dialog class (a Windows operating system window class, not to be confused with a C++ class). This identifier is provided to support C programming. If you are using a resource file with Microsoft Foundation Class Library support, this option is disabled. Type: Integer or String. String must be in double quotes. Default: None.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Dialog Properties: More Styles

**System Modal**
Creates a system-modal dialog box, which prohibits switching to another window or program while the dialog box is active. Type: Bool. Default: False.

**Absolute Align**
Determines whether the dialog box is aligned relative to the screen or relative to its parent window. If Absolute Align is True, the dialog is displayed at coordinates relative to the upper-left corner of the screen. Type: Bool. Default: False.

**Visible**
Specifies that the dialog box is visible when first displayed. Set this property to False for form views and dialog-bar template resources. Type: Bool. Default: True.

**Disabled**
Creates a dialog box that is initially disabled. Type: Bool. Default: False.

**Set Foreground**
Brings the dialog box to the foreground. Internally, Windows calls the SetForegroundWindow function for the dialog box. Type: Bool. Default: False.

**3D-look**
Gives the dialog box a nonbold font and draws three-dimensional borders around control windows in the dialog box. Type: Bool. Default: False.

**No Fail Create**
Creates the dialog box even if errors occur — for example, if a child window cannot be created or if the system cannot create a special data segment for an edit control. Type: Bool. Default: False.

**No Idle Message**
Suppresses the **WM_ENTERIDLE** message ordinarily sent to a dialog box's owner when no more messages are waiting in its message queue. Type: Bool. Default: False.

**Control**
Creates a dialog box that works well as a child window of another dialog box, much like a page in a property sheet. This style allows the user to tab among the control windows of a child dialog box, use its accelerator keys, and so on. Type: Bool. Default: False.

**Center**
Centers the dialog box in the working area — that is, the area not obscured by the tray. Type: Bool. Default: False.

**Center Mouse**
Centers the mouse cursor in the dialog box. Type: Bool. Default: False.

**Local Edit**
Specifies that edit-box controls in the dialog box will use memory in the application's data segment. Normally, all edit-box controls in dialog boxes use memory outside the application's data segment. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Dialog Properties: Resource

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Language**
This drop-down list box contains the language to be used for this resource.

**Condition**
Determines the inclusion of the resource. For example, if the condition is _DEBUG, this resource would only be included in debug builds.

**Preview**
A box showing what the bitmap looks like. It is useful for browsing through graphics resources without opening them.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Dialog Properties: Styles

**Style**
One of the following:

- Overlapped: Creates an overlapped window. An overlapped window is always a top-level window and should have a caption and a border.
- Popup (Default): Creates a pop-up window.
- Child: Creates a child window.

**Border**
One of the following:

- None: No border. A title bar is not available.
- Thin : A thin border.
- Resizing: Creates a thick border that can be used to resize the dialog box.
- Dialog Frame (Default): A dialog-box border.

**Titlebar**
Creates a title bar for the dialog box. This check box is cleared if the dialog box has no border. Type: Bool. Default: True.

**System Menu**
Creates a system menu for the dialog box. This check box is disabled if there is no title bar. Type: Bool. Default: True.

**Minimize Box**
Creates a minimize box for the dialog box. This check box is disabled if there is no title bar. Type: Bool. Default: False.

**Maximize Box**
Creates a maximize box for the dialog box. This check box is disabled if there is no title bar. Type: Bool. Default: False.

**Clip Siblings**
Clips child windows relative to each other; that is, when a particular child window is repainted, this style clips all other top-level child windows out of the region of the child window to be updated. If Clip Siblings is False and child windows overlap, it is possible, when drawing in the client area of a child window, to draw in the client area of a neighboring child window. Clip Siblings is for use with child windows only. Type: Bool. Default: False.

**Clip Children**
Excludes the area occupied by child windows when drawing within the parent window. This option is used when creating the parent window. Do not use this style if your dialog box contains a group box. Type: Bool. Default: False.

**Horizontal Scroll**
Creates a horizontal scroll bar for the dialog box. Type: Bool. Default: False.

**Vertical Scroll**
Creates a vertical scroll bar for the dialog box.

If you create horizontal or vertical scroll bars for a dialog box that uses the default border style (the Dialog Frame style), the scroll bars are drawn overlapping the borders of the dialog rather than within them, and the contents of the dialog box are clipped improperly. This is standard Windows behavior. To avoid this behavior, use a different border style when creating a scrollable dialog box. Type: Bool. Default: False.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Edit Properties: Extended Styles

**Client Edge**
Creates a border with a sunken edge around the edit control. Type: Bool. Default: False.

**Static Edge**
Creates a border around the edit control. Type: Bool. Default: False.

**Modal Frame**
Creates a control that has a double border. Type: Bool. Default: False.

**Transparent**
Specifies that a control created with this style is to be transparent. Any controls that are beneath this control are not obscured by this control. A control with this style receives WM_PAINT messages only after all sibling controls beneath it have been updated. Type: Bool. Default: False.

**Accept Files**
A control with this style accepts drag-drop files. If a user drops a file on this control, WM_DROPFILES messages will be sent to the control.Type: Bool. Default: False.

**Left Scrollbar**
Vertical scroll bar (if present) is to the left of the client area. Type: Bool. Default: False.

**Right-to-Left Reading Order**
The control text is displayed using Right to Left reading order properties. Type: Bool. Default: False.

**Right Aligned Text**
Specifies that text is right-aligned within the control. Type: Bool. Default: False.

**No Parent Notify**
Specifies that the child window does not send the WM_PARENTNOTIFY message to its parent window. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Edit Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: False.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: True.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Edit Properties: Styles

**Align Text**
Text aligns left, centered, or right when Multiline is selected. Default: Left.

**Multiline**
Creates a multiline edit-box control.

When the multiline edit-box control is in a dialog box, the default result of pressing the ENTER key is to choose the default button.

If AutoHScroll is selected, the multiline edit-box control automatically scrolls horizontally when the caret goes past the right edge of the control. To start a new line, the user must press ENTER. If AutoHScroll is not selected, the control automatically wraps words to the beginning of the next line when necessary. A new line is also started if the user presses ENTER, providing the Want Return property is set. The position of the wordwrap is determined by the window size. If the window size changes, the wordwrap position changes and the text is redisplayed.

Multiline edit-box controls can have scroll bars. An edit-box control with scroll bars processes its own scroll-bar messages. Edit-box controls without scroll bars scroll as described in the previous paragraph. They also process any scroll messages sent by the parent window. Type Bool. Default: False.

**Horizontal Scroll**
Provides a horizontal scroll bar for a multiline control. Type: Bool. Default: False.

**Auto HScroll**
Automatically scrolls text to the right when the user types a character at the right end of the box. Type: Bool. Default: True.

**Vertical Scroll**
Provides a vertical scroll bar for a multiline edit-box control. Type: Bool. Default: False.

**Auto VScroll**
In a multiline control, Auto VScroll automatically scrolls text up one line when the user presses ENTER on the last line. Type: Bool. Default: False.

**Password**
Displays all characters as an asterisk (*) as they are typed into the edit-box control. This property is not available in multiline controls. Type: Bool. Default: False.

**No Hide Selection**
Changes the way text is displayed when an edit box loses and regains focus. If NoHideSel is set to True, selected text in an edit box is displayed as selected at all times. Type: Bool. Default: False.

**OEM Convert**
Converts text typed in the edit-box control from the Windows character set to the OEM character set and then back to the Windows set. This ensures proper character conversion when the application calls the **AnsiToOem** function to convert a Windows string in the edit-box control to OEM characters. This style is most useful for edit-box controls that contain filenames. Type: Bool. Default: False.

**Want Return**
Specifies that a carriage return be inserted when the user presses the ENTER key while typing text into a multiline edit-box control in a dialog box. If this style is not specified, pressing the ENTER key has the same effect as pressing the dialog box's default push button. This style has no effect on a single-line edit-box control. Type: Bool. Default: False

**Border**
Creates a border around the edit box. Type: Bool. Default: True.

**Uppercase**

Converts all characters to uppercase as they are typed into the edit box. Type: Bool. Default: False.

**Lowercase**
Converts all characters to lowercase as they are typed into the edit box. Type: Bool. Default: False.

**Read-Only**
Prevents the user from typing or editing text in the edit box. Type: Bool. Default: False.

**Number**
Prevents the user from typing non-numeric characters. Type: Bool. Default: False.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Function Properties: General

**Name**
The function name.

**Return Type**
Identifies the type of value that the function returns.

**Parameters**
Lists the types of parameters the function takes in declaration order.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Group Properties: General

Use the Group Properties General property page to set the name for a group of files in a project.

**Group Name**
The name of the group.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Group Box Properties: Extended Styles

**Client Edge**
Creates a border with a sunken edge around the group box control. Type: Bool. Default: False.

**Static Edge**
Creates a border around the group box control. Type: Bool. Default: False.

**Modal Frame**
Creates a control that has a double border. Type: Bool. Default: False.

**Transparent**
Specifies that a control created with this style is to be transparent. Any controls that are beneath this control are not obscured by this control. A control with this style receives WM_PAINT messages only after all sibling controls beneath it have been updated. Type: Bool. Default: False.

**Accept Files**
A control with this style accepts drag-drop files. If a user drops a file on this control, WM_DROPFILES messages will be sent to the control.Type: Bool. Default: False.

**Right-to-Left Reading Order**
The control text is displayed using Right to Left reading order properties. Type: Bool. Default: False.

**No Parent Notify**
Specifies that the child window does not send the WM_PARENTNOTIFY message to its parent window. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Group Box Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Caption**
The text that appears as part of the control to label it. To make one of the letters in the caption of a control the mnemonic key, precede it with an ampersand (&). Default: A name based on the type of control (for example, "Check") plus a number based on the resource identifier.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: False.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: False.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Group Box Properties: Styles

**Horizontal Alignment**
Positions the control's caption text to the left, center, right or default position in the control.

**Icon**
Specifies that the button displays an icon.Type: Bool. Default: False.

**Bitmap**
Specifies that the button displays a bitmap. Type: Bool. Default: False.

**Notify**
Notifies the parent window if a group box has been clicked or double-clicked. Type: Bool. Default: False.

**Flat**
Makes a button look flat, not three-dimensional. Type: Bool. Default: False.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Header File Properties: General

This is a read-only property page.

**Filename**
Fully qualified name of the file.

**Last modified date**
Date on which the file was last modified.

**Status**
Shows the source-code control status, if the file is under source-code control in a source-code control system that conforms to the Microsoft Source Code Control API Specification.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# InfoView Properties: InfoViewer Topic

**Name**
The title of the book or topic currently selected.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Hot Key Properties: Extended Styles

**Client Edge**
Creates a border with a sunken edge around the hot key control. Type: Bool. Default: False.

**Static Edge**
Creates a border around the hot key control. Type: Bool. Default: False.

**Modal Frame**
Creates a control that has a double border. Type: Bool. Default: False.

**Transparent**
Specifies that a control created with this style is to be transparent. Any controls that are beneath this control are not obscured by this control. A control with this style receives WM_PAINT messages only after all sibling controls beneath it have been updated. Type: Bool. Default: False.

**Accept Files**
A control with this style accepts drag-drop files. If a user drops a file on this control, WM_DROPFILES messages will be sent to the control.Type: Bool. Default: False.

**Left Scrollbar**
Vertical scroll bar (if present) is to the left of the client area. Type: Bool. Default: False.

**Right-to-Left Reading Order**
The control text is displayed using Right to Left reading order properties. Type: Bool. Default: False.

**Right Aligned Text**
Specifies that text is right-aligned within the control. Type: Bool. Default: False.

**No Parent Notify**
Specifies that the child window does not send the WM_PARENTNOTIFY message to its parent window. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Hot Key Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: False.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: True.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.

**Border**
Creates a border around the hot key control. Type: Bool. Default: True.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Icon Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Width**
Image width in pixels. This property is determined by the currently selected target-device definition. (This property is displayed only and cannot be modified on the property page).

**Height**
Image height in pixels. This property is determined by the currently selected target-device definition. (This property is displayed only and cannot be modified on the property page).

**Colors**
Monochrome (2) or 16. This property is determined by the currently selected target-device definition. (This property is displayed only and cannot be modified on the property page).

**File Name**
The name of the file containing the icon resource.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Icon Properties: Resource

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Language**
This drop-down list box contains the language to be used for this resource.

**Condition**
Determines the inclusion of the resource. For example, if the condition is _DEBUG, this resource would only be included in debug builds.

**File Name**
The name of the file containing the resource.

**Preview**
A box showing what the bitmap looks like. It is useful for browsing through graphics resources without opening them.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# FILEFLAGS Properties: General

**VS_FF_DEBUG**
Specifies that the file contains debugging information.

**VS_FF_PRERELEASE**
Specifies that the file is a development version.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# List Box Properties: Extended Styles

**Client Edge**
Creates a border with a sunken edge around the list box control. Type: Bool. Default: False.

**Static Edge**
Creates a border around the list box control. Type: Bool. Default: False.

**Modal Frame**
Creates a control that has a double border. Type: Bool. Default: False.

**Transparent**
Specifies that a control created with this style is to be transparent. Any controls that are beneath this control are not obscured by this control. A control with this style receives WM_PAINT messages only after all sibling controls beneath it have been updated. Type: Bool. Default: False.

**Accept Files**
A control with this style accepts drag-drop files. If a user drops a file on this control, WM_DROPFILES messages will be sent to the control.Type: Bool. Default: False.

**Left Scrollbar**
Vertical scroll bar (if present) is to the left of the client area. Type: Bool. Default: False.

**Right-to-Left Reading Order**
The control text is displayed using Right to Left reading order properties. Type: Bool. Default: False.

**Right Aligned Text**
Specifies that text is right-aligned within the control. Type: Bool. Default: False.

**No Parent Notify**
Specifies that the child window does not send the WM_PARENTNOTIFY message to its parent window. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# List Box Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: False.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: True.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# List Box Properties: Styles

**Selection**
Determines how items in a list box can be selected. Possible values are as follows:

- Single (default): Only one item in a list box can be selected at a time.
- Multiple: More than one list-box item can be selected, but the SHIFT and CTRL keys have no effect. Clicking or double-clicking an unselected item selects it. Clicking or double-clicking a selected item deselects it.
- Extended: The SHIFT and CTRL keys can be used together with the mouse to select and deselect list-box items, select groups of items, and select non-adjacent items.

**Owner Draw**
Controls the owner-draw characteristics of the list box. This property can be one of the following values:

- No (default): Turns off the owner-draw style. The list box contains strings.
- Fixed: Specifies that the owner of the list box is responsible for drawing its contents and that the items in the list box are the same height.

  **CWnd::OnMeasureItem** is called when the list box is created and **CWnd::OnDrawItem** is called when a visual aspect of the list box has changed.
- Variable: Specifies that the owner of the list box is responsible for drawing its contents and that the items in the list box are variable in height.

  **CWnd::OnMeasureItem** is called for each item in the list when the list box is created and **CWnd::OnDrawItem** is called when a visual aspect of the list box has changed.

**Has Strings**
Specifies that an owner-draw list box contains items consisting of strings. The list box maintains the memory and pointers for the strings so the application can use the **LB_GETTEXT** message to retrieve the text for a particular item. By default, all list boxes except owner-draw list boxes have this style. An application can create an owner-draw list box either with or without this style.

This style is only available if the Owner Draw property is set to either Fixed or Variable. If Owner Draw is set to No, the list box contains strings by default. Type: Bool. Default: False.

