Main Index

Resource Manager

Editors

Resource Manager: Basics

The Resource Manager is the main window and the center of all activity in the Resource Toolkit. It provides access to existing resources and to each of the resource editors.

- To activate any button, click the button, or press the keyboard key corresponding to the button's underscored letter. For example, press U to choose the **Include** button.

- To cycle to a button or field, press the TAB key. For example, you can press TAB until a resource browser's edit field is activated.

Resource Manager: Procedures

Starting an Editor

To start one of the Resource Toolkit's editors, you use the Resource Manager. You can use either of two methods:

- Click the editor's button.

- Press the keyboard key that corresponds to the first letter in the editor. For example, press D to start the Dialog Editor.

Accessing Existing Resources

To access an existing resource, you must open the file that contains it.

- Before opening the file, you can choose the resource types you want to see displayed from the open file. By default, all resource types are displayed.

- Then use one of the Resource Manager's browsers to open the file and select a resource type. This displays individual resources of the selected type.

- Once the individual resources are listed, you can select one to edit, or select one or more to copy, rename, or delete.

Choosing Resource Types to Display

At any time, you can limit the resource types that will be displayed for an open file. For example, you can limit the display to only the file's icon and cursor resources, or only its menu resources.

To limit the display of resource types:

> Click the **Include** button. This opens a dialog that lists all the resource types you can edit, copy, or rename.

> Select each resource type you want to include in the display, and clear the selection from each resource type you want to exclude. An **X** in a resource type's check box indicates that it's selected.

There's a quick way to select or clear all resource types.

If none of the check boxes are checked, or some of them are checked, click the **Select All** button to select all the resource types.

If all the resources are checked, the button changes to an **Unselect All** button. When clicked, **Unselect All** clears all selections.

Opening an Existing File

To open an existing file:

> Use the **Include** button to select the resource types you want to see displayed when a file is open.

> Determine which resource browser you want to use and click its **Open** button. This opens the File Open dialog.

> Select the type of file you want by clicking the appropriate radio button in the File Type field. For example, to see a list of EXE files, click the EXE radio button.

> If the file you want isn't in the current directory, change to the directory where it's stored.

> Double-click the file you want, or select it and click the dialog's **OK** button.

> Select the resource type you want.

> Select the individual resource(s) you want to work with.

> Edit, copy, rename, or delete the selected resource(s).

To change the current directory:

- Double-click a directory in the Directories box.

- Alternatively, type a path in the Filename field. For example, to move up one directory level, you could type **..** in the Filename field and press ENTER or click **OK**.

Selecting a Resource Type

In a resource browser, you can select a resource type in any of the following ways:

- Press ENTER to accept the selected resource and start its corresponding editor.

- Activate the browser's edit field by clicking it, or by pressing the TAB key until the field is activated. Then type the first letter of the resource type you want to select. For example, click the edit field and type D to select **Dialog** resource types.

- Click the arrow that's to the right of the browser's edit field. This drops down a list box that displays all available resource types. To select a resource type from the list, click it, or press the key corresponding to the type's first letter. For example, press D to select **Dialog** resource types.

Once you select the resource type, individual resources of that type are listed in the browser's list box.

For copying, renaming, or deleting resources, you can select multiple resources.

To select a range of resources, select the first resource in the range, then press the SHIFT key and select the last resource in the range. This selects all inclusive resources as well.

To select several individual resources, press the CTRL key and click each resource you want to select.

To de-select an individual resource from a group of selected resources, press the CTRL key and click that resource.

Editing a Resource

To edit an existing resource:

> Open the file that contains the resource.

> Select a resource type from the resource browser's edit field.

> In the list box, double-click a resource's name or number. Alternatively, select the resource and click the **Edit** button. This automatically starts the appropriate editor and loads the resource into that editor.

Deleting a Resource

To delete a resource from a file:

> Open the file that contains the resource.

> Select a resource type from the resource browser's edit field.

> In the list box, select the resource(s) you want to delete.

> Click the **Delete** button. This deletes all selected resources from the open file.

Caution. The resources are immediately deleted from the file. Make sure you select only those resources you don't want or need.

Copying a Resource

To copy a resource from one file to another:

> Use one of the resource browsers to open the file whose resource you want to copy.

> Use the other browser to open the file you want to copy to. This can be a new or an existing file.

> From the edit field for the source file, type or select the resource type you want to copy.

> Select one or more resources to copy.

> Click the **Copy** button. This opens a dialog that asks whether you want to rename the selected resources.

> Click **Yes** or **No**:

- **No** copies the resources without renaming them.

- **Yes** opens the Resource Attributes dialog so you can rename the resource. If you've selected multiple resources, the dialog opens separately for each selected resource.

  Choose the Load and Memory options you want for the new resources. Click the **Default** button to get the defaults for the type of resource you're copying.

  If you don't want to rename the current resource, click the **OK** button. To rename the current resource, type the new name into the dialog's Name field, then click the **OK** button.

Renaming a Resource

To rename any resource in a file:

> Open the file that contains the resource.

> Select a resource type from the resource browser's edit field.

> In the list box, select the resource(s) you want to rename.

> Click the **Rename** button. This opens the Resource Attributes dialog so you can rename the resource. If you've selected multiple resources, the dialog opens separately for each selected resource.

> To rename the current resource, type the new name into the dialog's Name field, then click the **OK** button.

Creating a File

Use either of two methods to create a file in the Resource Toolkit:

- Start one of the Resource Toolkit's editors. Use this method to create a new file and begin defining a new resource.

- Create the new file in the Resource Manager. Use this method to create a new file so you can copy resources from an existing file. The advantage of this method is it lets you copy multiple resources at once, without having to start an editor to load each resource.

Creating a New File in the Resource Manager

To create a new file so you can copy existing resources into it:

> Determine which of the Resource Manager's resource browsers you want to use and click its **New** button. This opens the File New dialog.

> Select the type of file you want to create by clicking the appropriate radio button in the File Type field. For example, to create a RES file, click the RES radio button.

> To create the file in a different directory, change the directory.

> Either create and name the new file, or replace one of the existing files that are displayed in the Files box.

To copy resources into the file, use the Resource Manager's other resource browser to open an existing file.

Once both files are open, you can copy resources from the existing file into the new file.

Naming the New File

To create the new file:

> Click the File Extension radio button for the type of file you want to create.

> Position the cursor in the Filename field and type a name for the file. You don't have to include a file extension since the extension is taken from the File Type radio button.

Replacing an Existing File

To replace one of the files listed in the Files box:

> Select the file. This automatically copies the file's name into the Filename field.

> Click the dialog's **OK** button. This opens a dialog to confirm the replacement:

- **Yes** replaces the existing file with a new, empty file.
- **No** returns you to the dialog so you can specify another file name.

Caution. If you click **Yes**, the new file is immediately created on disk, replacing the existing file. Don't select **Yes** unless you're sure you want to replace the existing file.

Closing a File

When you use a resource browser to open a new or existing file, the browser's **Open** and **New** buttons change to a **Close** button.

To close the file when you're done with it, click the **Close** button.