**Border**
Creates a border around the list box. Type: Bool. Default: True.

**Sort**
Sorts the contents of the list box alphabetically. Type: Bool. Default: True.

**Notify**
Notifies the parent window if a list item has been clicked or double-clicked. Type: Bool. Default: True.

**Multicolumn**
Specifies a multicolumn list box that is scrolled horizontally. The **LB_SETCOLUMNWIDTH** message sets the width of the columns. Type: Bool. Default: False.

**Horiz. Scroll**
Creates a list box with a horizontal scroll bar. Type: Bool. Default: False.

**Vert. Scroll**
Creates a list box with a vertical scroll bar. Type: Bool. Default: True.

**No Redraw**
Specifies that the list box's appearance is not updated when changes are made. This style can be changed at any time by sending a **WM_SETREDRAW** message or by calling **CWnd::SetRedraw**. Type: Bool. Default: False.

**Use Tabstops**
Allows a list box to recognize and expand tab characters when drawing its strings. The default tab positions are 32 dialog box units (DLUs). Type: Bool. Default: False.

**Want Key Input**
Specifies that the owner of the list box receives **WM_VKEYTOITEM** or **WM_CHARTOITEM** messages whenever the user presses a key and the list box has the input focus. This allows an application to perform special processing on the keyboard input. If a list box uses the Has Strings style, the list box receives **WM_VKEYTOITEM** messages. If a list box does not use the Has Strings style, it receives **WM_CHARTOITEM** messages. Type: Bool. Default: False.

**Disable No Scroll**
Shows a disabled vertical scroll bar in the list box when the box does not contain enough items to scroll. Without this style, the scroll bar is hidden when the list box does not contain enough items to scroll. Type: Bool. Default: False.

**No Integral Height**
Specifies that the size of the list box is exactly the size specified by the application when it created the list box. Normally, Windows sizes a list box so that the list box does not display partial items. Type: Bool. Default: True.

**No Data**
If checked, list box does not store item data. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# List Control Properties: Extended Styles

**Client Edge**
Creates a border with a sunken edge around the list control. Type: Bool. Default: False.

**Static Edge**
Creates a border around the list control. Type: Bool. Default: False.

**Modal Frame**
Creates a control that has a double border. Type: Bool. Default: False.

**Transparent**
Specifies that a control created with this style is to be transparent. Any controls that are beneath this control are not obscured by this control. A control with this style receives WM_PAINT messages only after all sibling controls beneath it have been updated. Type: Bool. Default: False.

**Accept Files**
A control with this style accepts drag-drop files. If a user drops a file on this control, WM_DROPFILES messages will be sent to the control.Type: Bool. Default: False.

**Left Scrollbar**
Vertical scroll bar (if present) is to the left of the client area. Type: Bool. Default: False.

**Right-to-Left Reading Order**
The control text is displayed using Right to Left reading order properties. Type: Bool. Default: False.

**Right Aligned Text**
Specifies that text is right-aligned within the control. Type: Bool. Default: False.

**No Parent Notify**
Specifies that the child window does not send the WM_PARENTNOTIFY message to its parent window. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# List Control Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: False.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: True.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# List Control Properties: Styles

**View**
Sets the display view for the list control:

- Icon (default): Specifies icon view.
- Small Icon: Specifies small icon view.
- List: Specifies list view.
- Report: Specifies report view.

**Align**
Aligns the icons in the list:

- Top (default): Top aligns the icons in the view.
- Left: Left aligns the icons in the view.

**Sort**
Sorts the icons in the list in the following order:

- None (default): No sort applied.
- Ascending: Sorts items based on item text in ascending order.
- Descending: Sorts items based on item text in descending order.

**Auto Arrange**
Specifies that icons are automatically kept arranged in icon and small icon view. Type: Bool. Default: False.

**Single Selection**
Allows only one item at a time to be selected. By default, multiple items may be selected. Type: Bool. Default: False.

**Share Image List**
Specifies that the control does not take ownership of the image lists assigned to it;   that is, it does not destroy the image lists when it is destroyed. This style enables the same image lists to be used with multiple list view controls. Type: Bool. Default: False.

**No Label Wrap**
Displays item text on a single line in icon view. By default, item text may wrap in icon view. Type: Bool. Default: False.

**Edit Labels**
Allows item text to be edited in place. The parent window must process the LVN_ENDLABELEDIT notification message. Type: Bool. Default: False.

**Owner Draw Fixed**
Enables the owner window to paint items in report view. The list view control sends a WM_DRAWITEM message to paint each item; it does not send separate messages for each subitem. The itemData member of the DRAWITEMSTRUCT structure contains the item data for the specified list view item. Type: Bool. Default: False.

**No Scroll**
Disables scrolling. All items must be within the client area. Type: Bool. Default: True.

**No Column Header**
Specifies that a column header is not displayed in report view. By default, columns have headers in report view. Type: Bool. Default: False.

**No Sort Header**
Specifies that column headers do not work like buttons. This style is useful if clicking a column header in report view does not carry out an action, such as sorting. Type: Bool. Default: False.

**Border**
Creates a border around the list view control. Type: Bool. Default: True.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Menu Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Menu Properties: Resource

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Language**
This drop-down list box contains the language to be used for this resource.

**Condition**
Determines the inclusion of the resource. For example, if the condition is _DEBUG, this resource would only be included in debug builds.

**Preview**
A box showing what the bitmap looks like. It is useful for browsing through graphics resources without opening them.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Menu Item Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Caption**
The text that appears as part of the menu item to label it. To make one of the letters in the caption of a menu item the mnemonic key, precede it with an ampersand (&).

**Separator**
If True, the menu item is a separator. Type: Bool. Default: False.

**Checked**
If True, the menu item is initially checked. Type: Bool. Default: False.

**Pop-up**
If True, the menu item is a pop-up menu (a submenu). Type: Bool. Default: TRUE for top-level menu items on a menu bar; otherwise False.

**Grayed**
If True, the menu item is initially grayed and inactive. Type: Bool. Default: False.

**Inactive**
If the Grayed property is True, then the Inactive property is always True. Otherwise Inactive determines whether the menu item is initially inactive. Type: Bool. Default: False.

**Help**
Right justifies the menu item on the menu bar at run time. Type: Bool. Default: False.

**Break**
Can be one of these values:

- None (Default): No break.
- Column: For static menu-bar items, this value places the item on a new line. For pop-up menus, this value places the item in a new column with no dividing line between the columns. Setting this property affects the appearance of the menu only at run time, not in the menu editor.
- Bar: Same as Column except, for pop-up menus, this value separates the new column from the old column with a vertical line. Setting this property affects the appearance of the menu only at run time, not in the menu editor.

**Prompt**
Contains text to appear in the status bar when this menu item is highlighted. The text is placed in the string table with the same identifier as the menu item. This property is available only in resource files with Microsoft Foundation Class Library (MFC) support.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Multiple Property Pages

You have requested property information on more than one object. Before requesting property information, make sure that you have selected only one appropriate object (for example, a source file or icon).

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# No Property Page

You have requested property information on an object that has no associated properties. Before requesting property information, make sure that you have selected an appropriate object (for example, a source file or icon).

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Null Property Page

You have requested property information on an object that has no associated properties. Before requesting property information, make sure that you have selected an appropriate object (for example, a source file or icon).

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# OLE Control Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: False.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: False.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# OLE Control Properties: All

To change or edit a value, select the value from the list of values and edit the current value as it appears in the edit box.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Pane Properties: General

**Pane Name**
The name of the pane.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Picture: Extended Styles

**Client Edge**
Creates a border with a sunken edge around the picture control. Type: Bool. Default: False.

**Static Edge**
Creates a border around the picture control. Type: Bool. Default: False.

**Modal Frame**
Creates a control that has a double border. Type: Bool. Default: False.

**Transparent**
Specifies that a control created with this style is to be transparent. Any controls that are beneath this control are not obscured by this control. A control with this style receives WM_PAINT messages only after all sibling controls beneath it have been updated. Type: Bool. Default: False.

**Accept Files**
A control with this style accepts drag-drop files. If a user drops a file on this control, WM_DROPFILES messages will be sent to the control.Type: Bool. Default: False.

**No Parent Notify**
Specifies that the child window does not send the WM_PARENTNOTIFY message to its parent window. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Picture Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: False.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: False.

**Type**
Sets the type of static graphic to display. One of the following:

- Frame (Default): Displays a frame. You set the color of the frame in the Color box. Use a frame to visually group controls.
- Rectangle: Displays a filled rectangle. You set the color of the rectangle in the Color box.
- Icon: Displays an icon in the dialog box. Use the image box to specify the identifier of the icon you want to display.
- Bitmap: Displays a bitmap in the dialog box. Use the image box to specify the identifier of the bitmap you want to display.
- Enhanced Metafile: Displays an enhanced metafile in the dialog box.

**Image**
Select the identifier of the icon or bitmap to display. This property is only available when the picture type is icon or bitmap.

**Color**
Sets the color of a frame or rectangle to black, white, gray, or etched. Etched gives it a 3-D appearance. This property is not available when the picture type is icon, bitmap or enhanced metafile. Default: Black.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Picture: Styles

**Sunken**
Creates a border with a sunken edge around the picture control. Type: Bool. Default: False.

**Border**
Creates a border around the picture. Type: Bool. Default: True.

**Notify**
Notifies the parent window if a picture has been clicked or double-clicked. Type: Bool. Default: False.

**Center Image**
Specifies that, if the bitmap or icon is smaller than the client area of the picture control, the rest of the client area is filled with the color of the pixel in the top left corner of the bitmap or icon. Type: Bool. Default: False.

**Right Justify**
Specifies that the lower right corner of a picture control is to remain fixed when the control is resized. Only the top and left sides are adjusted to accommodate a new bitmap or icon. Type: Bool. Default: False.

**Real Size Image**
Prevents a static icon or bitmap control from being resized as it is loaded or drawn. If the icon or bitmap is larger than the destination area, the image is clipped. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Program Variable Properties

This properties page appears when you select a variable or expression in the Variables window or Watch window, then select Properties from the Edit window. It provides the following information for the selected variable or expression:

**Type Data type of the variable or expression.**

**Expression**
Name or representation of the variable or expression.

**Value**
Contents of the variable or expression.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Progress Properties: Extended Styles

**Client Edge**
Creates a border with a sunken edge around the progress control. Type: Bool. Default: False.

**Static Edge**
Creates a border around the progress control. Type: Bool. Default: False.

**Modal Frame**
Creates a control that has a double border. Type: Bool. Default: False.

**Transparent**
Specifies that a control created with this style is to be transparent. Any controls that are beneath this control are not obscured by this control. A control with this style receives WM_PAINT messages only after all sibling controls beneath it have been updated. Type: Bool. Default: False.

**Accept Files**
A control with this style accepts drag-drop files. If a user drops a file on this control, WM_DROPFILES messages will be sent to the control.Type: Bool. Default: False.

**Right-to-Left Reading Order**
The control text is displayed using Right to Left reading order properties. Type: Bool. Default: False.

**No Parent Notify**
Specifies that the child window does not send the WM_PARENTNOTIFY message to its parent window. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Progress Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: False.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: False.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.

**Border**
Creates a border around the progress control. Type: Bool. Default: True.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Project Folder Properties: General

**Project**
Identifies the project and the default configuration displayed.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Project Folder Properties: General

The Project Folder Properties General property page is a read-only property page that displays the filename and last modified date for the output file.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Project Folder Properties: Inputs

This is a read-only property page.

**Tool**
Tool that processes this input file. This is the tool that runs if you choose Compile from the pop-up menu with this file selected.

**Files**
Name of the input file relative to the project directory.

**Last modified date**
Date on which the file was last modified.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Project Folder Properties: Outputs

This is a read-only property page.

**Tool**
Tool that produces the output files. This is the tool that runs if you choose Compile from the pop-up menu with the input file selected.

**Files**
Names of the output files, relative to the project directory.

**Last modified date**
Date on which the selected file in the Files list was last modified.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Pushbutton Properties: Extended Styles

**Client Edge**
Creates a border with a sunken edge around the pushbutton control. Type: Bool. Default: False.

**Static Edge**
Creates a border around the pushbutton control. Type: Bool. Default: False.

**Modal Frame**
Creates a control that has a double border. Type: Bool. Default: False.

**Transparent**
Specifies that a control created with this style is to be transparent. Any controls that are beneath this control are not obscured by this control. A control with this style receives WM_PAINT messages only after all sibling controls beneath it have been updated. Type: Bool. Default: False.

**Accept Files**
A control with this style accepts drag-drop files. If a user drops a file on this control, WM_DROPFILES messages will be sent to the control.Type: Bool. Default: False.

**Right-to-Left Reading Order**
The control text is displayed using Right to Left reading order properties. Type: Bool. Default: False.

**No Parent Notify**
Specifies that the child window does not send the WM_PARENTNOTIFY message to its parent window. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Push Button Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Caption**
The text that appears as part of the control to label it. To make one of the letters in the caption of a control the mnemonic key, precede it with an ampersand (&). Default: A name based on the type of control (for example, "Check") plus a number based on the resource identifier assigned by Visual C++.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: False.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: True.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Pushbutton Properties: Styles

**Default Button**
If True, the control is the default button in the dialog box. The default button is drawn with a heavy black border when the dialog box first appears and is executed if the user presses ENTER without choosing another command in the dialog box. Windows allows only one default button in a dialog box. Type: Bool. Default: False.

**Owner Draw**
Creates an owner-draw button. Use an owner-draw button when you need to customize the appearance of a control by providing your own **OnDrawItem** message handler in the owner-window procedure (usually a dialog-box procedure or class derived from the Microsoft Foundation class **CDialog** or **CFormView**). You can also derive your own class from **CButton** and override **CButton::DrawItem**. See **CWnd::OnDrawItem** and **CButton::OnDraw** in the *Class Library Reference* for more information.

**Icon**
Specifies that the button displays an icon.Type: Bool. Default: False.

**Bitmap**
Specifies that the button displays a bitmap. Type: Bool. Default: False.

**Multiline**
Wraps the button text to multiple lines if the text string is too long to fit on a single line in the button rectangle.Type: Bool. Default: False.

**Notify**
Notifies the parent window if a pushbutton has been clicked or double-clicked. Type: Bool. Default: True.

**Flat**
Makes a button look flat, not three-dimensional. Type: Bool. Default: False.

**Horizontal Alignment**
Positions the control's caption text to the left, center, right or default position in the control.

**Vertical Alignment**
Positions the control's caption text to the top, bottom, center or default position in the control.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Radio Button Properties: Extended Styles

**Client Edge**
Creates a border with a sunken edge around the radio button control. Type: Bool. Default: False.

**Static Edge**
Creates a border around the radio button control. Type: Bool. Default: False.

**Modal Frame**
Creates a control that has a double border. Type: Bool. Default: False.

**Transparent**
Specifies that a control created with this style is to be transparent. Any controls that are beneath this control are not obscured by this control. A control with this style receives WM_PAINT messages only after all sibling controls beneath it have been updated. Type: Bool. Default: False.

**Accept Files**
A control with this style accepts drag-drop files. If a user drops a file on this control, WM_DROPFILES messages will be sent to the control.Type: Bool. Default: False.

**Right-to-Left Reading Order**
The control text is displayed using Right to Left reading order properties. Type: Bool. Default: False.

**No Parent Notify**
Specifies that the child window does not send the WM_PARENTNOTIFY message to its parent window. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Radio Button Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Caption**
The text that appears as part of the control to label it. To make one of the letters in the caption of a control the mnemonic key, precede it with an ampersand (&). Default: A name based on the type of control (for example, "Check") plus a number based on the resource identifier assigned by Visual C++.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: False.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: False.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Radio Button Properties: Styles

**Auto**
When the user selects a radio button with this property, the radio button is automatically selected and any other radio buttons in the same group are cleared (deselected). You must set this property to True if you are using a group of radio buttons with Dialog Data Exchange. Type: Bool. Default: True.

**Left Text**
Places the radio button's caption text on the left rather than the right. Type: Bool. Default: False.

**Pushlike**
Makes a button (such as a check box, three-state check box, or radio button) look and act like a push button. The button looks raised when it isn't pushed or checked, and sunken when it is pushed or checked. Type: Bool. Default: False.

**Multiline**
Wraps the button text to multiple lines if the text string is too long to fit on a single line in the button rectangle.Type: Bool. Default: False.

**Notify**
Notifies the parent window if a radio button has been clicked or double-clicked. Type: Bool. Default: True.

**Flat**
Makes a button look flat, not three-dimensional. Type: Bool. Default: False.

**Icon**
Specifies that the button displays an icon.Type: Bool. Default: False.

**Bitmap**
Specifies that the button displays a bitmap. Type: Bool. Default: False.

**Horizontal Alignment**
Positions the control's caption text to the left, center, right or default position in the control.

**Vertical Alignment**
Positions the control's caption text to the top, bottom, center or default position in the control.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Resource File Properties: General

Use the Resource File Properties General property page to enable MFC features.

**File Name**
The name of the file. (This property is displayed only and cannot be modified on the property page.)

**Enable MFC Features**
If checked, the file contains MFC features. Type: Bool. Default: True.

**Use 3D Controls**
Display controls using 3-D effects. Type: Bool. Default: True.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Scrollbar Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: False.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: False.

**Align**
One of the following values:

- None (Default): No special alignment is performed. The size of the scroll bar is the size specified in Visual C++.
- Top/Left: Aligns the upper-left corner of the scroll bar with the upper-left corner of the containing window specified in Visual C++.
- Bottom/Right: Aligns the lower-right corner of the scroll bar with the lower-right corner of the containing window specified in Visual C++.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Slider Properties: Extended Styles

**Client Edge**
Creates a border with a sunken edge around the slider control. Type: Bool. Default: False.

**Static Edge**
Creates a border around the slider control. Type: Bool. Default: False.

**Modal Frame**
Creates a control that has a double border. Type: Bool. Default: False.

**Transparent**
Specifies that a control created with this style is to be transparent. Any controls that are beneath this control are not obscured by this control. A control with this style receives WM_PAINT messages only after all sibling controls beneath it have been updated. Type: Bool. Default: False.

**Accept Files**
A control with this style accepts drag-drop files. If a user drops a file on this control, WM_DROPFILES messages will be sent to the control.Type: Bool. Default: False.

**Right-to-Left Reading Order**
The control text is displayed using Right to Left reading order properties. Type: Bool. Default: False.

**No Parent Notify**
Specifies that the child window does not send the WM_PARENTNOTIFY message to its parent window. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Slider Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: False.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: True.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

## Slider Properties: Styles

**Orientation**
Displays the trackbar (slider) horizontally (default) or vertically.

**Point**
Displays tick marks on either or both sides of the trackbar (slider) in the following orientation:

- Both (Default): Displays tick marks on both sides of the trackbar.
- Top/Left: Displays tick marks on one side of the trackbar; on the top of a horizontal trackbar, to the left of a vertical trackbar.
- Bottom/Right: Displays tick marks on the other side of the trackbar, on the bottom of a horizontal trackbar, and to the right of a vertical trackbar.

**Tick Marks**
Specifies the display of tick marks on a trackbar (slider). Type: Bool. Default: False.

**Autoticks**
Specifies that a tick mark is placed for each increment in the trackbar's (slider's) range of values. These tick marks are created automatically when an application sends the TBM_SETRANGE message. Type: Bool. Default: False.

**Enable Selection**
Specifies that a selection range is displayed (with triangles and a highlighted area) on the trackbar (slider). Type: Bool. Default: False.

**Border**
Creates a border around the track bar control. Type: Bool. Default: False.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Source File Properties: General

This is a read-only property page.

**Filename**
Fully qualified name of the file.

**Last modified date**
Date on which the file was last modified.

**Status**
Shows the source-code control status, if the file is under source-code control in a source-code control system that conforms to the Microsoft Source Code Control API Specification.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Source File Properties: Dependencies

This is a read-only property page.

**Tool**
Tool that processes the input file that uses these dependent files. This is the tool that runs if you choose Compile from the pop-up menu with the input file selected.

**Files**
Names of the files on which the input file depends, relative to the project directory.

**Last modified date**
Date on which the selected file in the Files list was last modified.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Source Window Properties: General

Use the Source Window Properties General property page to enable syntax-coloring or to set a file as read only.

**File Name**
The name of the file. (This property is displayed only and cannot be modified on the property page).

**Language**
This drop-down list box contains language names. Your choice determines the syntax coloring.

**Size**
The size of the file in lines. (This property is displayed only and cannot be modified on the property page).

**Saved**
The date the file ws saved. (This property is displayed only and cannot be modified on the property page).

**Tab Size**
The number of spaces in a tab stop.

**Indent Size**
The amout of space used when indenting or unindenting text.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Spin Properties: Extended Styles

**Client Edge**
Creates a border with a sunken edge around the slider control. Type: Bool. Default: False.

**Static Edge**
Creates a border around the slider control. Type: Bool. Default: False.

**Modal Frame**
Creates a control that has a double border. Type: Bool. Default: False.

**Transparent**
Specifies that a control created with this style is to be transparent. Any controls that are beneath this control are not obscured by this control. A control with this style receives WM_PAINT messages only after all sibling controls beneath it have been updated. Type: Bool. Default: False.

**Accept Files**
A control with this style accepts drag-drop files. If a user drops a file on this control, WM_DROPFILES messages will be sent to the control.Type: Bool. Default: False.

**No Parent Notify**
Specifies that the child window does not send the WM_PARENTNOTIFY message to its parent window. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Spin Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: False.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: False.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Spin Properties: Styles

**Orientation**
Displays the spin (up-down) control horizontally or vertically (default).

**Alignment**
Value can be set as follows:

- Unattached (Default).
- Left: Positions the up-down control next to the left edge of the buddy window. The buddy window is moved to the right and its width decreased to accommodate the width of the up-down control.
- Right: Positions the up-down control next to the right edge of the buddy window. The width of the buddy window is decreased to accommodate the width of the up-down control.

**Auto Buddy**
Automatically selects the previous window in the Z order as the up-down control's buddy window. The buddy window displays as text the values set by the spin control.Typically, the buddy window is an edit control or a static text control. Type: Bool. Default: False.

**Set Buddy Integer**
Causes the up-down control to set the text of the buddy window (using the WM_SETTEXT message) when the position changes. The text consists of the position formatted as a decimal or hexadecimal string. Type: Bool. Default: False.

**No Thousands**
Does not insert a thousands separator between every three decimal digits. Type: Bool. Default: False.

**Wrap**
Causes the position to "wrap" if it is incremented or decremented beyond the ending or beginning of the range. Type: Bool. Default: False.

**Arrow Key**
Causes the up-down control to increment and decrement the position when the UP arrow and DOWN arrow keys are pressed. Type: Bool. Default: True.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Stored Procedure Properties

**Name**
The name of the stored procedure.

**Return Type**
The data type returned by the stored procedure.

**Owner**
The user name as reported by the DBMS. This may differ from the login name.

**Parameters**
A list of each of the stored procedure's parameters that includes the name, data type, and whether the parameter accepts and/or returns values.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

## String Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Caption**
A string of up to 255 bytes (single characters, escape sequences, or ASCII values). Type: Text.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# String Table Properties: Resource

For string tables, the ID is always **STRINGTABLE** and cannot be modified.

**Language**
This drop-down list box contains the language to be used for this resource.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Tab Control Properties: Extended Styles

**Modal Frame**
Creates a control that has a double border. Type: Bool. Default: False.

**Client Edge**
Creates a border with a sunken edge around the tab control. Type: Bool. Default: False.

**Static Edge**
Creates a border around the tab control. Type: Bool. Default: False.

**Transparent**
Specifies that a control created with this style is to be transparent. Any controls that are beneath this control are not obscured by this control. A control with this style receives WM_PAINT messages only after all sibling controls beneath it have been updated. Type: Bool. Default: False.

**Accept Files**
A control with this style accepts drag-drop files. If a user drops a file on this control, WM_DROPFILES messages will be sent to the control.Type: Bool. Default: False.

**Control Parent**
Allows the user to navigate among the child windows of the tab control by using the TAB key. Type: Bool. Default: False.

**Left Scrollbar**
Vertical scroll bar (if present) is to the left of the client area. Type: Bool. Default: False.

**Right-to-Left Reading Order**
The control text is displayed using Right to Left reading order properties. Type: Bool. Default: False.

**Right Aligned Text**
Specifies that text is right-aligned within the control. Type: Bool. Default: False.

**No Parent Notify**
Specifies that the child window does not send the WM_PARENTNOTIFY message to its parent window. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Tab Control Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: False.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: False.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Tab Control Properties: Styles

**Alignment**
Can be one of the following values:

- Right Justify (Default): Width of each tab is increased so that each row of tabs fills the entire width of the tab control.
- Fixed Width: Sizes tabs to the width of the widest label.
- Ragged Right: In multiline tabs, the width of the tabs does not stretch to fill the rows.

**Focus**
Can be one of the following values:

- Default: Specifies that the user can use the keyboard to give the input focus to a tab in the control.
- On Button Down: Specifies that a tab recieves the input focus when clicked.
- Never: Specifies that a tab never recieves the input focus when clicked.

**Buttons**
Specifies that the tabs in the control resemble buttons. Tabs in this type of tab control should serve the same function as button controls; that is, clicking a tab should carry out a command instead of displaying a page. Type: Bool. Default: False.

**Tool Tips**
Specifies that a tool tip is created for each tab in the tab control. Type: Bool. Default: False.

**Share Image Lists**
Specifies that the control does not take ownership of the image lists assigned to it;   that is, it does not destroy the image lists when it is destroyed. This style enables the same image lists to be used with multiple controls. Type: Bool. Default: False.

**Multiline**
Displays multiple rows of tabs.Type: Bool. Default: False.

**Force Icon Left**
Left aligns the icon, leaving the label centered. Type: Bool. Default: False.

**Force Label Left**
Left-aligns both the icon and label. Type: Bool. Default: False.

**Owner Draw Fixed**
Specifies that the parent window draws the tabs in the control. Type: Bool. Default: False.

**Border**
Creates a border around the tab control. Type: Bool. Default: False.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Table or View Properties

**Name**
The name of the currently selected table or view.

**Owner**
The user name as reported by the DBMS. This may differ from the login name.

This property page also tells you whether the currently selected table or view can be updated.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Text Properties: Extended Styles

**Client Edge**
Creates a border with a sunken edge around the text control. Type: Bool. Default: False.

**Static Edge**
Creates a border around the text control. Type: Bool. Default: False.

**Modal Frame**
Creates a control that has a double border. Type: Bool. Default: False.

**Transparent**
Specifies that a control created with this style is to be transparent. Any controls that are beneath this control are not obscured by this control. A control with this style receives WM_PAINT messages only after all sibling controls beneath it have been updated. Type: Bool. Default: False.

**Accept Files**
A control with this style accepts drag-drop files. If a user drops a file on this control, WM_DROPFILES messages will be sent to the control.Type: Bool. Default: False.

**No Parent Notify**
Specifies that the child window does not send the WM_PARENTNOTIFY message to its parent window. Type: Bool. Default: False.

**Right-to-Left Reading Order**
The control text is displayed using Right to Left reading order properties. Type: Bool. Default: False.

**Right Aligned Text**
Specifies that text is right-aligned within the control. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Text Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: True.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: False.

**Caption**
The text that appears as part of the control to label it. To make one of the letters in the caption of a control the mnemonic key, precede it with an ampersand (&). Default: A name based on the type of control (for example, "Check") plus a number based on the resource identifier assigned by Visual C++.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Text Properties: Styles

**Align Text**
Controls how text is aligned in the static-text control. The possible values are Left, Center, and Right. Set this option to Left when No Wrap is selected. Default: Left.

**No Prefix**
Prevents ampersands (&) in the control's text from being interpreted as the mnemonic character. Normally a string containing an ampersand is displayed with the ampersand removed and the next character in the string underlined. The No Prefix style is most often used when filenames or other strings that may contain an ampersand need to be displayed.

**No Wrap**
Displays text left-aligned. Tabs are expanded but words are not wrapped. Text that extends past the end of a line is clipped. Type: Bool. Default: False.

**Simple**
Disables No Wrap and Text Align. Text in static text controls with this property set does not wrap and is not clipped. In addition, setting this property means that overriding **WM_CTLCOLOR** in the parent window has no effect on the control. Type: Bool. Default: False.

**Notify**
Notifies the parent window if a check box has been clicked or double-clicked. Type: Bool. Default: False.

**Sunken**
Creates a border with a sunken edge around the static text control. Type: Bool. Default: False.

**Border**
Creates a border around the text control. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Toolbar Button Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Width**
The width of the toolbar button.

**Height**
The height of the toolbar button.

**Prompt**
Contains text to appear in the status bar when this menu item is highlighted. The text is placed in the string table with the same identifier as the menu item. This property is available only in resource files with Microsoft Foundation Class Library (MFC) support.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Toolbar Properties: Resource

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Language**
This drop-down list box contains the language to be used for this resource.

**Condition**
Determines the inclusion of the resource. For example, if the condition is _DEBUG, this resource would only be included in debug builds.

**File Name**
The name of the file containing the resource.

**Preview**
A box showing what the bitmap looks like. It is useful for browsing through graphics resources without opening them.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Tree Control Properties: Extended Styles

**Client Edge**
Creates a border with a sunken edge around the tree control. Type: Bool. Default: False.

**Static Edge**
Creates a border around the tree control. Type: Bool. Default: False.

**Modal Frame**
Creates a control that has a double border. Type: Bool. Default: False.

**Transparent**
Specifies that a control created with this style is to be transparent. Any controls that are beneath this control are not obscured by this control. A control with this style receives WM_PAINT messages only after all sibling controls beneath it have been updated. Type: Bool. Default: False.

**Accept Files**
A control with this style accepts drag-drop files. If a user drops a file on this control, WM_DROPFILES messages will be sent to the control.Type: Bool. Default: False.

**Left Scrollbar**
Vertical scroll bar (if present) is to the left of the client area. Type: Bool. Default: False.

**Right-to-Left Reading Order**
The control text is displayed using Right to Left reading order properties. Type: Bool. Default: False.

**Right Aligned Text**
Specifies that text is right-aligned within the control. Type: Bool. Default: False.

**No Parent Notify**
Specifies that the child window does not send the WM_PARENTNOTIFY message to its parent window. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Tree Control Properties: General

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Visible**
Determines whether or not the control is visible when the application is first run. Type: Bool. Default: True.

**Disabled**
Determines if the resource is displayed as disabled when the dialog box is created. Type: Bool. Default: False.

**Group**
Specifies the first control of a group of controls in which the user can move from one control to the next by using the arrow keys. All controls in the tab order after the first control with the Group property set to False belong to the same group. The next control in the tab order with Group set to True ends the first group of controls and starts the next group. Type: Bool. Default: False.

**Tabstop**
Specifies that the user can move to this control with the TAB key. Type: Bool. Default: True.

**Help ID**
Assigns a help ID to the control. The help ID is based on the resource ID. Type: Bool. Default: False.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Tree Control Properties: Styles

**Has Buttons**
Displays plus (+) and minus (-) buttons next to parent items. The user clicks the buttons to expand or collapse a parent item's list of child items. To include buttons with items at the root of the tree view, Lines at Root must be set to true. Type: Bool. Default: False.

**Has Lines**
Uses lines to show the hierarchy of items. Type: Bool. Default: False.

**Border**
Creates a border around the tree view control. Type: Bool. Default: True.

**Lines at Root**
Uses lines to link items at the root of the tree view control. This value is ignored if Has Lines is set to false.