When you click the **Close** button, all changes you made to the file are immediately saved. **Close** *does not* open a dialog to confirm you want to save the changes.

Accelerator Editor

Fields: Accelerator Editor

Alt
Code
Ctrl
Invert
Key
Shift
Symbol
Type
Value

Keys Used in the Accelerator Editor

Accelerator Editor: Basics

The Accelerator Editor lets you create and edit accelerator resources. Accelerators are *hot keys* for issuing an application command.

The Accelerator Editor lets you create accelerators by pressing the key rather than entering its code. For example, to make the F2 key an accelerator, press F2 while the table's active field is in the Key column.

The Accelerator Editor stores accelerator keys and codes in an *accelerator table*. An accelerator table normally contains seven columns that display information about the accelerator. If a header file is open, an additional column shows the symbolic name of the accelerator.

Files: Accelerator Editor

The Accelerator Editor can edit accelerator resources from resource (RES) files, executable (EXE) files, and dynamic link libraries (DLLs).

The Accelerator Editor can save an accelerator resource directly into a new or existing RES file, or into an existing EXE file or DLL.

The Accelerator Editor can open a header file (also called an include file) to correlate symbolic constants from the file with the accelerator resource. This adds a Symbol column to the accelerator table, and opens the Header Editor. You can edit symbol values with the Header Editor, or edit them directly from the accelerator table.

Alt Field

The Alt field indicates whether the ALT key must be depressed to activate the accelerator displayed in the Key field.

If Virtkey is selected in the Type field, the Alt column cycles between Yes and No. The value is automatically updated when an accelerator is defined in the Key field, so it's usually unnecessary to change this field yourself. To change it, use the SPACEBAR, or press Y for Yes, or N for No.

If ASCII is selected in the Type field, this field is blank. The ALT key doesn't apply to ASCII accelerator codes.

Code Field

The Code field shows the accelerator's keyboard scan code and is automatically filled in when the Key field is filled. The code depends on whether Virtkey or ASCII is selected in the Type field.

The Code is a key's ID number on the keyboard. It is not the ID used in source code to identify the accelerator; that is displayed in the Value field.

Ctrl Field

The Ctrl field indicates whether the CTRL key must be depressed to activate the accelerator displayed in the Key field.

If Virtkey is selected in the Type field, the Ctrl column cycles between Yes and No. The value is automatically updated when an accelerator is defined in the Key field, so it's usually unnecessary to change this field yourself. To change it, use the SPACEBAR, or press Y for Yes, or N for No.

If ASCII is selected in the Type field, this field is blank.

Invert Field

The Invert field indicates whether the main menu item associated with the accelerator should be highlighted when the accelerator is pressed. The value must be either Yes or No.

Key Field

The Key field defines the accelerator key. To define the accelerator, press the desired key. For example, to define SHIFT+INSERT as the accelerator -- a common accelerator for **Paste** -- hold down the SHIFT key and press the INSERT key.

When you define an accelerator key, all fields in the row, except for Value, are automatically filled in.

Depending on the accelerator you define, different strings may appear in the Key field. For example, if you define a Virtkey CTRL+A, an **A** appears in the field. However, if you define an ASCII CTRL+A, **^A** appears in the field.

If you try to enter an accelerator that's invalid for the format selected in the Type field, the editor beeps to indicate it can't use that key.

Shift Field

The Shift field indicates whether the SHIFT key must be depressed to activate the accelerator displayed in the Key field.

If Virtkey is selected in the Type field, the Shift column cycles between Yes and No. The value is automatically updated when an accelerator is defined in the Key field, so it's usually unnecessary to change this field yourself. To change it, use the SPACEBAR, or press Y for Yes, or N for No.

If ASCII is selected in the Type field, this field is blank. The SHIFT key doesn't apply to ASCII accelerator codes.

Symbol Field

The Symbol field shows the symbol whose ID number in the header file matches the value for the current accelerator. To load the accelerator into your application, you can refer to this symbol in the application's source code.

Type Field

The Type field indicates whether the accelerator is defined in Virtkey or ASCII code.

- Virtkey stands for virtual key and names accelerators according to a set of virtual key definitions defined for Windows.

- ASCII shows the accelerator in terms of its ASCII value. For example, the key combination CTRL+S is Virtkey **S** and ASCII **^S**.

To change the type, press the SPACEBAR or type a V or an A while in the Type field.

Once an accelerator has been defined, you can view it in its other form by cycling to the opposite choice. However, not all accelerators can be converted from one type to the other. For example, there's no ASCII equivalent for F5 or the PgUp key, and there's no Virtkey equivalent for an exclamation point, '!'.

Value Field

The Value field shows the ID number for the accelerator defined in the Key field. To load the accelerator into your application, refer to this number in the application's source code. Alternatively, define a symbol for the accelerator.

Enter the value as it's defined in the source code. The value must be an integer.

Generally, if the accelerator duplicates a menu choice, use the same ID as the menu item's ID. If the accelerator doesn't duplicate a menu choice, use a unique ID.

The ID values of accelerators can be the same as the ID's used in other resources, such as for the controls in a dialog box. This is because Windows interprets the ID value within the context of the resource currently being used in the application.

Navigating an Accelerator Table

CTRL+ENTER	Adds a new row to the accelerator table.
TAB	Moves one column to the right
SHIFT+TAB	Moves one column to the left.
Up arrow	Moves the active field up one row.
Down arrow	Moves the active field down one row.
Right arrow	In the Value or Symbol fields, moves the cursor one position to the right. In any other field, same as TAB.
Left arrow	In the Value or Symbol fields, moves the cursor one position to the left. In any other field, same as SHIFT+TAB.
HOME	In the Value or Symbol fields, moves the cursor to the beginning of the field. In the Keys field, defines the HOME key as an accelerator. HOME has no function in any other field.
END	In the Value or Symbol fields, moves the cursor to the end of the field. In the Keys field, defines the END key as an accelerator. END has no function in any other field.

Typing in the Accelerator Table

If you activate a field from the Value column, or when a header file is opened, the Symbol column, all text in the field is selected.

- To replace the text, begin typing, or press the DELETE key.

- To preserve the text, move the cursor in the field, using the mouse, an arrow key, or the HOME or END key. You can then edit some portion of the text.

SHIFT+right arrow	Selects the character one position to the right.
SHIFT+left arrow	Selects the character one position to the left.
SHIFT+END	Selects all characters from the current cursor position to the end of the line.
SHIFT+HOME	Selects all characters from the current cursor position to the beginning of the line.

Selection Fields in an Accelerator Table

The Type, Shift, Ctrl, and Invert fields let you select one option from a set of available options:

To cycle through available selections in the field, press the SPACEBAR.

Alternatively, type the first letter of the selection. For example, in a Yes/No field, if Yes is the current value, press the N key to cycle to No.

Bitmap Editor

Cursor Editor

Icon Editor

Bitmap Editor: Basics

The Bitmap Editor can create and edit device-independent bitmaps. To create a bitmap, you draw it with tools like those from a paint program. In addition to choosing the type of line or shape to draw, you choose the colors for the bitmap.