**Edit Labels**
Allows the user to edit the labels of tree view items. Type: Bool. Default: True.

**Disable Drag Drop**
Prevents the tree view control from sending TVN_BEGINDRAG notification messages. Type: Bool. Default: False.

**Show Selection Always**
Uses the system highlight colors to draw the selected item. Type: Bool. Default: False.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

## Trigger Properties

**Name**
The name of the trigger.

**Owner**
The user name as reported by the DBMS. This may differ from the login name.

**Type**
The conditions under which the trigger will fire. This will be either INSERT, DELETE, and/or UPDATE because a trigger can fire when data is inserted into, deleted from, and/or updated in a table.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# User Defined Data Type Properties

This property page shows the following properties of the selected user-defined data type:

**Name**
The name of the user-defined data type.

**Base Datatype**
The type on which the user-defined data type is based.

The property page also shows whether the user-defined data type is variable data type and whether columns that use this type can take a **NULL** value.

For information on the Precision, Length, and Scale Data Type fields, see the topic, "Precision, Scale, Length, and Display Size" in the ODBC SDK documentation found in Books Online. You can use the Search command from the Help menu to quickly find this topic.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Version Properties: Resource

**ID**
The resource ID is a symbol and is defined in the header file. Type: Symbol, Integer, or Quoted String.

**Language**
This drop-down list box contains the language to be used for this resource.

**Condition**
Determines the inclusion of the resource. For example, if the condition is _DEBUG, this resource would only be included in debug builds.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

## About Windows and Pop-up Menus

Microsoft Developer Studio contains three types of windows—Multiple Document Interface (MDI) document windows, dockable windows, and browse windows. Some windows are available for use only when a particular tool is active, such as the debugger or the dialog editor. Other windows will always be available, regardless of the current action. This reference lists all windows alphabetically.

In many windows, clicking the right mouse button will display a pop-up menu that contains commands applicable in the given context. The commands on the pop-up menu are all equivalent to existing menu commands.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Types of Windows in Microsoft Developer Studio

The MDI document windows include the text editor, workspace, and resource editor windows. All MDI windows are sizable and minimizable within Developer Studio. The Tile Horizontally and Tile Vertically commands on the Window menu arrange any open MDI windows so that none of the windows overlap.

The dockable windows include the debugging windows (Watch, Locals, Register, Memory, Call Stack, and Disassembly) and the Output window. The Developer Studio toolbars and controls also appear in dockable forms. Dockable windows contain a thin border and thin title bar. They are not minimizable and are not affected by the tiling commands. They always appear on top of MDI document windows in the workspace.

When a dockable window or toolbar is dragged to the edge of the Developer Studio application frame, its outline changes shape to match the application frame size. This indicates that it is ready to "dock" itself to the window frame either horizontally or vertically, depending on which edge of the application window it touches. To complete docking the window, just release the mouse button. To undock a window, grab the thin window frame and drag the window into the central area of the Developer Studio application frame.

**Tip**    You can use the Docking View command on the window pop-up menu to switch a window between dockable window and MDI window views. You can also double-click the window frame to dock and undock the window.

Browse windows display information about the symbols in your program. Before you build a project, you can set an option to have the compiler create a file with the information about the symbols in your project. This file has the project's base name and the extension .BSC. You can also browse through any .BSC file created with the BSCMAKE.EXE utility located in the directory with other executable files. You view browse information in browse windows, which have different appearances and different controls depending on the type of information that they are displaying.

In browse windows, you can examine:

- Information about all the symbols in any source file
- The source code line in which a symbol is defined
- Each source code line that contains a reference to a symbol
- The relationships between base classes and derived classes
- The relationships between calling functions and called functions

{ewl msdncd.dll, ewcright, /c"Microsoft"}

## Common Window Features

Developer Studio uses the standard commands used by most applications for Windows for displaying and arranging windows.

In general, all windows have cut-and-paste editing capabilities, cursor selection, and drag-and-drop editing capabilities.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

## Pop-up Menus

In many windows, you can click the right mouse button to display a pop-up menu of frequently used commands. The commands available from it depend on the context in which it is displayed, namely, what the pointer is pointing at. For example, if you click while pointing at a toolbar, the pop-up menu shows a list of toolbars that you can toggle on or off and commands to open the Toolbars dialog box and the Customize dialog box.

When you use a pop-up menu, the action is equivalent to selecting the appropriate menu item and clicking with the left mouse button. However, when the pop-up menu is invoked with the mouse pointer over a selection, the selection is not affected.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Application Frame

The Developer Studio application frame is displayed when all files are closed.

The status bar at the bottom of the frame displays the current status of Developer Studio, the current status of an ongoing process, or the current action that you can take.

If you can edit in the currently active window in Developer Studio, the status bar indicates the line and column position of the insertion point, if you are in column select mode, if you are in insert or overwrite mode—toggled with the INS key, or if the file in the currently active window is read-only. It also indicates if the you are currently recording a macro. In addition, you can display a digital clock on the status bar.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Accelerator Editor Window

Use the accelerator editor window to add, delete, change, or browse the accelerator-key assignments in your project.

**Accelerator Table Editor Pop-up Menu**
The accelerator table editor pop-up menu has these commands when you are editing the accelerator table resource in Microsoft Developer Studio:

| Pop-up Menu Commands | Description |
| --- | --- |
| Cut | Removes the selection and puts it on the Clipboard. |
| Copy | Copies the selection and puts it on the Clipboard. |
| Paste | Inserts Clipboard contents at the insertion point. |
| New Accelerator | Creates a new accelerator key. |
| ClassWizard | Edits application classes and ties resources to code. |
| Properties | Displays the property page of the selected accelerator. |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Binary Editor Window

Use the binary data editor window to edit an existing custom resource at the binary level in either hexadecimal or ASCII format.

**Binary Data Editor Pop-up Menu**
The binary data editor pop-up menu has these commands when you are editing or viewing binary data in Microsoft Developer Studio:

| Pop-up Menu Commands | Description |
| --- | --- |
| Cut | Removes the selection and puts it on the Clipboard. |
| Copy | Copies the selection and puts it on the Clipboard. |
| Paste | Inserts Clipboard contents at the insertion point. |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Bitmap Editor Window

The bitmap editor window is used by the graphic editor to edit bitmaps, icons, and cursors, as well as features to support the creation of toolbar bitmaps and the management of icon and cursor images.

**Note**    The bitmap editor window does not have a pop-up menu since the right-mouse button is already used for image editing functions.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Browse Window

Browse windows display information about the symbols in your program. When you build a project, you can specify that the compiler create a file with the information about the symbols in your project. This file has the project's base name and the extension .BSC. You can also browse through any .BSC file created with the BSCMAKE.EXE utility located in the directory with other executables.

You view browse information in browse windows, which have different appearances and different controls depending on the type of information that they are displaying.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Symbol Type Abbreviations in the Browse Window

When you display symbols in the browse window, they are preceded by abbreviations denoting the type of symbol. The browse window uses the following abbreviations:

| Abbreviation | Meaning |
| --- | --- |
| c | class |
| f | function |
| d | data |
| m | macro |
| t | non-class type |
| V | Virtual function or data member |
| S | Static function or data member |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Browse Window: Base Classes and Members

The Base Classes and Members view of the browse window displays the following information:

| Window Element | Function | Action |
| --- | --- | --- |
| Left pane | Displays a graph of derivations. | Click the plus (+) or minus ( – ) sign to expand or contract the graph. Click the node or title to select it and display the corresponding information in the right panes. Double-click a symbol to open the source at the definition. |
| Top | Display | Click |

| | | |
|---|---|---|
| right pane | s member functions and member variables of the class selected in the left pane. | to display the definition and references in the bottom right pane. Double-click to open the source at the definition. |
| Bottom right pane | Displays definitions and references for the item currently selected in either the left or top right panes. | Double-click to open the source at the definition or a specific reference. |
| Push pin | Determines whether or not the window disappears after it loses focus. | Click to push or pull the pin. |
| Help button | Displays help for the window. | Click for help. |
| Filter lists | Filters the browser | Select a filter type |

query to from
display   the
selected lists.
types of
informat
ion.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Browse Window: Call Graph

The Call Graph view of the browse window displays the following information:

| Window Element | Function | Action |
|---|---|---|
| Left pane | Displays a graph of functions that the selected function calls. | Click the plus (+) or minus ( − ) sign to expand or contract the graph. Click the node or title to select it and display the corresponding information in the right pane. Double-click to open the source at the definition or a specific reference. |
| Ri | Displays | Double |

| | | |
|---|---|---|
| ght pane | definitions and references for the item currently selected in the left pane. | -click to open the source at the definition or a specific reference. |
| Push pin | Determines whether the window disappears after it loses focus. | Click to push or pull the pin. |
| Help button | Displays help for the window. | Click for help. |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Browse Window: Callers Graph

The Callers Graph view of the browse window displays the following information:

| Window Element | Function | Action |
|---|---|---|
| Left pane | Displays a graph of functions that call the selected function. | Click the plus (+) or minus (−) sign to expand or contract the graph. Click the node or title to select it and display the corresponding information in the right pane. Double-click to open the source at the definition or a specific |

| | | |
|---|---|---|
| | | reference. |
| Right pane | Displays definitions and references for the item currently selected in the left pane. | Double-click to open the source at the definition or a specific reference. |
| Push pin | Determines whether the window disappears after it loses focus. | Click to push or pull the pin. |
| Help button | Displays help for the window. | Click for help. |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Browse Window: Definitions and References

The Definitions and References view of the browse window displays the following information:

| Window Element | Function | Actions |
|---|---|---|
| Left pane | Displays the selected symbol or a list of the matching symbols if you used a wildcard. | Click to select a symbol from the list. Double-click to open the source at the definition. |
| Right pane | Displays definitions and references for the item currently selected in the left pane. | Double-click the location specified for the definition or reference to open the source at the definition or reference. |
| Push pin | Determines whether the window disappears | Click to push or pull the pin. |

| | | |
|---|---|---|
| | after it loses focus. | |
| Help butt on | Display s help for the window . | Click for help. |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Browse Window: Derived Classes and Members

The Derived Classes and Members view of the browse window displays the following information:

| Window Element | Function | Action |
|---|---|---|
| Left pane | Displays a graph of derivations. | Click the plus (+) or minus ( – ) sign to expand or contract the graph. Click the node or title to select it and display the corresponding information in the right pane. Double-click a symbol to open the source at the definition. |
| Top right | Displays | Double-click |

| | | |
|---|---|---|
| pane | member functions and member variables of the class selected in the left pane. | to open the source at the definition. |
| Bottom right pane | Displays definitions and references for the item currently selected in either the left or top right panes. | Double-click to open the source at the definition or a specific reference. |
| Push pin | Determines whether the window disappears after it loses focus. | Click to push or pull the pin. |
| Help button | Displays help for the window. | Click for help. |
| Filter lists | Filters the browser query to display selected types of information. | Select filter types from the lists. |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Browse Window: File Outline

The File Outline view of the browse window displays the following information:

| Window Element | Function | Actions |
|---|---|---|
| Left pane | Displays the functions and classes in the file by default. | Click to select a symbol from the list. Double-click to open the source at the definition. |
| Right pane | Displays definitions and references for the item currently selected in the left pane. | Double-click the location specified for the definition or reference to open the source at the definition or reference. |
| Push pin | Determines whether the window disappears | Click to push or pull the pin. |

| | | |
|---|---|---|
| | after it loses focus. | |
| Help button | Displays help for the window. | Click for help. |
| Filter buttons | Filters the browser query to display selected types of information. | Click to toggle the filter. |

The left pane displays the following information about symbols:

| Left entries | Middle entry | Right entry |
|---|---|---|
| Applicable filter types from the selected filters and virtual or static member indicator | Symbol name | Type of symbol if the symbol name is ambiguous about type |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Call Stack Window

The Call Stack window lists the function calls that led to the current statement, with the current function on the top of the stack. By default, each call is shown with the parameter types and values passed to it. You can turn the display of parameter types and values off using the pop-up menu commands shown below.

If you double-click on frame (function) in the call stack, the debugger updates the windows as if you were in that frame. If you select a frame in the call stack and press F7, the program executes until it reaches that frame.

You can copy information from the Call Stack window and drag information to another window using drag-and-drop.

**Call Stack Window Pop-up Menu**
When the right mouse button is clicked in the call stack window, the pop-up menu contains the following commands:

| Pop-up Menu Command | Description |
| --- | --- |
| Go to Code | Displays the source code or disassembled object code and debugger windows as if the debugger was in the selected function. |
| Insert/ Remove Breakpoint | Inserts or removes a breakpoint where the cursor is currently located. |
| Enable/ Disable Breakpoint | Toggles the breakpoint where the cursor is currently located. |
| Run to Cursor | Runs to where the cursor is currently located. |
| Parameter | Toggles parameter |

| | |
|---|---|
| Values | value display on (checked) and off (unchecked). |
| Parameter Types | Toggles parameter type display on (checked) and off (unchecked). |
| Hexadecimal Display | Changes the call stack window to hexadecimal display. |
| Docking View | Toggles the state of the window between dockable (checked) and undockable (unchecked). |
| Hide | Closes the Call Stack window. |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Cursor Editor Window

You can edit icon and cursor bitmaps in the graphic editor of Microsoft Developer Studio.

Icons and cursors are like bitmaps, and you edit them in the same ways. But icons and cursors have attributes that distinguish them from bitmaps. For example, each icon or cursor resource can contain multiple images for different display devices.   In addition, a cursor has a "hot spot" — the location Windows NT uses to track its position.

**Note**    The cursor editor window does not have a pop-up menu since the right-mouse button is already used for image editing functions.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Dialog Editor Window

Use the dialog editor window to quickly create dialog boxes, place and arrange controls, and test the finished product. In addition, you can also add OLE controls, import Visual Basic forms to a dialog box resource and save dialogs as templates for future dialog boxes. For easier layout, guides or a grid helps align groups of controls.

**Dialog Editor Pop-up Menu**
The dialog editor pop-up menu has these commands when you are editing or viewing dialog boxes in Microsoft Developer Studio:

| Pop-up Menu Commands | Description |
| --- | --- |
| Cut | Removes the selection and puts it on the Clipboard. |
| Copy | Copies the selection and puts it on the Clipboard. |
| Paste | Inserts Clipboard contents at the insertion point. |
| Insert OLE Control | Inserts an OLE control as a stand-alone control without the wrapper class. |
| Size to Content | Resizes selected controls to fit their caption text. |
| Align Left Edges | Aligns the left edges |

|  |  |
|---|---|
|  | of the selected controls with the dominant control. |
| Align Top Edges | Aligns the top edges of the selected controls with the dominant control. |
| Check Mnemonics | Confirms any duplicate mnemonics. |
| ClassWizard | Edits application classes and ties resources to code. |
| Properties | Displays property page of selected control or dialog box. |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Disassembly Window

Use the Disassembly window to view the assembly-language instructions that the compiler generates for your source code.

**Disassembly Window Pop-up Menu**
When the right mouse button is clicked in the disassembly window, the pop-up menu contains the following commands:

| Pop-up Menu Command | Description |
| --- | --- |
| Copy | Copies the current selection to the clipboard. |
| Go to Source | Displays the source associated with the current selection. |
| Run to Cursor | Treats the cursor location as a temporary breakpoint and runs the program to that point. |
| Set Next Statement | Sets the instruction pointer to the line containing the cursor. |
| Source Annotation | Toggles the display of source annotations between on (checked) and off (unchecked). |
| Docking View | Toggles the state of the window between |

| | dockable (checked) and undockable (unchecked). |
|---|---|
| Close | Closes the Disassembly window. |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Icon Editor Window

You can edit icon and cursor bitmaps in the graphic editor of Microsoft Developer Studio.

Icons and cursors are like bitmaps, and you edit them in the same ways. But icons and cursors have attributes that distinguish them from bitmaps. For example, each icon or cursor resource can contain multiple images for different display devices.   In addition, a cursor has a "hot spot" — the location Windows NT uses to track its position.

**Note**    The icon editor window does not have a pop-up menu since the right-mouse button is already used for image editing functions.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Memory Window

The Memory Window displays memory contents starting at a specified address (0x00010000 by default). A toolbar at the top of the window displays the starting address for the memory display. Edit the value in the toolbar and press return to change the starting address. Use the scrollbar at the side of the window to view other memory locations in the program's address space without changing the starting address of the display.

If you drag a memory address from another window to the Memory window, the Memory window displays the memory contents starting at that address. If you drag a pointer from another window to the Memory window, the Memory window displays the memory contents starting at the address the pointer points to.

**Memory Pop-up Menu**
When the right mouse button is clicked in the memory window, the pop-up menu contains the following commands:

| Pop-up Menu Command | Description |
| --- | --- |
| Copy | Copies the current selection. |
| Byte Format | Toggles display to byte (8-bit hex) format. |
| Short Hex Format | Toggles display to short hex format (four hexademical digits per group). |
| Long Hex Format | Toggles display to long hex format (eight hexademical digits per group). |
| Toolbar | Toggles the state of the Memory window toolbar, which contains the starting memory address, between |

| | |
|---|---|
| | visible (checked) and hidden (unchecked). |
| Docking View | Toggles the window between dockable (checked) and undockable (unchecked). |
| Hide | Closes the Memory window. |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Menu Editor Window

The menu editor window is displayed by Microsoft Developer Studio to edit your program's menu resources.

**Menu Editor Pop-up Menu**
The menu editor pop-up menu has these commands when you are editing the menu resources in Developer Studio

| Pop-up Menu Commands | Description |
| --- | --- |
| Cut | Removes the selection and puts it on the Clipboard. |
| Copy | Copies the selection and puts it on the Clipboard. |
| Paste | Inserts Clipboard contents at the insertion point. |
| View As Popup | Displays the menu resource as a pop-up menu. |
| Check Mnemonics | Confirms any duplicate mnemonics. |
| ClassWizard (Visual C++) | Edits application classes and ties resources to code. |
| Properties | Displays the property page for |

the
selected
item or
resource.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Output Window

The Output window is used to display several types of information. The Output window has a separate tab output from each tool—Build, Debug, Find in Files, Profile, and Source Control. To display the output from a given tool, select the tab at the base of the Output window. You can copy and print information from the Output window.

**Note**   To enable autoscrolling in the Output window, place the cursor on the last line.

If you select the Build tab, the Output window displays progress and error messages from the compiler and linker. The list includes all errors that prevent a program from building, with filename, line number, and error number. To find the source code corresponding to an error, select the error, then click the right mouse button and select Go To Error/Tag from the pop-up menu. If you display the status bar, it gives a summary of the current error.

If you select the Find In Files tab, the Output window displays the result of the most recent Find In Files search.

If you select the Debug tab, the Output window displays messages generated by the debugger.

If you select the Profile tab, the Output window displays information generated by the profiler.

**Output Window Pop-up Menu**
When the right mouse button is clicked in the output window, the pop-up menu contains the following commands:

| Pop-up Menu Commands | Description |
| --- | --- |
| Copy | Copies the selection and puts it on the Clipboard. |
| Clear | Clears the current output page. |
| Go To Error/Tag | Moves to the line containing the current error tag. |
| Docking View | If checked, the Output window is displayed as a dockable window. |
| Hide | Closes the Output window. |

**Note**   If you have installed additional packages that work in conjunction with Microsoft Developer Studio, click the See Also button if it is active to find related topics.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Project Workspace Window

The project workspace window displays relationships in your projects. The information about your workspace is displayed in several panes.

You can access information about elements of your projects from the panes. Clicking a button at the bottom of the workspace window displays the associated pane; each button displays a pop-up label if the pointer rests on it. You can also switch panes using CTRL+PAGE UP and CTRL+PAGE DOWN. Double-clicking a folder in a pane expands the folder to show its contents. Double-clicking a bottom-level item in a folder opens that item in the appropriate viewer: source editor for text files, resource editor for a resource, help topic window for help topics, and so on.

Each pane has a pop-up menu of commands appropriate for the current selection in the pane. You access the pop-up menu by clicking the right mouse button in the pane with the selection.

The FileView pane shows information about the projects that you have created and the relationships among the source files for those projects. You can select the default project configuration to build with the Set Default Configuration command on the Build menu, or with the Set as Default Configuration command on the pop-up menu. The icons also give you information about the files.

The ResourceView pane shows resource types, and under each resource type, the individual resources included in your projects.

The InfoView pane displays the Table of Contents for Books Online. You can select a subset to view by clicking the Select Subset button on the Help Contents toolbar, and you can define a new subset using the Define Subset command on the pop-up menu. You can view a different title using the Select Documentation button on the Help toolbar, or by hitting CTRL+SHIFT+O from within a topic window.

You can hide the workspace window by using the following methods:

- Set the focus to the workspace window and press SHIFT+ESC.
  –or–
- Click the Workspace Window toolbar button.
  –or–
- Choose Hide from the pop-up menu in any workspace window pane.
  –or–
- Move the mouse pointer into the border of any toolbar, click the right mouse button to display the pop-up menu and choose Workspace Window.

If you have hidden the workspace window, you can display it again by using the following methods:

- Click the Workspace Window toolbar button.
  –or–
- Choose Workspace Window from the View menu.
  –or–
- Press ALT+0.
  –or–
- Move the mouse pointer into the border of any toolbar, click the right mouse button to display the pop-up menu and choose Workspace Window.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# FileView Icons

| Icon | Meaning |
| --- | --- |
| {ewcmsdncd,EWGraphic,dev25d0/a"MSDEV400BLD | Developer Studio can use this file in a build, and it is included in the build for this project. |

I.BMP"}

{ewcmsdncd, EWGraphic, dev25d1/a" MSDEV40BLDI

Developer Studio can use this file in a build, but it is not included in the build for this project.

2.BMP"} {ewc msdncd, EWGraphic, dev25d2/a" MSDEV40DEP

Developer Studio uses this file as an explicit dependency in a project.

I.
B
M
P
"}
{ewcmsdncd,
EWGraphic,
dev25d3/a"
MSDEV40NBLD

Developer Studio cannot build this file using the default tools. Files in this category might include documentation or specifications, for instance; you could specify custom tools for them.

.BMP"}

{ewcmsdncd,EWGraphic,dev25d4/a"MSDEV40SPRO

Developer Studio refers to this project as a subproject of the project that contains it. When Developer Studio builds the containing project, it first builds the output of this subproject if it is out of date with respect to its input files.

J.
B
M
P
"}

If you have installed a source code control system that conforms to the Microsoft Common Source
Code Control Interface, the icons also represent some source-code control states. Grayed means
that a file is under source-code control. A check next to the icon for a file under source-code control
indicates that you have the file checked out.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Registers Window

The Registers window displays the names and current values of the native CPU registers and flags. It also displays the floating-point stack. You can change the value of any register or flag in the Registers window while the program is being debugged.

**Registers Window Pop-up Menu**
When the right mouse button is clicked in the registers window, the pop-up menu contains the following commands:

| Pop-up Menu Commands | Description |
| --- | --- |
| Copy | Copies the current selection to the clipboard. |
| Floating Point Registers | Toggles the display floating-point registers between on (checked) and off (unchecked). |
| Docking View | Toggles the state of the window between dockable (checked) and undockable (unchecked). |
| Hide | Closes the Registers window. |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Resource Browser (RC) Window

The resource browser (RC) window displays the resources used by the current project.

**Resource Browser Window Pop-up Menu**
The resource browser window pop-up menu is selection-sensitive. Separate menus are used for resources, resource types, and the resource script. The appropriate menu is chosen based on where the right mouse button is clicked.

When shown for a resource script, the pop-up menu contains these commands:

| Pop-up Menu Command | Description |
| --- | --- |
| Insert | Creates a new resource of any type. |
| Symbols | Browses and edits the symbols in the active file. |
| Set Includes | Edits symbol filename and preprocessor directives. |
| Properties | Displays the property page for the script. |

When shown for a resource type, the pop-up menu contains these commands:

| Pop-up Menu Command | Description |
| --- | --- |
| Insert | Creates a new resource of any type. |
| Insert type | Adds a new resource of selected |

|  |  |
|---|---|
|  | type to script. |
| Import | Import a resource. |

When invoked for an individual resource item, the pop-up menu contains these commands:

| Pop-up Menu Command | Description |
|---|---|
| Insert | Creates a new resource of any type. |
| Insert type | Adds a new resource of selected type to script. |
| Open | Opens the selected resource for editing. |
| Open Binary Data | Opens the selected resource for binary editing. |
| Import | Imports a resource item. |
| Export | Exports the current resource to a new file. |
| Properties | Displays the property page for that resource. |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# String Table Editor Window

You can use the string table editor window to edit a string table resource. A string table resource contains a list of IDs, values and captions for all the strings in your application. An application can have only one string table.

**String Table Editor Pop-up Menu**
The string table editor pop-up menu will have these commands when you are editing the string table resource:

| Pop-up Menu Command | Description |
| --- | --- |
| Cut | Removes the selection and puts it on the Clipboard. |
| Copy | Copies the selection and puts it on the Clipboard. |
| Paste | Inserts Clipboard contents at the insertion point. |
| New String | Creates a new string. |
| Properties | Displays the property page for the string resource. |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Text Editor Window

The text editor window displays text files of any type, such as language source and header files.

**Text Editor Pop-up Menu**
The text editor pop-up menu will have these commands when you are editing or viewing source code in Microsoft Developer Studio:

| Pop-up Menu Command | Description |
| --- | --- |
| Cut | Removes the selection and puts it on the Clipboard. |
| Copy | Copies the selection and puts it on the Clipboard. |
| Paste | Inserts Clipboard contents at the insertion point. |
| Insert/ Remove Breakpoint | Adds a new breakpoint or deletes the old breakpoint at the current line. |
| Enable/ Disable Breakpoint | Toggles the selected breakpoint between active and inactive. |
| Insert File into Project | Displays the dialog to select |

| | |
|---|---|
| | and insert a file into the current project. |
| Open | Opens a file based on the selected text. |
| Go To Definition | Displays the definition of the selected symbol. |
| Go To Reference | Displays the reference to the selected symbol. |
| ClassWizard | Edits application classes and ties resources to code. |
| Toolbar | Displays the Wizard Bar in the text editor window. The Wizard Bar is only available after you have created ClassWizard data in the .CLW file. |
| Properties | Edits properties of the selected item. |

When debugging, the pop-up menu becomes:

| Pop-up Menu Command | Description |
| --- | --- |
| Open | Opens a file based on the selected text. |
| Go To Definition | Displays the definition of the selected symbol. |
| Go To Reference | Displays the reference to the selected symbol. |
| Go To Disassembly | Activates the disassembly window for this instruction. |
| QuickWatch | Displays the value of the selected symbol. |
| Step into Specific Function | Steps into the selected function. For example, if there are three functions on a single line, you can select the third function and step directly into that function. |
| Insert/ | Adds a |

| | |
|---|---|
| Remove Breakpoint | new breakpoint or deletes the old breakpoint at the current line. |
| Enable/ Disable Breakpoint | Toggles the selected breakpoint between active and inactive. |
| Run to Cursor | Runs current program to the line containing the cursor. |
| Set Next Statement | Sets the instruction pointer to the line containing the cursor. |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Text Tool Window

The text tool window is used to add text information to a cursor, bitmap, or icon resource.

The text area of the window contains the text that appears as part of the resource. Initially this area is empty.

Choose the Font button to change the font, style, or size of the cursor font.

**Text Tool Pop-up Menu**
When the right mouse button is clicked in the text tool window, the pop-up menu contains the following commands:

| Pop-up Menu Command | Description |
| --- | --- |
| Output | Shows/ hides this window. |
| Standard | Shows/ hides this toolbar. |
| Edit | Shows/ hides this toolbar. |
| Resource | Shows/ hides this toolbar. |
| Graphics | Shows/ hides this toolbar. |
| Colors | Shows/ hides this toolbar. |
| Text Tool | Shows/ hides this toolbar. |
| Debug | Shows/ hides this toolbar. |
| Browse | Shows/ hides this toolbar. |
| Toolbars | Hides or shows toolbars. |
| Customize | Customizes toolbars. |

**Note**   The text tool window uses the default pop-up menu, which contains a list of windows and toolbars and the Toolbars and Customize commands.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Variables Window

The Variables window contains three tabs:

- The Auto tab displays information about variables used in the current statement and the previous statement.
- The Locals tab displays information about variables local to the current function.
- The "this" tab displays information about the object pointed to by **this**.

Each tab contains a spreadsheet field with three resizable columns. The debugger automatically fills these columns with the type, name, and value of variables appropriate to the tab.

A toolbar located above the tabs contains a drop-down list for specifying the current scope of the variable display. This toolbar can be hidden, or redisplayed, using the pop-up menu.

The Variables window replaces the Locals from previous versions of Visual C++ and adds new functionality.

If the Variables window contains an array, object, or structure variable, a button appears next to the variable name. By clicking on the button, you can expand or contract your view of the variable. The button displays a plus sign (+) when the variable is displayed in contracted form, and a minus sign when it is displayed in expanded form.

The Variables window supports editing. You can cut, copy, or drag information from the Variables window. You can edit the Value column to change the value of a variable while debugging.

**Row and Column Behavior**
To autosize a column to fit its contents, double-click on the vertical divider at the column edge. To size a column manually, drag the right divider to the left or right.

**Note**  Rows are sized to fit the current font and cannot be resized manually. To change the font size, use the Fonts and Colors tab of the Options command from the Tools menu.

**Auto Tab**
The Auto Tab displays information about variables from the current statement and the previous statement. Variables appear in alphabetical order. If a statement spans multiple lines, the Auto tab displays variables from the lines corresponding to that statement, up to a 10-line limit.

**Locals Tab**
The Locals tab displays the names, values, and types of all local variables in the current function. As you trace through a program, new variables come into scope.

**this Tab**
This tab displays type, name, and value information about the object pointed to by the pointer **this**. All base classes of the object are automatically expanded.

**Variables Window Pop-up Menu**
When you click the right mouse button in the Variables window, a pop-up menu appears, containing the following commands:

| Pop-up Menu Command | Description |
| --- | --- |
| Copy | Copies the current selection to the toolbar. |

| | |
|---|---|
| Hexadecimal Display | Toggles the display between hexadecimal (checked) and decimal (unchecked). |
| Type Column | Toggles the Type column between displayed (checked) and hidden (unchecked). |
| Toolbar | Toggles the toolbar between displayed (checked) and hidden (unchecked). |
| Docking View | If checked, the Variables window is displayed as a dockable window. |
| Hide | Closes the Variables window. |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Version Information Editor Window

You can use the version information editor window to create and maintain a version information resource for an application. Version information consists of company and product identification, a product release number and copyright and trademark notification.

**Version Information Editor Pop-up Menu**
The version information editor pop-up menu has these commands when you are editing the version information resource:

| Pop-up Menu Command | Description |
| --- | --- |
| New String Block | Adds a new string information block. |
| Delete String Block | Deletes the string information block. |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Watch Window

The Watch window contains four tabs: Watch1, Watch2, Watch3, and Watch4. Each tab contains a spreadsheet field that displays variable information. You can enter variable names into the Name column of this field; the debugger fills in the Type and Value columns with the corresponding information as the program runs.