To create a new bitmap, choose **New** from the **File** menu. This opens a dialog so you can define the height and width of the new bitmap, and set the number of colors.

By default, the dialog specifies a 16-color bitmap with dimensions of 72-by-72 pixels. You can select a two-color bitmap, and/or change the dimensions for the height and width. The values you can specify for both height and width are limited only by available memory.

Cursor Editor: Basics

To create a cursor in the Cursor Editor, you draw it with tools like those from a paint program. In addition to choosing the type of line or shape to draw, you also choose the colors. You can even view how those colors will appear against different background screen colors.

> To create a new image for an existing cursor, choose **New** from the **Image** menu. To create a new cursor resource, choose **New** from the **File** menu. Either choice opens a dialog so you can specify a resolution for the image. Available resolutions are read from file WRT.DAT.

> To select a resolution, click the arrow that's to the right of the dialog's edit field. This drops down a list box that displays all the available resolutions from the file WRT.DAT.

> Select a resolution and click **OK**.

Icon Editor: Basics

To create an icon in the Icon Editor, you draw it with tools like those from a paint program. In addition to choosing the type of line or shape to draw, you also choose the colors. You can even view how those colors will appear against different background screen colors.

> To create a new image for an existing icon, choose **New** from the **Image** menu. To create a new icon resource, choose **New** from the **File** menu. Either choice opens a dialog so you can specify a resolution for the image. Available resolutions are read from file WRT.DAT.

> To select a resolution, click the arrow that's to the right of the dialog's edit field. This drops down a list box that displays all the available resolutions from the file WRT.DAT.

> Select a resolution and click **OK**.

Files: Bitmap Editor

The Bitmap Editor can edit bitmaps resources from resource (RES) files, executable (EXE) files, and dynamic link libraries (DLLs).

The Bitmap Editor can save a bitmap resource directly into a new or existing RES file, or into an existing EXE file or DLL. It can also create, edit, and save bitmaps into bitmap (BMP) files.

Files: Cursor Editor

The Cursor Editor can edit cursor resources from resource (RES) files, executable (EXE) files, and dynamic link libraries (DLLs).

The Cursor Editor can save a cursor resource directly into a new or existing RES file, or into an existing EXE file or DLL. It can also create, edit, and save cursors into cursor (CUR) files.

Files: Icon Editor

The Icon Editor can edit icon resources from resource (RES) files, executable (EXE) files, and dynamic link libraries (DLLs).

The Icon Editor can save an icon resource directly into a new or existing RES file, or into an existing EXE file or DLL. It can also create, edit, and save icons into icon (ICO) files.

Working with Color: Bitmap Editor

Drawing and Editing Bitmaps

Working with Color: Cursor and Icon Editors

Drawing and Editing Icons

Drawing and Editing Cursors

Menu Commands

Menu Commands

Choosing Colors: Bitmap Editor

The Color palette shows the available colors for the current resource. The palette uses two types of colors, pure colors and dithered colors.

Before you begin drawing with a selected tool, choose the color you want for the new line or shape by clicking a color in the color palette. This selects the color and displays it in the color box above the palette.

If you select a dithered color for drawing a line, the editor uses the closest pure color instead.

Choosing Colors: Cursor and Icon Editors

The Color palette shows the available colors for the current resource. The palette uses two types of colors, pure colors and dithered colors.

Before you begin drawing with a selected tool:

> Determine whether you want to draw in Color mode, Screen mode, or Inverse mode. To select the mode, click its corresponding color box. Only one mode at a time can be in effect, but you can switch modes at any time. The current mode is indicated by a heavy border around its color box.

> Choose the color you want for the new line or shape by clicking a color in the color palette. This selects the color and displays it in the color box for the current drawing mode.

Pure colors are colors that can be directly generated by the output device. Pure colors are combined values of red, green, and blue (RGB) that are guaranteed to be distinct on devices that support 16 or more colors. For that reason, all lines drawn by the graphic tools use pure colors.

Hint. If you're not sure whether a color is pure, use the Pouring tool to fill a large area with the color. If the color in the area appears solid, it's pure. If the color has a checkered pattern, it's dithered.

Dithered colors are colors the windowing environment synthesizes because the output device can't generate them directly. This is done by a process known as *dithering*, where pure colors are combined in a pattern of dots to approximate another color. The simplest example is creating gray by dithering black and white. On VGA devices, the first eight columns in the color palette are pure, and remaining colors are dithered.

Dithered colors are best used for coloring areas. This is because they're simulated by a series of dots and don't have good resolution for drawing lines.

Dithered colors have several disadvantages. For example, though they may look good on your display device, they won't necessarily look good on all devices. Moreover, for some display devices, the windowing environment might find it necessary to switch a dithered color to a pure color, and there's no guarantee the pure color will blend well with other colors in the image.

Color mode specifies a permanent color that retains its hue, regardless of the background screen color.

Screen mode specifies the color of the background screen. This color mode lets you see how your image will look against various background colors. The current color for Screen mode is displayed as a background color in the View window.

Any lines you draw in the image with Screen mode will always take on the color of the screen background, making them appear invisible in the application. Screen mode can therefore be used to suggest a transparent area.

Inverse mode specifies the color that is the inverse of the screen color. If a line or shape is drawn in Inverse mode, it always takes the inverse color to its background on a display device.

Permanent colors are invisible when the permanent color is the same color as the background screen color. Inverse colors, on the other hand, change color as the screen color changes. Thus, Inverse colors are always visible against any screen background color. To ensure an image is always visible against any screen background, you can outline it with a color in Inverse mode.

Customizing Colors

The Bitmap Editor lets you customize the color palette by changing its colors. The hue for a color is always a mixture of red, green, and blue (RGB).

Customizing Color in a 16-color Bitmap

Customizing Color in a Two-color Bitmap

Customizing Color in a 16-color Bitmap

To edit any color from the palette in a 16-color bitmap:

> Double-click the color in the palette. Alternatively, select the color, then choose **Edit Color** from the **Palette** menu. This opens a dialog that lets you change the color's RGB color number.

> If you know the RGB color number you want, type it into the editing fields. Otherwise, use the scroll bars to scroll a color blend. The color box to the right of the scrolls shows the current blend of red, green, and blue.

> Click **Accept**, **Cancel**, or **Default**:
- **Accept** uses the color you specified to replace the original color in the palette.
- **Cancel** cancels the editing operation and restores the original color to the palette.
- **Default** selects the palette position's default color value. Every position in the palette has a default color. That color may or may not be the original color you selected.

Customizing Color in a Two-color Bitmap

To edit the text or background color for a two-color bitmap:

> Choose either **Text Color** or **Background Color** from the **Options** menu. This opens a dialog that lets you change the color's RGB color number.

> If you know the RGB color number you want, type it into the editing fields. Otherwise, use the scroll bars to scroll a color blend. The color box to the right of the scrolls shows the current blend of red, green, and blue.

> Click **Accept** or **Cancel**:
- **Accept** uses the color you specified to replace the original color in the palette.
- **Cancel** cancels the editing operation and restores the original color to the palette.

For a two-color bitmap, you can't save the modified palette as you can with a 16-color bitmap. However, saving the bitmap into a file also saves its color values. When you later open the bitmap file, the palette contains the edited rather than the default text and background colors.

Saving a Customized Color Palette

To save an edited Color palette for a 16-color bitmap:

> Choose **Save Colors** from the **Palette** menu. This opens a dialog so you can name the new palette.
> Specify a directory and name for the new palette. The file extension must be PAL.

Getting a Customized Color Palette

At any time while editing a 16-color bitmap, you can get a customized color palette from a PAL file to use during the editing session:

> Choose **Get Colors** from the **Palette** menu. This opens a dialog so you can specify an existing palette.

> Specify a directory and name for the palette, or select them from the list boxes. Only files with file extension PAL are available.

Choosing a Line Width

Before using any drawing tool, use the Line Width tool to select a line width. The Line Width tool provides three options:

- Thin line: 1 pixel.
- Medium line: 3 pixels.
- Thick line: 5 pixels.

By default, the Line Width tool draws a thin line. To select another thickness, click the tool, or use the **Options** menu to choose a width.

Each time you click the Line Width tool, the width cycles to the next selection. The tool indicates the current selection by filling the bullet that's to the left of the selected width.

All lines in the editors are drawn with pure color, even when you choose the thick line width.

Magnifying the Editing Area

Before drawing, use the Magnifying tool to select a scale for the editing area. The Magnifying tool provides three options:

- Actual size.
- Magnify four-times actual size.
- Magnify eight-times actual size.

To select a magnification, click the tool, or use the **Options** menu to choose a magnification.

Each time you click the Magnifying tool, the scale of the editing area cycles to the next selection. The tool indicates the current scale by filling the square that represents the current editing area.

With four- and eight-times magnification, the entire graphic may not be visible in the editing area, though you can see it in its entirety in the View window. You can also shift your view of the graphic in the editing area.

Toggling to an Inverse Color Value

Use the Toggle tool to draw with the color whose color value is the inverse of the color value at the point where you begin drawing. The Toggle tool works only with the Pencil tool.

By default, the Toggle tool is off. To turn it on, click it. To turn it off again, click it again. The tool indicates its current status through its two sets of rectangles.

The color toggles with each mouse click. When you click the mouse button, the Pencil reads the color value of the pixel at the current cursor position and paints with the inverse color. Thus, when you begin drawing over a white area, the Pencil draws in black. When you begin drawing over a black area, the Pencil draws in white. It always draws with the inverse color to the starting pixel's color value.

The mouse button must be released for the tool to respond to a new background color. Thus, if the Pencil is currently drawing in white and you drag it over a white area, it doesn't automatically toggle to black.

In the Cursor and Icon editors, the Toggle tool changes only color values. It *doesn't* change the drawing mode to Inverse mode.

When the toggle is off, the rectangles suggest movement from white to white and from black to black, indicating the color won't toggle.

When the toggle is on, the rectangles suggest movement from white to black and from black to white, indicating the color will toggle.

Changing Your View of a Graphic

When the editing area is magnified to four- or eight- times the size of the actual graphic, the graphic is sometimes larger than can be displayed in the editing area. When this is the case, you can change your view of the graphic in any of the following ways:

Use the horizontal and vertical scroll bars to change the view of the editing area.

Use the Dragging tool to drag the graphic. To use the Dragging tool, select it, then click anywhere in the editing area and drag in any direction. This has the same effect as but is faster than using the scroll bars.

Click the left mouse button in the View window and hold the button down. The dark area represents the visible work area in the editing area. To change the view, click the right mouse button in the View window. The editing area scrolls so the point where you right-clicked is moved as close as possible to the center of the editing area.

Using the Tools

Drawing with the Tools

The procedure is the same for using any tool:

> Select the color you want to work with. For cursors and icons, you can select a Screen or Inverse drawing mode in addition to a permanent Color mode. For bitmaps, there's only a permanent Color mode.

> Choose a line thickness with the Line Width tool, and choose the editing area's magnification with the Magnifying tool.

> If you want the Pencil tool to toggle to inverse colors, turn on the Toggle tool.

> Select the tool you want to use. By default, the Pencil tool is selected. To select a tool, click it in the Tools palette.

> Position the pointer where you want to begin drawing.

> Click the left mouse button. Holding the button down, drag the mouse in any direction. This extends a line or rectangle, depending on the tool you've selected. As long as you keep the left mouse button depressed, you can continue to adjust the size of the line or shape.

> Release the mouse button when you have the size you want. This draws the line or shape, using the position and size you specified. For a polygon, repeat the drawing steps for each side of the line.

Free-form Lines and Shapes

Use the Pencil tool to draw free-form lines and shapes. Like a physical pencil, a digital pencil draws a line anywhere you drag it. It can also fill in points and areas. For filling large areas, however, it's best to use the Pouring tool.

To draw a point with the Pencil, click the desired spot without dragging the mouse.

Straight Lines

Use the Line tool to draw straight lines. The Line tool draws in any direction.

Use the Constrained Line tool to draw a line that's vertical, horizontal, or at any multiple of 45 degrees.

Rectangles

Use the Rectangle tool to draw a hollow rectangle. Once the rectangle is drawn, you can leave it hollow, or use the Pencil or Pouring tool to fill it with a color that's different from the color you used to draw it.

To fill a rectangle with the same color as you used to draw it, it's quicker to draw it with the Filled Rectangle tool.

The Filled Rectangle tool draws a solid rectangle in the selected color. You can either leave the rectangle filled in, or select another color and draw over selected sections.

Ellipses

Use the Ellipses tools to draw hollow or filled ellipses. Once the ellipse is drawn, you can leave it hollow, or use the Pencil or Pouring tool to fill it with a color that's different from the color you used to draw it.

To fill a ellipse with the same color as you used to draw it, it's quicker to draw it with the Filled Ellipse tool.

The Filled Ellipse tool draws a solid ellipse in the selected color. You can either leave the ellipse filled in, or select another color and draw over selected sections.

As you drag the mouse to draw the ellipse, its shape seems perfect. When you release the mouse button in a magnified editing area, the ellipse suddenly becomes jagged. This is because the editing area shows a magnified view of the graphic, and each pixel maps to a square on your screen. When the graphic is resolved down to its actual size, the jagged lines appear more curved. If you look in the View window, you'll see that the shape looks closer to an ellipse.

Polygons

Use the Polygon tool to draw hollow or filled polygons. Once the polygon is drawn, you can leave it hollow, or use the Pencil or Pouring tool to fill it with a color that's different from the color you used to draw it.

To fill a polygon with the same color as you used to draw it, it's quicker to draw it with the Filled Polygon tool.

The Filled Polygon tool draws a solid polygon in the selected color. You can either leave the polygon filled in, or select another color and draw over selected sections.

Unlike rectangles and ellipses, which can be drawn with a single click and drag of the mouse, you have to click-drag-and-release the mouse button for each side of the polygon. However, to draw the final side, you can double-click the mouse button and let the tool draw it for you. In fact, if you don't close a polygon, the tool automatically completes it for you.