**Note**    You can enter expressions, as well as variable names, in the Watch window's Names columns. The debugger evaluates the expression continuously as your program executes.

If the Watch window contains an array, object, or structure variable, a button appears next to the variable name. By clicking on the button, you can expand or contract your view of the variable. The button displays a plus sign (+) when the variable is displayed in contracted form, and a minus sign when it is displayed in expanded form.

The Watch window supports editing functions. You can cut or copy information from the Watch window. You can cut or copy a variable from another window and paste it into the Watch window or drag it in using drag-and-drop. You can edit the Value column to change the value of a variable while debugging.

**Row and Column Behavior**
To autosize a column to fit its contents, double-click on the vertical divider at the column edge. To manually size a column, drag the vertical divider at the edge of the column.

Note    Rows are sized to fit the current font. To change the font size, use <u>the Fonts and Colors tab of the Options command from the Tools menu.</u>

**Watch Window Pop-up Menu**
When you click the right mouse button in the Watch window, a pop-up menu appears, containing the following commands:

| Pop-up Menu Command | Description |
|---|---|
| Copy | Copy the current selection. |
| Paste | Paste text into the current selection. |
| Hexadecimal Display | Toggles the values display between hexadecimal (checked) and decimal (unchecked). |
| Docking View | Toggles the window between dockable |

|      | (checked) and undockable (unchecked). |
|------|---------------------------------------|
| Hide | Closes the window.                    |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Help Topic Window

The Help Topic window displays a topic from the InfoViewer help system.

This window offers a toolbar for navigating within the help system: With this toolbar, you can jump to the previous or next topic in the help system, or synchronize the Table of Contents (in the help pane of the Project Workspace window) to show you where the current topic resides within the help system.

## Help Topic Window Pop-up Menu

| Pop-up Menu Command | Description |
|---|---|
| Copy | Copy the current selection. |
| Select All | Select the entire topic. |
| Search... | Display the dialog box for searching the help system. |
| Annotation | Open the annotation window for this topic. |
| Add Bookmark... | Add a bookmark to the current location. |
| Options | Display the Options dialog open to the Help tab. |
| Highlight | Toggle highlighting of search hits resulting from a full-text search. |

| | |
|---|---|
| Print Topic | Print the current topic. |
| Show Toolbar | Toggle display of the topic window toolbar. |
| Show Title | Toggle display of the topic title. |
| Docking View | Toggle the window between dockable (checked) and undockable (unchecked). |
| Close | Close the help topic window. |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Default Pop-up Menu

In many cases, the pop-up menu available simply lists available windows and toolbars to open and the commands to customize toolbars. The default pop-up menu contains the following commands:

| Pop-up Menu Command | Description |
| --- | --- |
| Output | Shows/ hides this window. |
| Watch (if debugging) | Shows/ hides this window. |
| Locals (if debugging) | Shows/ hides this window. |
| Registers (if debugging) | Shows/ hides this window. |
| Memory (if debugging) | Shows/ hides this window. |
| Call Stack (if debugging) | Shows/ hides this window. |
| Disassembly (if debugging) | Shows/ hides this window. |
| Standard | Shows/ hides this toolbar. |
| Edit | Shows/ hides this toolbar. |
| Resource | Shows/ hides this toolbar. |
| Debug | Shows/ hides this toolbar. |
| Browse | Shows/ |

| | |
|---|---|
| | hides this toolbar. |
| Toolbars | Hides or shows toolbars. |
| Customize | Customizes toolbars. |

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Toolbar Editor Window

You can edit toolbar resources in the toolbar editor window. This toolbar editor also converts bitmaps into toolbar resources. There is also a menu command on the Image menu to switch between the graphic editor and the toolbar editor.

**Note**    The toolbar editor window does not have a pop-up menu since the right-mouse button is already used for image editing functions.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# ClassWizard: Adding a Class

Use the Adding a Class dialog box to specify whether you want to associate a new or an existing class with a dialog box, menu, toolbar, or accelerator resource. The Adding a Class dialog box is displayed if you invoke ClassWizard when a resource editor is open and the resource is not yet associated with a class.

**Create a new class**
Choose this option to create an entirely new class.

**Import an existing class**
Choose this option to import a class into the ClassWizard database. This allows you to reuse a class that you created in a different project.

**Select an existing class**
Choose this option to select an existing class from the ClassWizard database.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# ClassWizard: Select Class

Use the Select a Class dialog box to associate a new dialog box, menu, toolbar, or accelerator resource with an existing class. The association will cause ClassWizard to make the resource command IDs available for mapping when the class is selected in the Message tab or in WizardBar. You can select any class from the displayed list of classes.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# ClassWizard: Import Class Information

Use the Import Class Information dialog box to import an existing class, from another project, into the ClassWizard database (.CLW).

**Class name**
The name of the class that you want to import.

**Header file**
The name of the file that contains header information for the imported class.

**Implementation file**
The name of the file that contains implementation code for the imported class.

**Browse**
Opens the Browse Files dialog box, which you can use to select a directory that has the header and implementation files that contain the class source code.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

## Component Gallery: Change Icon

Use the Change Icon dialog box to add an icon for a component, or to change an component's current icon.

Under Windows 95, you get help on the following controls by selecting the control and then pressing F1:

**Look in**
**File name**
**Files of type**


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Create New Class

Use the Create New Class dialog box to add new message-handling classes to your application.

**Name**
The name of the class you want to add to your application.

**Base Class**
The base class from which to derive the new class.

**Change**
Opens the Change Files dialog box, which you use to specify names for the new class's header and implementation files.

**Dialog ID**
Allows selection of a resource ID. This option is available only for classes that must be associated with a resource ID, such as those associated with dialog boxes.

**None**
OLE automation not selected.

**Automation**
Exposes your class to OLE Automation clients. This enables objects of this class to be accessed by automation clients, such as Microsoft Visual Basic and Microsoft Excel. This option is available only for some classes.

**Createable by Type ID**
Similar to "Automation" above, but adds the ability for automation clients to create your automation objects directly. Use the text box to specify the name that Automation clients use to request an object of your class. This option is available only for some classes.

**Add to Component Gallery**
Adds the new class, and any associated resource, to the Component Gallery.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# ClassWizard: Change Files

Use the Change Files dialog box to name the header (.H) and implementation (.CPP) files in which the new class will be generated.

**Header File**
The name of the .H file in which the class will be declared.

**Implementation File**
The name of the .CPP file in which the class will be defined.

**Browse**
These buttons open the Browse Files dialog box. Use this dialog box to select existing .H and .CPP files in which the class declaration and definition will be generated. In addition, you can use the browse button to change the directory in which the new files are created.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Add Event

Use the Add Event dialog box to add an OLE event that the object will send.

**External Name**
The name used by automation clients to request an event from your class.

**Internal Name**
The name of the member function that sends the event.

**Stock**
Events, such as button clicks, that the Control Development Kit defines.

**Custom**
Events that you define for a control.

**Name**
The name of a parameter to this event. You may add multiple parameters. To add a parameter, double-click the first empty row under the Name label.

**Type**
The type of the parameter. To specify a type, double-click the first empty row under the Type label and then select a type from the drop-down list.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# ClassWizard: Add Member Function

Use the Add Member Function dialog box to specify the name of a message-handling member function that you add to a dialog class.

**Member Function Name**
The name of the member function you want to add to the dialog class.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# ClassView: Add Member Function

Use the Add Member Function dialog box to specify the name of a member function. You can easily add member functions that do not handle messages.

**Note**    If you want to add a message-handler function for user-interface objects, you should use ClassWizard. ClassWizard is specifically designed to work with message-handling functions for MFC user-interface classes.

**Function Type**
Enter the type that the member function returns.

**Function Declaration**
Enter the name of the member function followed by a parentheses-enclosed list of the names and types of any formal parameters.

**Access**
Choose a keyword to control access to the member function.

**Static**
If you want a static function, select the static check box.

**Virtual**
If you want a virtual function, select the virtual check box.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# ClassWizard: Add Member Variable

Use the Add Member Variable dialog box to specify the name, properties, and type of a member variable that you add to a dialog class.

**Member variable name**
The name of the new variable. By default, ClassWizard provides the m_ prefix to identify it as a member variable.

**Category**
Specifies whether this variable is a Value variable or a Control variable.

**Variable Type**
Lists available types of variables.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# ClassView: Add Member Variable

Use the Add Member Variable dialog box to specify the type, name, and properties of a member variable you add from ClassView.

**Note**   If you want to add a member variable and use the MFC data exchange and data validation features, you should use ClassWizard. ClassWizard is specifically designed to work with the data maps of MFC user-interface classes.

**Variable Type**
Enter the type of the variable.

**Variable Declaration**
Enter the variable name.

**Access**
Choose an access specifier for the variable from the Access group of options.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Add Method

Use the Add Method dialog box to add OLE Automation methods to your class. Other classes use these methods to make requests of your class. ClassWizard automatically updates the dispatch map of your class when you add or delete methods.

**External Name**
The name of the new method. Automation clients use this name to make requests of your class.

**Internal Name**
The name of the new method. This is the name of the class member function that implements the exposed method. This can differ from the external name.

**Return Type**
The return type of the method.

**Stock**
Indicates use of the default implementation of this property.

**Custom**
Indicates that you intend to provide your own implementation of this property.

**Name**
The name of a parameter to this method. You may add multiple parameters. To add a parameter, double-click the first empty row under the Name label.

**Type**
The type of the parameter. To specify a type, double-click the first empty row under the Type label and then select a type from the drop-down list.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Add Property

Use the Add Property dialog box to add OLE Automation properties to your class. Properties are attributes of your class that OLE Automation clients can change. ClassWizard automatically updates the dispatch map of your class when you add or delete properties.

**External name**
The name of the new property. Automation clients use this name to change the property. Instead of defining a new property, you can select from a set of predefined properties in the list box.

**Type**
Type is always available and specifies the type of the property.

**Variable name**
The C++ class data member name associated with the property. This option is available only when you select the Member Variable implementation method.

**Notification function**
The name of the user-written function called whenever the property value is changed. This option is available only when you select the Member Variable implementation method.

**Stock**
Indicates use of the default implementation of this method. Stock methods are available only for OLE controls.

**Member variable**
A type of implementation that enables direct access to the member variable. This is commonly used for properties that do not affect the user interface when changed.

**Get/Set methods**
A type of implementation that enables controlled access to the property. Use this type of implementation when you need to know when the value changes or when a property is calculated. This implementation is commonly used for properties that will affect the user interface when changed.

**Name**
The name of a parameter to this property. You may add multiple parameters. To add a parameter, double-click the first empty row under the Name label.

**Type**
The type of the parameter. To specify a type, double-click the first empty row under the Type label and then select a type from the drop-down list.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Advanced Options

The Advanced Options dialog box allows you to specify specific options.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Document Template Strings

Use the Document Template Strings tab in the Advanced Options dialog box to specify the filenames and extensions that will identify your application.

**File extension**
The file extension associated with a document created by your application. Entering a file extension allows the Windows 95 Explorer to print your application's documents, without launching your application, when they are dropped on a printer icon.

**File type ID**
This ID is used to label your document type in the system registry.

**Language**
This selection shows the language in which strings are displayed in the edit boxes of the Localized Strings control group.

**Main frame caption**
The name displayed in the title bar of your application's main frame window.

**Doc type name**
The filename associated with the selected class. This option is available only if the selected class is derived from class **CDocument**.

**Filter Name**
The string that appears in the List Files of Type list box in the Open and Save As dialog boxes. This field does nothing unless you type a file extension in the File Extension edit box.

**File new name (OLE short name)**
The name that appears in the File New dialog box if there is more than one new document template. If your application is an OLE server, this name is used as the short name of your OLE object.

**File type name (OLE long name)**
If your application is an OLE server, this name is used as the long name of your OLE object. It is also used as the file type name in the system registry.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Windows Styles

Use the Windows Styles tab in the Advanced Options dialog box to specify the caption of your application's main frame window and the styles of your main frame and MDI windows.

**Use split window**
Enables your application's windows to use a splitter bar. The splitter bar will split the application's main views. In an MDI application, the MDI child frame's client window is a splitter window, and in an SDI application, the main frame's client window is a splitter window.

**Thick frame**
Creates a window that has a sizing border.

**Minimize box**
Specifies that the main frame window include a minimize box. This is the default option.

**Maximize box**
Specifies that the main frame window include a maximize box. This is the default option.

**System menu**
Specifies that the main frame window include a system menu. This is the default option.

**Minimized**
Specifies that the main frame window open as an icon.

**Maximized**
Specifies that the main frame window open to the full size of the display.

**Thick frame**
Specifies that the frame of all MDI child windows have a sizing border.

**Minimize box**
Specifies that MDI child windows include a minimize box. This is the default option.

**Maximize box**
Specifies that MDI child windows include a maximize box. This is the default option.

**Minimized**
Specifies that MDI child windows open as icons.

**Maximized**
Specifies that MDI child windows open maximized.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Macintosh-Specific

Use the Macintosh-Specific tab in the Advanced Options dialog box to specify options for your Macintosh application.

**Application sgnature**
An ID that the Finder uses to locate a document or application in the file system.

**Doc file type**
A file attribute that the finder uses to determine which application to use to interpret the associated file's content.

**Filter name**
A string, such as "Text files", which is displayed in the combo box found on the File Open and File Save As dialogs. This string reports the document file type to the user as follows:

- When opening a file, reports the type of files to display in the File Name box.
- When saving a file, reports what type of files to display in the File Name box.

The Document Filter Name differs from the Document File Type in that it is displayed to the user whereas the Document File Type is used by the Finder.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Confirm Classes

Use the Confirm Classes dialog box to select an exported class or classes from the current type library.

**New Dispatch Classes**
Lists the external name of classes described in the type-library file.

**Class Name**
The name of the dispatch class that you want to import.

**Base Class**
The base class from which the dispatch class is inherited.

**Header File**
The name of the file to contain header information for the imported dispatch class.

**Implementation File**
The name of the file to contain implementation code for the imported dispatch class.

**Browse**
Opens the Browse Files dialog box, which you can use to select a directory that contains the header and implementation files where the class source code can be found.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Connect Network Drive

Use this dialog box to connect to shared directories on the network.

**Drive**
Displays the first available drive letter for the connection. You can select another drive letter.

**Path**
Specifies the network path for the connection. A network path consists of   a computer name followed by the name of a shared directory, such as \\STRAD\PUBLIC. You can:

- Select a previous path from the path box.
- Select a new network path from the Shared Directories list.
- Type the name of a computer and a shared directory into the Path box.

**Connect As**
By default, you are connected under the user name you used to log on.

To connect under a user name other than your logon name, type the user name in the Connect As box.

On a Microsoft Windows-based network, you can connect using an account on a different domain by specifying the domain and user name for the account. Separate the domain from the user name with a backslash; for example, PROJECTS\CHRISBUR.

**Reconnect at Logon**
Clear this check box if you do not want to connect to the shared directory each time you log on.

**Expand by Default**
By default, the list in the Shared Directories box expands to display the computers in your computer's domain or workgroup. You may want to switch this off if you are connecting over a shared network, such as connecting through Remote Access Service. To stop automatic expansion of the Shared Directories list, clear the Expand by Default check box.

**Shared Directories**
Shows networks, domains, workgroups, computers, and shared directories. Double-click an item to expand the list. Selecting a shared directory places it in the Path box.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

## Data Binding

Enables the container to determine what level of data binding the OLE control supports. The data binding notifications maintain a connection between a control and a data source.

**Bindable Property**
Specifies that this is a bindable property. The control sends an OnChanged notification after this property is changed.