Filling Large Areas

Use the Pouring tool to fill large areas with color. By default, the Pouring tool floods the entire editing area with color. This makes it easy to provide a background color for a bitmap. To fill the editing area, select the Pouring tool and click any open area.

Alternatively, you can fill a bounded area with color. To do so, select the color, then click the tool anywhere within the area you want to fill. When you click the mouse button, the Pouring tool reads the color value of the pixel at the current cursor position and pours the selected color into all surrounding pixels with that value. It stops pouring color when it encounters a boundary whose color value is different from the color value it initially read. This makes it easy to fill irregular areas with color.

Setting the Hot Spot

Every Windows cursor has a unique pixel that precisely identifies the one point at which cursor operations occur. This point is called the *hot spot*. Regardless of a cursor's shape, it must have one pixel that can be mapped to the hot spot. Windows uses the hot spot's location to inform an application of pointing activity.

To designate a hot spot:

> Select the Hot Spot tool from the Tools palette.

> Click the desired location.

Selecting a Graphic Region

Use the Selecting tool to select a section of the graphic to cut or copy to the Clipboard. Once in the Clipboard, you can perform any of the operations available from the **Edit** menu.

Dialog Editor

Drawing and Editing Dialog Boxes

Menu Commands

Dialog Editor: Basics

The Dialog Editor gives you a visually-interactive method for designing dialog box resources.

To create a dialog box, you draw it with tools like those from a drawing program. You create and edit dialogs by selecting, locating, and modifying their controls.

The Dialog Editor has a Tools palette that lets you create a dialog box and all its controls. It also has an Alignment palette lets you change the alignment of the controls within the dialog.

By default, the palettes are located above the editing area, with the Tools palette to the left of the Alignment palette. When you change the size of the Dialog Editor on your desktop -- particularly when you reduce its size -- you may cut off your view of the palettes. In these instances, you can change the palette positions:

- To position the palettes below the editing area, choose **Palettes on Bottom** from the **Dialog** menu.

- To position the Alignment palette to the left of the Tools palette, choose **Swap Palettes** from the **Dialog** menu.

Files: Dialog Editor

The Dialog Editor can edit dialog resources from resource (RES) files, executable (EXE) files, and dynamic link libraries (DLLs).

The Dialog Editor can save a dialog resource directly into a new or existing RES file, or into an existing EXE file or DLL. In addition, it can save the same dialog specification into a dialog resource script (DLG) file.

The Dialog Editor can open a header file (also called an include file) to correlate symbolic constants from the file with the dialog resource. Opening a header file opens the Header Editor so you can edit symbolic constants in the header file.

Tools Palette

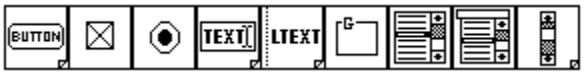
The Tools palette provides a Pointer tool, Dialog tool, and a set of control tools tools. Some of these tools are multi-style tools.

To select a tool from the Tools palette , click it in the palette, or open the Tools menu and select the tool by name.

Dialogs
Buttons
Edit controls
Static text
Group boxes
List boxes
Combo boxes
Scroll bars
Icons
Custom controls

Pointer Tool

After you select a tool, the Pointer tool is automatically activated. The pointer lets you draw with the selected tool, or select the dialog or one or more existing controls so you can move them or change their size.



Use the control tools to create controls within the dialog box. For example, one control tool creates list boxes, and another creates buttons.

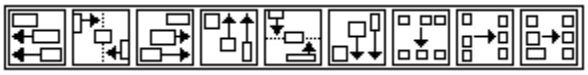
A multi-style tool is a tool that can create more than one style of dialog or control. For example, the Button tool is a multi-style tool because it can create different kinds of buttons.

The bottom right corner of a multi-style tool is folded. To cycle through the tool's options, double-click it. This changes the tool and also selects it.

Multi-style tools:

- Dialog tool
- Button tool
- Edit tool
- Text tool
- Scroll Bar tool

Alignment Palette



The Alignment palette lets you

- Adjust the position of a selected set of controls. The controls can be aligned relative to the top, bottom, left, right, or center of one control from the set. Or, they can be spread evenly across the dialog's height and/or width.

- Adjust the size of a selected set of controls, setting them all equal.

Generally, you use the alignment tools any time you create two or more controls of the same type within the dialog. That way you can ensure that they're properly aligned and proportioned.

To align controls:

> Select the control you want to use as the standard for the alignment.

> Hold down the SHIFT key and select each additional control you want to align.

> Select an alignment tool from the palette by clicking it with the mouse.

The control used as the standard for an adjustment is always the first one selected in the set. Thus, if you select two controls and then click the Make Same Size tool, the second control you selected is made the same size as the first one you selected.

If you're spreading controls across the dialog's width or height, this first control isn't used as a standard. It's simply one of the controls whose position will be adjusted.

Drawing a Dialog or Control

To create a new dialog or control:

> Select the appropriate tool from the Tools palette. For example, to create a button, select the Button tool.

> Position the pointer where you want to begin drawing. For a dialog, the position you choose determines the position the dialog will be displayed in your application. You can see the coordinates of this position in the Dialog Attributes box.

> Click the left mouse button and hold the button down.

> Keeping the button depressed, drag the mouse in any direction. This forms a rectangle to show you the size of the item you're creating. As long as you keep the left button depressed, you can continue to adjust the rectangle's size. However, you can't drag the mouse beyond any border of the window.

> When you have the size rectangle you want, lift your finger off the left mouse button. This draws the item, using the size and position you specified.

After you create a control, the Tools palette automatically resets itself to the Pointer tool. To create another control of the same type, you can either select the tool again, or hold down the CTRL key as you click and drag the mouse to create the next control. Holding down the CTRL key automatically selects the last tool used.

This technique can be used to create a control without actually drawing it. Position the pointer where you want the control, press the CTRL key, and quickly click the left mouse button. This creates the control with a default size.

Selecting a Dialog or Control

Before moving or sizing items in the Dialog Editor, you must first select them. When an item is selected, you'll see a sizing box in each of its four corners.

To Select a Dialog:

> Click the cursor within the dialog. Be sure to click on a position that isn't occupied by a control.

To Select a Control:

> Click the cursor anywhere within the control. Be sure the cursor is well within the control. If it's too close to one of the control's borders, you may inadvertently select the dialog instead.

The Dialog Editor lets you select multiple controls so you can use the alignment tools to adjust their positions, or collectively move them.

To Select Multiple Controls:

Select the first control in the regular way, then hold down the SHIFT key and click each additional control you want to select.

Alternatively,
> Position the cursor near the outside corner of the set of controls you want to select.
> Click and drag the mouse until you form a rectangle around all controls you want to select.
> Release the mouse button.

De-selecting a Dialog or Control

Do any one of the following to de-select all items that are currently selected:

- Click the mouse button outside of the dialog box, but within the editing area.
- Select another item.
- Create another control. After it's created, the new control is automatically selected.

Moving a Dialog

To move a dialog:

> Position the cursor on any of the dialog's borders. The cursor changes from an arrow to a hand when it's on the border.

> Click and drag the dialog to its new location.

> Release the mouse button.

When you move a dialog, all its controls move with it. Each control remains in its relative position within the dialog.

Moving a Control

To move a control:

> Determine whether you want to move one control or multiple controls. To move only one control, you can skip this step. To move multiple controls, select all the controls you want to move.

> Position the cursor inside the control, then click and hold down the mouse button. The cursor changes from an arrow to a hand when you click the mouse button.

> Drag the control to its new location, then release the mouse button.

When you move a control, only the selected control moves. If multiple controls are selected, they all move together, maintaining their positions relative to each other.

If you've selected multiple controls, position the cursor on any one of them. It doesn't matter which one.

Changing the Size of a Dialog or Control

To change an item's size:

> Select the item you want to size. To size multiple controls, select each control. A sizing box appears in each selected item's corners.

> Position the cursor over one of the sizing boxes. When correctly positioned, the cursor changes to a cross with arrow heads.

> Click the left mouse button and hold it down.

> Keeping the button depressed, drag the mouse in any direction. This shows a rectangle of varying size. As long as you keep the left button depressed, you can continue to adjust the size of the rectangle.

> When you have the size rectangle you want, lift your finger off the left mouse button.

When you change a dialog's size, the sizes of its controls don't change. Each control retains its current size and position. Thus, the minimum size you can make a dialog is determined by the number, size, and position of controls within it.

When you change a control's size, only the selected control is affected. If multiple controls are selected, they all change size proportionately.

Restricting Movement or Sizing to One Dimension

You can restrict the movement of the mouse to only one dimension when you

- Create a new dialog or control.
- Change the size of a dialog or control.
- Move a dialog or control.

To restrict mouse movement to a single dimension:

> Hold down the SHIFT key, then drag the mouse either horizontally or vertically. The direction you choose restricts the mouse movement to that direction.

Setting Attributes for a Dialog or Control

Attributes are the characteristics defined for a dialog or control. Each dialog and control in the Dialog Editor has a corresponding attributes box. Through the attributes dialogs, you can

- Define attributes or styles
- Assign a title to a captioned dialog
- Assign text to a control
- Assign Fonts to a Dialog's Text
- Change an Item's Coordinates
- Change the ID Number of any Control

To define dialog or control attributes:

> Open the item's attributes box.

> Edit the attributes you want to change.

> To advance to the styles screen for a control, click the **Styles** button. The styles vary according to the control whose attributes you're defining.

Use either of two methods to open an item's attributes box:

- Double-click inside the item. For a dialog, be sure to double-click over a position that isn't occupied by a control.

- Select a dialog and choose **Attributes** from the **Dialog** menu, or select a control and choose **Attributes** from the **Controls** menu.

To by-pass the attributes box and go directly to the style dialog, press the ALT key when you double-click in the first step.

When the Styles dialog opens, an item's current styles are selected in the check boxes and radio buttons. You can change the styles by selecting alternative buttons. For example, you can change a radio button to a check box by selecting Check Box in the Styles dialog. Or you can change a captioned dialog to a standard dialog by selecting Standard Dialog from the Dialog Attributes box.

Assigning a Title to a Captioned Dialog

To assign a title to a captioned dialog box, type the text in the Title field of the Dialog Attributes box.

Assigning Text to a Control

To assign text to a control, type the text in the Text field of the control's attributes box. The different types of control display the text in different places:

Buttons. The Text field determines the text displayed within or beside the button.

Edit fields. The Text field determines the default text displayed when the dialog first appears.

Static text. The Text field determines what text is displayed in the static field.

Group boxes. The Text field is used to specify the title for the group.

For remaining controls, you can assign text, but the text isn't displayed in the application.

Assigning Fonts to Text

To assign fonts to the text within a dialog:

> Open the Dialog Attributes box.

> Click the **Font** button on the attributes box. This opens a dialog that has two combo boxes for setting fonts: one determines the font type, the other determines the point size.

> To change the font type, click the arrow to the right of the Font combo box. This opens a drop down list box so you can select a font that's installed on your machine. Alternatively, type a font type in the combo box's edit field. The font you specify doesn't have to be installed on your machine or listed in the list box.

> Follow the same procedure for specifying a point size in the Point Size combo box.

The font applies to all controls within the dialog. However, the font doesn't have to be the same font used by other dialog's within your application.

Changing an Item's Coordinates

In the Dialog Attributes box, the (x,y) field shows the x and y coordinates of the dialog's upper-left corner. These coordinates represent the dialog's position within the window that displays it in your application. They also correspond to the dialog's position within the Dialog Editor. The (cx,cy) field shows the dialog's width and height.

In the attributes box for controls, the (x,y) field shows the coordinates of the control's upper-left corner within the dialog. The (cx,cy) field shows the control's width and height.

The Dialog Editor automatically calculates the (x,y) and (cx,cy) coordinates for you, based on the item's current position.

- You can change an item's position by changing the numbers in these fields. You must specify integer values.

- Alternatively, move the item in the Dialog Editor by dragging it or using the alignment tools to adjust control positions. This automatically updates values in the (x,y) and (cx,cy) fields.

A horizontal unit in the (x,y) or (cx,cy) fields is 1/4 of the dialog base width unit. A vertical unit is 1/8 of the dialog base height unit. The current dialog base units are calculated as a function of the width and height of the current system font. For a more detailed explanation of dialog base units, see Petzold's *Programming Windows*.

Changing Control ID Numbers

Controls within a dialog are identified by the ID number defined for them in the Item ID field in their attributes boxes. The ID must be numeric, though you can define a symbol for it in a header file.

You can change the control's ID number in the Item ID field. To show the ID numbers for all controls in the dialog, choose **Show Item Id** on the Dialog menu.

For a new resource, the Dialog Editor automatically assigns ID numbers to each control you create. The controls are numbered consecutively as they're created, using positive integers that begin with 101.

For an existing resource, the controls already have ID numbers and new controls you create are assigned an ID number that's 1 higher than the highest number for existing controls.

The ID number for each control must be unique within the dialog box. However, the ID's can be the same as those used in other resources -- including other dialog boxes defined for the application. For example, two different dialog boxes in the same application can use ID numbers **101**, **102**, and **103** for their controls, and a menu resource in the application can use these ID's for its menu items. This is because Windows interprets the ID within the context of the resource currently being used in the application.

Setting Tab Order for Controls

The tab order determines the order the input focus moves among controls in an application when you press the tab key. To set the tab order for controls, choose **Set Tabstop** from the **Controls** menu.

Defining Logical Groups for Controls

A logical group defines a set of controls within which you can move input focus by pressing the arrow keys. To define logical groups, choose **Start Group** from the **Controls** menu.

Dialog Tool

By default, a captioned dialog box is automatically drawn when you create a new file in the Dialog Editor. However, if you should delete that dialog box -- accidentally or intentionally -- the Dialog tool lets you draw a new one.

Use the Captioned Dialog tool to create a captioned dialog box. Create a captioned dialog when you want to be able to drag the dialog within the application.