**Sends OnRequestEdit**
Specifies that the selected control sends an OnRequestEdit notification before the property is changed.

**Visible to End User**
Specifies that the container should display this property to the end-user as a bindable property.

**Default Bindable Property**
Specifies that this property is the default bindable property.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Component Gallery: Import Component

Use this dialog box to import one or more components into the Component Gallery. This is how you add OLE controls and other components to the Component Gallery.

**Copy to Gallery Directory**
Copies the imported component to the MSDEV\TEMPLATE directory.

Under Windows 95, you get help on the following controls by selecting the control and then pressing F1:

**Look in**
**File name**
**Files of type**


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Component Gallery: Export Component

Use this dialog box to export a component to a file so that you can share it with others. You can only export a component that you have created using the Create New Class dialog box.

Under Windows 95, you get help on the following controls by selecting the control and then pressing F1:

**Look in**
**File name**
**Save as type**


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# MFC AppWizard: Step 1

Use this dialog box to specify the application architecture you want to create.

**Single document**
Creates an application that allows a user to work with just one document at a time. Windows Notepad is an example of a single-document interface.

**Multiple documents**
Creates an application that allows a user to open multiple documents, each with its own window. Windows File Manager is an example of a multiple-document interface.

**Dialog based**
Creates an application that is based on a dialog-template resource—that is, it is created using a resouce compiler **DIALOG** statement.

**What language would you like your resources in?**
Selects the language you want to use for your resources. The drop-down list displays only languages whose DLLs are available on your system.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# MFC AppWizard: Step 1 of 1

Use this dialog box to select the functionality you want to include in your DLL project.

**Regular DLL with MFC statically linked**
Links your application to the static MFC library at build time. Both Win32 and MFC applications can call functions in your DLL.

**Regular DLL using shared MFC DLL**
Links your application to the MFC library at run time. This reduces the disk and memory requirements of your application if it is composed of multiple executable files that all use the MFC library. Both Win32 and MFC applications can call functions in your DLL.

**MFC Extension DLL (using shared MFC DLL)**
Your application makes calls to the MFC library at run time. This reduces the disk and memory requirements of your application if it is composed of multiple executable files that all use the MFC library. Only MFC applications can call functions in your DLL.

**OLE automation**
OLE Automation makes it possible for your application to manipulate objects implemented in another application, or to "expose" objects so they can be manipulated.

**Windows Sockets**
Windows sockets allows you to write applications that communicate over TCP/IP networks.

**Yes, please**
AppWizard generates and inserts comments in the source files that guide you in writing your program. These comments include indicators where you need to add your own code. This is a default option.

**No, thank you**
AppWizard does not insert comments in the source files it generates.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# MFC AppWizard: Step 2 of 4

Use this dialog box to select the application options you want to include when you choose to create a dialog-based application.

**About box**
AppWizard generates code for a message box called the About Box that displays the software version of the application. This is a default option.

**Context sensitive Help**
AppWizard generates a set of help files that are used to provide context-sensitive Help. Help support requires the help compiler. If you do not have the help compiler, you can install it by re-running Setup.

**3D controls**
Creates a dialog box with three-dimensional shading. This is the default option.

**OLE automation**
Allows your application to manipulate objects implemented in another application, or to "expose" objects so they can be manipulated.

**OLE controls**
Allows your application to use OLE Controls.

**Note**    If you do not choose this option and, at a later time, want to insert OLE controls into your project, you must add a call to **AfxEnableControlContainer** in your application's **InitInstance** member function.

**Windows Sockets**
Allows you to write applications that communicate over TCP/IP networks.

**Please enter a title for your dialog**
Type the name of your dialog box in the text box. By default the dialog title is the same as the name of the project.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# MFC AppWizard: Step 3 of 4

Use this dialog box to select the project options you want to include when you choose to create a dialog-based application.

**Yes, please**
AppWizard generates and inserts comments in the source files that guide you in writing your program. These comments include indicators where you need to add your own code. This is the default option.

**No, thank you**
AppWizard does not insert comments in the source files it generates.

**As a shared DLL**
Links MFC to your application as a shared DLL. Your application makes calls to the MFC library at run time. This reduces the disk and memory requirements of your application if it is composed of multiple executable files that all use the MFC library.

**As a statically linked library**
Links your application to the static MFC library at build time.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# MFC AppWizard: Step 4 of 4

Use this dialog box to change the new or base class names in your dialog-based application.

**AppWizard creates the following classes for you**
Lists the new classes that AppWizard will generate.

**Class name**
The name of the class that you have selected in the New Classes list box. You can change the name of the class by typing in the Class Name text box.

**Base class**
The class from which the selected class in the New Classes list box is derived.

**Header file**
The name of the header file associated with the selected class.

**Implementation file**
The name of the source code file associated with the selected class.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# MFC AppWizard: Step 2 of 6

Use this dialog box to select database support options for your application.

**None**
The application that AppWizard creates will not include the AFXDB.H header file, which provides database support. This is the default option.

**Header files only**
The application that AppWizard creates includes the AFXDB.H header file, providing basic database support. You can create recordsets and use them to examine and update records.

**Database view without file support**
The application that AppWizard creates has a **CRecordView**-derived class as its view class. This class is associated with a **CRecordset**-derived class, which AppWizard also creates for you. This option gives you a form-based application in which the record view is used to view and update records via its recordset. The application does not support document serialization, since most database applications operate on a per-record basis rather than on a per-file basis.

If you choose to include a database view, you must specify the source of the data.

**Database view with file support**
The application that AppWizard creates has a **CRecordView**-derived class as its view class. This class is associated with a **CRecordset**-derived class, which AppWizard also creates for you. This gives you a form-based application in which the record view is used to view and update records via its recordset.

This application supports document serialization, which you can use, for example, to update a user profile file. Database applications typically operate on a per-record basis rather than on a per-file basis and so do not need serialization. But you may have a special use for it.

If you choose to include a database view, you must specify the source of the data.

**Data Source**
Opens the Database Options dialog box, which you use to specify the database files for your application. This option is available only if you choose to include a database view in your application.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# MFC AppWizard: Step 3 of 6

Use this dialog box to select an option for OLE support in your application.

**None**
Select this option if you do not want OLE support. This is the default option.

**Container**
Select this option if you want your application to contain linked and embedded objects.

**Mini-server**
Select this option if you want your application to have the ability to create and manage compound document objects. Note that mini-servers cannot run stand-alone and only support embedded items.

**Full-server**
Select this option if you want your application to have the ability to create and manage compound document objects. Full-servers are able to run stand-alone and support both linked and embedded items.

**Both container and server**
Select this option if you want your application to be both a container and a server. A container is an application that can incorporate embedded or linked items into its own documents. An server is an application that can create OLE items for use by container applications.

**ActiveX document server**
Select this option if you want your application to be an ActiveX document server (DocObject server). This option is available only if your application is a server, and you have selected one of the server options: **Mini-server**, **Full-server**, or **Both container and server**.

**Yes, please**
Select this option to serialize your OLE container application's documents using the OLE compound-file format. The compound-file format stores a document that contains one or more OLE objects to one file and still allows access to the individual OLE objects' files.

**No, thank you**
Select this option to not serialize your OLE container application's documents using the OLE compound-file format.

**OLE automation**
Select this option if you want to expose your application to OLE Automation clients. Selecting this option allows your application to be accessed by other Automation clients, such as Microsoft Excel.

**OLE controls**
Select this option if you want your application to use OLE Controls.

**Note**    If you do not choose this option and, at a later time, want to insert OLE controls into your project, you must add a call to **AfxEnableControlContainer** in your application's **InitInstance** member function.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# MFC AppWizard: Step 4 of 6

Use this dialog box to select the functionality you want to include in your application.

**Docking toolbar**
Specifies whether the toolbar that AppWizard generates can be moved from its present location and placed along one of the borders of your application window. By default, the toolbar is dockable and contains buttons for creating a new document, opening and saving document files, cutting, copying, pasting, printing, displaying the About dialog box, and invoking Help.

**Initial status bar**
Specifies whether your application has a status bar. The status bar contains automatic indicators for the keyboard's CAPS LOCK, NUM LOCK, and SCROLL LOCK keys and a message line that displays help strings for menu commands and toolbar buttons. Enabling this option also adds menu commands to display or hide the toolbar and status bar. This is the default option.

**Printing and print preview**
AppWizard generates the code to handle print, print setup, and print preview commands by calling member functions in the **CView** class from the MFC library. It also adds commands for these functions to the application's menu. This is the default option.

**Context-sensitive Help**
AppWizard generates a set of help files that are used to provide context-sensitive help. Help support requires the help compiler.

**3D controls**
Specifies whether the visual interface of the application has three-dimensional shading. This is the default option.

**MAPI (Messaging API)**
Allows you to write an application that creates, manipulates, transfers, and stores mail messages.

**Windows Sockets**
Allows you to write an application that communicates over TCP/IP networks.

**How many files would you like on your recent file list?**
Specifies the number of files to be listed on the "most recently used" list. The default number is 4.

**Advanced**
Opens the Advanced Options dialog box, which you use to specify options for document template strings and frame characteristics.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# MFC AppWizard: Step 5 of 6

Use this dialog box to select the functionality you want to include in your application.

**Yes, please**
AppWizard generates and inserts comments in the source files that guide you in writing your program. These comments include indicators where you need to add your own code. This is the default option.

**No, thank you**
AppWizard does not insert comments in the source files it generates.

**As a shared DLL**
Links MFC to your application as a shared DLL. Your application makes calls to the MFC library at run time. This reduces the disk and memory requirements of your application if it is composed of multiple executable files that all use the MFC library.

**As a statically linked library**
Links your application to the static MFC library at build time.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# MFC AppWizard: Step 6 of 6

Use this dialog box to change the new or base class names in your application.

**AppWizard creates the following classes for you**
A list of the new classes that AppWizard will generate.

**Class name**
The name of the class that you have selected in the New Classes list box. You can change the name of the class by typing a new name in the class name text box.

**Base class**
The class from which the selected class in the New Classes list box is derived.

**Header file**
The name of the header file associated with the selected class.

**Implementation file**
The name of the source code file associated with the selected class.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# MFC ClassWizard: Class Info

Use the Class Info tab in the MFC ClassWizard dialog box to browse and edit the characteristics of a selected class.

When you select a class, you see two kinds of information: read-only and editable. The Base Class and Resource information are read-only.

**Project**
The name of a project. You can select from projects in the list box.

**Class Name**
The name of a class in your application. You can select from classes in the list box.

**Header**
The name of the class's header file. (This information is displayed only and cannot be modified in the dialog box.)

**Implementation**
The name of the class's implementation file. (This information is displayed only and cannot be modified in the dialog box.)

**Base Class**
The name of a class from which the selected class is derived. (This information is displayed only and cannot be modified in the dialog box.)

**Resource**
The name of the resource associated with the selected class, if it is a dialog class. (This information is displayed only and cannot be modified in the dialog box.)

**Message Filter**
The message filter currently in effect for the class you have selected on this tab.

The Messages box on the Message Maps tab initially lists the most appropriate messages for your class. You can change the messages displayed by selecting a different message filter in the Message Filter box. Available filters are categorized by the type of window or dialog box they pertain to.

**Foreign Class**
Name of the foreign class associated with the class you have selected on this tab. The value is "<None>" unless the selected class is a dialog, form view, or record view class.

**Foreign Variable**
Generally used for database classes. When you edit a RecordView, you can select the associated Recordset class.

**Add Class**
Allows you to add a class to your project. You can create an entirely new class, import an existing class into the ClassWizard database, or import a class from an OLE type library.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# MFC ClassWizard: Member Variables

Use the Member Variables tab in the MFC ClassWizard dialog box to add or edit member variables associated with:

- Dialog box, form view, or record view controls.
- Recordset fields.

You can add new variables and edit or delete existing variables. Member variables created with ClassWizard use the Microsoft Foundation Class Library's built-in routines for dialog data exchange and validation (DDX/DDV). Recordset member variables also use built-in routines for record field exchange (RFX) between the recordset's fields and the columns of a table on a data source.

**Tip**　To work with a member variable, you should select a class name and then select a control ID or column name. You can then choose Add Variable or Delete Variable, depending on which is available for your selected item.

**Project**
The name of a project. You can select from projects in the list box.

**Class name**
The name of a class in your application. You can select from the classes in the list box.

**Control IDs**
The list of controls in the dialog box that you can map to member variables. The Type column displays the type for the member variable. The Member column displays the name of the member variable. The Control IDs box is available only when a dialog, form view, or record view class is selected. The Control IDs box is filled only for classes that have a data map: dialog, form view, record view, or recordset.

**Type**
The storage type for this variable.

**Member**
The name of the member variable.

**Column Names**
A list of all column names currently bound to recordset field data members. The names appear in the list as they appear in the table on the data source. Column Names is only available when a recordset class is selected. The Column Names box is filled only for classes that have a data map: dialog, form view, record view, or recordset.

Select a column name and choose Add Variable to bind the column name to a recordset member variable.

**Maximum Characters**
If a member variable is type **CString**, you can specify the maximum number of characters that the user can type in a text box control. Otherwise, this box does not appear. Maximum Characters is available only for variables of type **CString**.

**Maximum Value**
Specifies the maximum value that the user can type in a text box control. Maximum Value is only available for numeric type variables, such as **int**, **UINT**, or **float**. Otherwise, this box does not appear.

**Minimum Value**
Specifies the minimum value that the user can type in a text box control. Minimum Value is only available for variables of a numeric type, such as **int**, **UINT**, or **float**. Otherwise, this box does not appear.

**Add Class**

Allows you to add a class to your project. You can create an entirely new class, import an existing class into the ClassWizard database, or import a class from an OLE type library.

**Add Variable**
Opens the Add Member Variable dialog box, which you use to add a member variable for the selected control.

**Delete Variable**
Deletes the selected member variable. Delete Variable is only available if one or more of the items in Control IDs/Column Names is already associated with a member variable.

**Update Columns**
Opens the Database Options dialog box, where you can select the data source containing the table associated with your recordset. After you select a data source, open the Tables dialog box to select the table to use. Update Columns is only available for recordset classes.

**Bind All**
Binds all unbound recordset field data members to the corresponding columns in a table on the data source. By default, AppWizard and ClassWizard bind all columns, so you seldom need to use Bind All. If you have unbound some or all columns (by deleting their associated data members), you can rebind them all with Bind All. Bind All is only available for recordset classes.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# MFC ClassWizard: Message Maps

Use the Message Maps tab in the MFC ClassWizard dialog box to browse messages or control notifications associated with each object and to create appropriate handler routines.

**Project**
The name of a project. You can select from projects in the list box.

**Class Name**
The name of a class in your application. You can select from the classes in the list box.

**Object IDs**
The IDs that can generate messages, such as menu items, dialog box controls, and so on. The first entry in this list is always the name of the current class.

**Messages**
The messages that the selected object in the Object IDs list can handle. You can display Windows messages the window can receive by selecting the name of a class associated with a window in the Object IDs list. Messages in bold already have message-handler functions. Messages preceeded by an equal sign (=) are the reflected messages of control classes. Reflected messages allow objects of control classes to handle their own messages.

**Member Functions**
The member functions of the selected class. These member functions are message-handler functions or MFC virtual functions. Items marked "V" are virtual functions, and items marked "W" are Windows messages.

**Add Class**
Allows you to add a class to your project. You can create an entirely new class, import an existing class into the ClassWizard database, or import a class from an OLE type library.

**Add Function**
Adds a member function to the Member Functions list box.

**Delete Function**
Deletes the member function declaration from the header file and the function reference from the message map for the selected function.