Use the Dialog tool to create a standard dialog. A standard dialog box is a dialog that doesn't have a title bar. Create a standard dialog when you don't want to be able to drag the dialog within the application.

Once a dialog is created, you can create its controls.

You can change the dialog to a standard dialog in either of two ways:

- Double-click the Dialog tool. This automatically changes the style of the dialog box currently displayed in the editing area.

- Change the dialog's style in its attributes box.

Buttons

- Push buttons
- Check boxes
- Radio buttons

Push Buttons

Push buttons are used to trigger an immediate action, without maintaining an on/off status. The Resource Toolkit lets you create

Standard buttons

Default buttons

Owner-draw buttons

Button Tool

The Button tool creates a standard push button. A standard push button is a button that has to be selected before it can be activated. Create a standard button for each action that isn't the dialog's default action.

In the Dialog Editor, the Button tool is selected by default. To create a standard button, you don't have to select the Button tool before you begin dragging the mouse.

Default Tool

The Default tool creates a default push button. A default push button is a button that is automatically selected when the dialog is open. This means you can simply press ENTER to activate it.

You should define only one default push button for each dialog.

Owner Draw Tool

The Owner Draw tool creates an *owner-draw* button. An owner-draw button is a button whose behavior is different from the standard behavior of push buttons, check boxes, and radio buttons.

The advantage of creating owner-draw buttons is that they're not limited to displaying text. For example, the buttons you press in the Resource Manager to start an editor are owner-draw buttons. These buttons display bitmaps rather than text.

With owner-draw buttons, the application takes responsibility for the button's appearance. When the button is selected, the parent window is notified and also receives a request to paint, invert, or disable the button. You need to define the additional support for these functions in the application's source code.

Check Boxes

Use the Check Box tool to provide one or more options that can be turned on or off. For example, a File Save dialog might have a Save Backup check box. If the box is checked, the application backs up the current file before saving new changes. If the box isn't checked, the application saves changes without first making a backup file.

To enclose the check boxes so they appear as a visual group on the screen, use the Group Box tool.

Radio Buttons

Use the Radio Button tool to provide a set of options, *only one* of which can be selected. For example, a drawing program that can save a graphic image in several formats might use radio buttons to let you select only one of those formats for saving the current image.

All the radio buttons from a set must be defined as a logical group. To define controls as a logical group, choose **Start Group** from the **Controls** menu.

To enclose the radio buttons so they appear as a visual group on the screen, use the Group Box tool.

Edit Controls

Edit controls are used primarily for text entry. They can also display default text in the edit field. The Resource Toolkit lets you create the following styles of edit control:

Single line edit.

Multiple line edit.

Multiple line edit that allows vertical scrolling.

Multiple line edit that allows vertical and horizontal scrolling.

Static Text

Use the Text tool to display static text. For example, the text might mention the consequences of an action or ask a question. Or it might label an edit control.

The Text tool is a multi-style tool that provides three formatting options:

Left-justified text.

Right-justified text.

Centered text.

Group Box

Use the Group Box tool to draw a titled border around a set of push buttons, check boxes, or radio buttons so they're displayed as a visual group.

Group boxes enclose controls to visually associate them. However, they *do not* define controls as a logical group. A logical group defines a set of controls within which you can move input focus by pressing the arrow keys. To define controls as a logical group, choose **Start Group** from the **Controls** menu.

List Boxes

Use the List Box tool to create a list box. A list box lets you select an item from among a list of text items. For example, you might create a list box that permits selection of one file name from a list of file names.

If text in the box is too long to be displayed all at once, list boxes have a built-in scroll bar that allows vertical scrolling.

Customizing List Boxes

Customizing List Boxes

To create a list box whose behavior is different from the standard list box, you can create an *owner-draw* list box. Owner-draw list boxes have only the general characteristics of standard list boxes, so you need to define their functionality in the source code.

Whereas standard list boxes draw their own contents, owner-draw list boxes do not. The application takes responsibility for the visual appearance of the list box. For example, an owner-draw list box can be used to display a series of bitmaps rather than text strings.

To define an owner-draw list box:

> Create a standard list box and open its style box.

> Choose either Owner Draw Fixed or Owner Draw Variable:
- Fixed: Items in the list box must all be the same height.
- Variable: Items in the list box vary in height.

Open the style box in either of two ways:

- Press the ALT key and double-click the control.

- Select the control, then choose **Attributes** from the **Controls** menu. On the attributes box, click the **Style** button.

Combo Boxes

Combo boxes combine the features of an edit box and a list box -- they let you select an item from among a list of items, but they also let you type a selection into an edit field.

You can create the following styles of combo boxes:

Simple combo box

Drop down combo box

Drop down list combo box

Customized combo box

To create any style of combo box:

> Choose the Combo Box tool and draw the box. This creates a simple combo box. If this is the style you want, you're finished. Otherwise:

> Open the combo box's style box and choose the style of combo box you want.

A simple combo box always displays both its features: the edit area and the list box. The edit area behaves just like an edit control t lets you enter and edit text. The text need not match an item from the list in the list box. If it does, the corresponding list item is selected.

As an alternative to typing text into the edit area, you can select an item from the list box. The list box in a combo box behaves just like a standard list box.

A drop down combo box behaves like a simple combo box, except that its list area isn't displayed when it first opens. To display the list area, you click the down arrow that's provided to the right of the editing area.

Drop down combo boxes are useful when you want to fit a lot of controls into a small area. Initially they take up only as much space as the editing area needs. When the list box is opened, it overlaps the dialog and doesn't require additional space.

A drop down combo box is also useful when you want to provide a list of selections, but allow an alternative selection. For example, you might use a drop down combo box in a dialog for opening or saving disk files. You can use the list box to search disks for an existing file name, or use the edit field to enter the name of a new or existing file.

A drop down list combo box is similar to a drop down combo box. It has a list area that drops down when needed and retracts when not needed. The boxes differ, however, in the behavior of their editing areas. Whereas the editing area in a drop down combo box lets you enter any text, the editing area of a drop down list combo box lets you enter only an item from the list box. You can select the item from the list box, or type it into the editing area, but you can't enter an item that isn't listed in the list box.

Drop down list combo boxes are useful in cases where only the listed selections are acceptable. For example, even though a computer might be linked to several printers, you can only choose one among the several to print to at any given time.

To create a combo box whose behavior is different from the pre-defined combo boxes, you can create an *owner-draw* combo box. The advantage of creating owner-draw combo boxes is that they're not limited to displaying text strings. For example, they can display a series of bitmaps.

Owner-draw combo boxes have only the general characteristics of standard combo boxes, so you need to define their functionality in the source code. Whereas pre-defined combo boxes draw the contents of their list boxes, owner-draw combo boxes do not. The application is responsible for the visual content of the list box portion of the combo box control.

To define an owner-draw combo box, choose either Owner Draw Fixed or Owner Draw Variable on the combo box's Styles dialog:

- Fixed: Items in the list box must all be the same height.
- Variable: Items in the list box vary in height.

Scroll Bars

Use the Scroll Bar tool to create a stand-alone scroll bar in the dialog. Edit controls, list boxes, and combo boxes have their own, attached scroll bars and don't need stand-alone scroll bars.