**Edit Code**
Opens an editor window with the insertion point at the selected member function. Here you insert code for the actions that a member function takes when its corresponding object receives a message. This code defines the message handler for that object.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# MFC ClassWizard: OLE Automation

Use the OLE Automation tab in the MFC ClassWizard dialog box to create and expand the OLE Automation capabilities of your application. You can:

• Add classes that support OLE Automation.
• Add methods and properties to your classes that already support OLE Automation.
• Create a C++ class for another OLE Automation object on your system, such as Microsoft Word for Windows.

**Project**
The name of a project. You can select from projects in the list box.

**Class Name**
The name of a class in your application. You can select from classes in the list box.

**External names**
The names of the methods and properties that you have added to the OLE class. These are the names that objects of this class expose to automation clients, such as Microsoft Visual Basic and Microsoft Excel.

**Implementation**
Shows how the method or property that is selected in the External names box is implemented in your C++ class: "S" indicates stock property, "C" indicates custom property, "M" indicates method, and the item in bold typeface is the default property.

**Add Class**
Allows you to add a class to your project. You can create an entirely new class, import an existing class into the ClassWizard database, or import a class from an OLE type library.

**Add Method**
Displays the Add Method dialog box**,** which you use to add new OLE Automation methods to your class. Other classes use methods to make requests of your class. ClassWizard automatically updates the dispatch map of your class when you add or delete methods. Add Method is only available when you select a class that supports OLE Automation.

**Add Property**
Displays the Add Property dialog box, which you use to add new OLE Automation properties to your class. Properties are attributes of your class that can be changed by OLE Automation clients. ClassWizard automatically updates the dispatch map of your class when you add or delete properties. Add Property is only available when you select a class that supports OLE Automation.

**Delete**
Deletes the name of the method or property that is currently selected in the External names list.

**Note**    If this property was implemented with the Get/Set type of implementation, you must manually delete the Get and Set member functions from your implementation file.

**Edit Code**
Opens an editor window with the selected property or method. You insert code at this point for the actions that a member function takes when its corresponding object receives a message. This code defines the automation handler for that object.

**Data Binding**
Displays the Data Binding dialog box, which you use to specify the level of data binding the OLE control supports.

**Default Property**
Makes the selected property the default property for this OLE object.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# MFC ClassWizard: OLE Events

Use the OLE Events tab to define the OLE events your object supports.

**Project**
The name of a project. You can select from projects in the list box.

**Class Name**
The name of a class in your application. You can select from the classes in the list box.

**External names**
The names of the OLE class's methods and properties. These are the names that objects of this class expose to automation clients, such as Microsoft Visual Basic and Microsoft Excel.

**Implementation**
Shows how the method or property that is selected in the External names box is implemented in your C++ class: "S" indicates stock property, "C" indicates custom property, "M" indicates method, and the item in bold typeface is the default property.

**Add Class**
Allows you to add a class to your project. You can create an entirely new class, import an existing class into the ClassWizard database, or import a class from an OLE type library.

**Add Event**
Displays the Add Event dialog box, which you use to add an OLE event handler.

**Delete**
Deletes the selected OLE event handler.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Component Gallery: Move

Use this dialog box to move components from one category to another you select in the Move To pane.

**Move To**
Use the mouse or the ARROW keys to select a category in which to move the component(s) currently selected from the Component pane on the Customize Component Gallery dialog box.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# New Project Information

Use the New Project Information dialog box to confirm the AppWizard (or Custom AppWizard) selections you have made. Choose OK to build the basic MFC application using this information. To modify the information AppWizard uses to create the application, first choose the Cancel button. Then choose the Back button until AppWizard displays the step with the information you want to change.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Component Gallery

Use this dialog box to store components. Components can be reusable code, such as OLE Controls, or they can be useful tools, such as a code analysis tool. You can insert components into projects.

Components are displayed on, and organized by, tabbed catagories in the top pane of the dialog box. Component Gallery allows you to create, rename, and delete categories. You can also add and remove components as well as modify many of their properties such as their names, icons, and descriptions. You perform these component-management tasks from the Customize Component Gallery dialog box.

**Component Description**
A component displays its own description in a line just below the top pane. This description is one of the properties controlled from the General tab on the Properties dialog box. To reach the Properties dialog box, click the Customize button and then the Properties button.

{ewc msdncd, EWGraphic, dev39e 0 /a "MSDEV40.BMP"}

A question-mark button to the left of the help line allows the component to display a more complete help topic about itself.

**Insert**
If a component is reusable code, inserting it into a project adds the component's files to the project.

If a component is a tool, the result of inserting it depends on the purpose of the tool. Consult the component's documentation.

**Customize**
Displays the Customize Component Gallery dialog box. You can use this dialog box to manage categories and components.

**Close**
Closes the Component Gallery.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Customize Component Gallery

Use this dialog box to create new categories, move components between categories, add components to a category, and delete components or categories.

**Categories**
This pane lists the categories displayed as folders in the Component Gallery. You can perform one of the following actions:

- Select a category and then move the focus to the category's components using the TAB key.
- Select a category and move it to a new position in the Categories pane. To move the category with the mouse, hold down the left mouse button and move the mouse pointer to a new location. A gray horizontal bar indicates the insertion point. To move it with the keyboard, hold down ALT and press the UP or DOWN ARROW keys to reposition the category.
- Select a category and then type a name to rename the category.
- Select the new-item box and then type a name to add a new category.
- Select a category and press the DEL key to remove it.

**Component**
Lists the components contained in the currently open category. You can perform one of the following actions:

- Select a component and then type a name to rename the component.
- Select one or more components and then click the MOVE button to move them, under keyboard control, to a new category. Press CTRL and then use the UP or DOWN key to select multiple components with the keyboard.
- Select a component and press the DEL key to remove it.
- Use the CTRL+SHIFT+TAB accelerator to move the focus to the Categories pane.

**Properties**
Displays the Properties dialog box, which allows you to view and modify component properties.

**Move**
Displays the Move dialog box, which allows you to use the keyboard to move one or more components to another category.

**Import**
Displays the Import Component dialog box, which allows adding one or more components, such as OLE controls, to a category from an external source.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# OLE ControlWizard - Step 1 of 2

**How many controls would you like your project to have?**
Specifies the number of OLE controls that your project will create. OLE ControlWizard allows you to create a project that contains up to 99 OLE controls.

**Would you like the controls in this project to have a runtime license?**
OLE ControlWizard inserts several function calls and generates a separate .LIC file that supports licensing for your control.

**Would you like source comments to be generated?**
OLE ControlWizard inserts comments in the source and header files that guide you in writing your control. The comments indicate where you need to add your own code. This option is enabled by default.

**Would you like help files to be generated?**
ControlWizard generates a set of help files that are used to provide context-sensitive help. Help support requires using the Help compiler, which is provided with Visual C++.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# OLE ControlWizard - Step 2 of 2

**Select the control whose options you wish to edit. You may edit its class and file names if you wish.**
Use this drop-down list to select one of the controls in your project. OLE ControlWizard allows you to create a project containing as many as 99 OLE controls. The default names of the controls are created from the project name you specify in the Name box on the New Project Workspace dialog box.

**Edit Names**
Double-click this button to modify the default names associated with the currently selected control.

**Activates when visible**
Check this box to have the currently selected control indicate to the container that the control prefers to be automatically activated when it is visible. The container is not required to support this request.

**Invisible at runtime**
Check this box to have the currently selected control indicate a preference to its container that the control should be invisible in the container's run-time mode and visible in the container's design-time mode. A container may ignore the control's preference. In such a container, your control will be visible at all times.

**Available in "Insert Object" dialog**
Check this box to have the currently selected control listed in the Insert Object dialog box of a container application.

**Has an About box**
Check this box to create a standard About dialog box and AboutBox method for the currently selected control. The About dialog box is displayed when your control's AboutBox method is invoked by the container.

**Acts as a simple frame control**
Check this box to have the currently selected control support the ISimpleFrameSite protocol. When the container and the control both support this protocol, the container uses simple-frame controls as parents for other controls in the container. In effect, the simple frame control operates as an OLE compound document container, but the frame control's container does nearly all the work.

**Which window class, if any, should this control subclass?**
Choose a common Windows control from the drop-down list, such as a button, toolbar, or edit box, to subclass.

**Would you like advanced ActiveX enhancements?**
Double-click the Advanced button to display the Advanced ActiveX Features dialog box. In the dialog box, you can choose Windowless, flicker-free, asynchronous, and other ActiveX control optimizations.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# OLE ControlWizard: Edit Names

OLE ControlWizard names the currently selected control, its classes, and its files by using the project name you specify in the Name box on the New Project Worksapce dialog box. The Edit Names dialog box allows you to modify the default names associated with the currently selected control.

**Short name**
The base name of the control. The default is based on the name of the project. If you change the default, OLE ControlWizard changes the other names that appear on this dialog box.

**Control Class Name**
The name of the C++ class that represents the control.

**Header File**
The name of the header file (.H) that contains the control's class definition.

**Type Name**
The name exposed to the programmer from an Insert Object list.

**Implementation File**
The name of the source file (.CPP) that contains the class implementation.

**Type ID**
The ID of the OLE control class. This is the string that an OLE control registers in the registry when it is applied to a project. This string is used by a container application to create an instance of an OLE control.

**Property Page Class Name**
The C++ class that manages the user interface for viewing and editing the properties of an OLE control.

**Header File**
The name of the header file (.H) that contains the class definition.

**Type Name**
The name exposed to the programmer. The property page user type name is rarely used.

**Implementation File**
The name of the source file (.CPP) that contains the class implementation.

**Type ID**
The ID of the OLE property page class. This is the string that an OLE control registers in the registry when it is applied to a project. This string is used by a container application to create an instance of an OLE control's property page.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# OLE ControlWizard: Advanced ActiveX Features

OLE ControlWizard offers advanced features that can accelerate the display and operation of controls in your application.

**Windowless activation**
Makes your control use windowless activation. Often a control does not need a window of its own and can use the window services of its container. Windows add code size to controls and degrade their creation speed. You must use windowless activation for transparent or nonrectangular controls.

**Unclipped device context**
Disables tests for clipping. Selecting this option results in a small but detectable speed gain. You should only select this option if you are certain your control doesn't paint outside its client rectangle. Not available for windowless controls.

**Flicker-free activation**
Eliminates drawing operations and the accompanying visual flicker in transition between inactive and active states. Select this option only if the control draws itself identically in the inactive and active states. Not available for windowless controls.

**Mouse pointer notifications when inactive**
Allows your control to process **WM_SETCURSOR** and **WM_MOUSEMOVE** messages when it is not active. The container delegates messages to **IPointerInactive**, which dispatches the messages through your control's message map. You process the messages like ordinary window messages, by adding the corresponding entries to the message map.

**Optimized drawing code**
Indicates that the control wishes to perform optimized drawing if the container supports it.

**Loads properties asynchronously**
Specifies that your control can have properties that point to data that is loaded in the background. You should call **COleControl::InternalSetReadyState** when the readiness of the control changes (for example, when the asynchronous properties are done loading).


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# ClassWizard: Repair Class Information

Use the Repair Class Information dialog box to:

- Delete obsolete classes from the ClassWizard .CLW file.
- Update the information in the .CLW file with new names or locations of classes that you have changed.

**Class Name**
The name of the class that you want to repair.

**Header File**
The name of the file to contain header information for the repaired class.

**Implementation File**
The name of the file to contain implementation code for the repaired class.

**Browse**
Click this button to select from a list of files.

**Remove**
Removes the class from ClassWizard's list of classes.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Select a Record Set

Use the Select A Record Set dialog box to associate a record set with a new record view class.

**Class**
Select a record set for the new record view class.

**New**
Opens the SQL Data Sources dialog box, which you use to select a database.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Select Source Files

Use the Select Source Files dialog box to add or delete files in existing projects when rebuilding the ClassWizard .CLW file. It initially lists all the files contained in the project. You can add or delete files if necessary.

**Tips**    You can double-click a filename in the list box to add it to the list of files used to build the .CLW file used by ClassWizard.

You can double-click a filename in the Files In Project list box to delete it from the list.

**File name**
Type the name of a file to open, save, or select. If you type a pattern using a wildcard (* or ?) and press ENTER, the list box displays files matching that pattern.

**Directories**
Double-click a directory entry to display files from that directory in the Files box.

**List files of type**
Click the arrow to display a list of file types. The filename extension determines the file type. Click a file type to display a list of all files of that type from the current drive and directory in the File Name box.

**Drives**
Select the drive containing the directories you want to browse.

**Files in project**
Lists the currently defined ClassWizard project files.

**Add**
Adds selected files to the list of Files in Project.

**Add All**
Adds all of the files listed in the Files in Project to the ClassWizard .CLW file for the project.

**Remove**
Removes the selected file from the Files in Project list. The file that you remove is not used to build the ClassWizard .CLW file.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Database Options

Use this dialog box to select a database.

**ODBC**
This list box displays all data source names currently defined on your machine via the ODBC Administrator. Double-click the name of your data source.

After you select a data source, a dialog box opens to let you select a table on the data source. The wizard binds all columns of the table to the member variables of a **CRecordset**-derived class.

**DAO**
This edit box allows you to specify an existing database. If you click the button to the right of the edit box, a File Open dialog appears to help you select an existing database. By default, the dialog box shows only Microsoft Jet (.MDB) databases. If you want to work with another database management system (DBMS), such as an ODBC data source, it is more efficient to attach tables from that source to an .MDB database than to open the database directly. For more info, see the article DAO External: Working With External Data Sources in *Programming With MFC*.

**Snapshot**
A snapshot is the result of a query and is a view into a database at one point in time. A snapshot is static in nature. All records found as a result of the query are cached. Using a Snapshot, you will not see any changes that occur to the original records.

**Dynaset**
A dynaset is the result of a query that provides an indexed view into the queried database's data. A Dynaset caches only an integral index into the original data and thus offers a performance gain over a Snapshot. Because you have an index that points directly to each record that was found as a result of a query, you can tell if a record is removed. You will also have access to updated information in the queried records.

**Table**
A table provides you with a means of directly manipulating the records and data in a database.

**Detect dirty columns**
Sets **m_bCheckCacheForDirtyFields** to TRUE which creates a data cache to detect whether data values or NULL status has changed. For more information, see DAO RecordFieldExchange: Double Buffering Records in *Programming with MFC: Encyclopedia*.

**Bind all columns**
Creates a recordset variable for each column in the selected database. This option is available only from ClassWizard.


{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Select Database Tables

Displays the tables contained in the selected data source. Select the tables for which you want recordsets created.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Component Gallery: General Properties Tab

Displays the title, description, and an icon of a component selected from the Customize Component Gallery dialog box. You can perform one of the following actions:

- Type a short description, to be displayed on the Component Gallery's main dialog box, into the Descriptions box.
- Use the Change Icon button to select an icon for the component.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Component Gallery: Custom Properties Tab

Displays the following attributes of the selected OLE control or component:

- The name and icon.
- If the component was imported, the path to the Component and the date that it was last changed.
- The path to the workspace where the component was created.
- The contents of the component.

You can use the Export button to save the current component in file form.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Component Gallery: OLE Control Properties Tab

Displays the following attributes of the selected OLE control:

- The name and icon.
- The path to the OLE control and the date that it was last changed.
- The OLE control's version number.

You can use the Register button to reregister the control should that ever be necessary.

{ewl msdncd.dll, ewcright, /c"Microsoft"}

# Type Library Tool

Use the Type Library Tool dialog box to make your application an OLE Automation Client. You can do this by scanning for type libraries that contain exported interfaces available to your application. After selecting a type library, you can create a C++ class to encapsulate it.

Under Windows 95, you get help on the following controls by selecting the control and then pressing F1:

**Look in**
**File name**
**Files of type**

{ewl msdncd.dll, ewcright, /c"Microsoft"}