A stand-alone scroll bar is useful in a dialog that needs more than one vertical or horizontal scroll. For example, the Bitmap Editor uses color palettes for drawing bitmaps. You can edit a color in the palette by changing its mixture of red, green, and blue. To let you change the mixture, the editor provides a dialog with three stand-alone scroll bars.

The Scroll Bar tool is a multi-style tool that lets you create

A scroll bar with standard width. You cannot change the width of a standard scroll bar.

A scroll bar with adjustable width.

Icons

Use the Icon tool to create a box within which you can place an icon. Windows automatically sizes this box, so all you have to do is position it where you want the icon to appear within the dialog.

To specify the icon resource to display in the box, open the box's attributes dialog and type the icon resource's name in the Text field.

Because the Tools palette has more tools than can be displayed in a maximized window, the Icon tool isn't displayed on the default palette. To display the Icon tool, double-click the Scroll Bar tool until it changes to the Icon tool.

Custom Controls

Use the Custom Control tool to create a box within which you can place a customized control, giving it characteristics that aren't available with any of the control tools.

A custom control must have a dynamic-link library (DLL) that defines the window procedure for the control, as well as the functions that interact with the Dialog Editor.

Because the Tools palette has more tools than can be displayed in a maximized window, the Custom Control tool isn't displayed on the default palette. To display the Custom Control tool, double-click the Scroll Bar tool until it changes to the Custom Control tool.

Align Left

Align Left aligns all selected controls by the standard's left border. The vertical position of each individual control is unaffected.

Align Right

Align Right aligns all selected controls by the standard's right border. The vertical position of each individual control is unaffected.

Align Top

Align Top aligns all selected controls by the standard's top border. The horizontal position of each individual control is unaffected.

Align Bottom

Align Bottom aligns all selected controls by the standard's bottom border. The horizontal position of each individual control is unaffected.

Center on Horizontal

Center on Horizontal aligns the center of all selected controls with the standard's horizontal axis. The horizontal position of each control is unaffected.

Center on Vertical

Center on Vertical aligns the center of all selected controls with the standard's vertical axis. The vertical position of each control is unaffected.

Spread Horizontally

Spread Horizontally evenly spreads the selected controls within the dialog's width.

Spread Vertically

Spread Vertically evenly spreads the selected controls within the dialog's height.

Make Same Size

Make Same Size makes all selected controls the same size as the standard.

Header Editor

Header Files

Symbolic Constants

Keys Used in the Header Editor

Button Commands

Put

To End

To Top

Header Editor: Basics

The Header Editor lets you create and edit symbolic constants, called symbols, that can substitute for the ID numbers Windows uses to identify resources and controls. Once the symbols are defined, you can load resources from your source code by referring to their defined symbols rather than their numeric values.

The Header Editor lets you define the symbols while you're creating or editing the resources they substitute for, then save them in a header file. The header file can be a C-style header (H) file, or a Pascal include (INC) file.

Header files don't have to be created or edited with the Header Editor. They're text files that can be edited separately with any text editor. However, the Header Editor supports only restricted syntax in header files; its only purpose is to associate symbols with resource ID's. The syntax is restricted whether the file is an H file or an INC file.

Symbols from the header file become global constants when they're compiled. Each symbol must therefore be unique within the application.

For C-style header (H) files, the Header Editor supports only **#define** statements that define constants in decimal or hexadecimal format, though the file can also contain C-style comments beginning with **/*** and ending with ***/**. If you try to open an H file containing anything else, the Header Editor issues an error message and opens a new, blank H file.

If you open an H file containing comments, the comments are ignored when the file is opened. If you save an H file with the Header Editor, all comments in the file are deleted and only the **#define** statements are saved.

For Pascal include (INC) files, the Header Editor supports only **const** declaration statements that define constants in decimal or hexadecimal format. The file cannot contain anything else, not even comments. If you try to open an INC file containing anything else, the Header Editor issues an error message and opens a new, blank include file.

Starting the Header Editor

Because the Header Editor lets you define symbols while you create or edit the resources they substitute for, you never start the Header Editor by itself. Rather, you begin editing the resource, then start the Header Editor from within the resource's editor. For example, to define symbols for string resources, you first start the String Editor, then from within the String Editor, start the Header Editor.

The **File** menu in the Accelerator, Dialog, Menu, and String editors lets you create a new header file or open an existing header file. Once you select either **New Header** or **Open Header**, the Header Editor is automatically started and the file is opened. Only one header file at a time can be open within each resource editor.

Creating New Header Files
Opening Existing Header Files
Saving Header Files

The Header Editor is a child window of the editor used to start it. Thus, if you minimize or close the associated editor to an icon, the Header Editor is also minimized or closed.

Display Format for Values

Before working with a header file, determine whether you want the resource ID values to be displayed in decimal format or hexadecimal format. The radio buttons in the top-right corner of the Header Editor let you select the appropriate format. Decimal format is the default.

Order of Symbols in the Header File

The Header Editor searches for values from the top of a header file to the bottom, displaying the first symbol it finds for a value. This means the order of symbols in the header file may determine which symbol is displayed for a value in the associated editor.

For example, if you define symbol CW_FILE_NEW for menu item 101 in the Menu Editor, and SAV_BUTTON is defined for value 101 higher in an associated header file, SAV_BUTTON is displayed in the Menu Editor as the symbol for menu item 101. To display CW_FILE_NEW as the symbol, you must move it above SAV_BUTTON in the header file.

Creating New Header Files

To create a new header file, whether a C-style header (H) file or a Pascal include (INC) file, choose **New Header** from the **File** menu in the Accelerator, Dialog, Menu, or String editor. This opens an empty file so you can define new symbols for your application.

Typically, you would create only one new header file for each application you develop. This is because it's easier to ensure the symbols are unique within a single file than it would be to ensure they're unique within multiple files. However, determining the number of header files to create for an application is a design decision. You can create as many as you need.

A defined symbol becomes a global constant in the application source code. Thus, if you include more than one header file in your application, the symbols in each file must be unique. Any particular symbol can't be defined in more than one file, unless it's associated with the same value.

For example, if symbol CW_FILE_NEW is associated with value 101 in one header file, no other header file created for the same application can contain symbol CW_FILE_NEW, unless it's associated with value 101.

Opening Existing Header Files

To open an existing header file, choose **Open Header** from the **File** menu in the Accelerator, Dialog, Menu, or String editor.

Syntax for the file you open is limited, regardless of whether you open a C-style header file (H) or a Pascal include file (INC).

Saving Header Files

To save a header file, click the **Save** or **Save As** button in the Header Editor:

- **Save** saves the file under its original name.
- **Save As** saves the file under a new name. **Save As** can also save the file under a new format. For example, if you open a Pascal include (INC) file, you can save it as a C-style header (H) file.

Caution: If you open a C-style header file (H) containing comments, the comments are deleted from the file when it's saved. Only the #define statements are saved.

If you assign the same file name to the header file as you assign to the application's resource (RES) file or executable (EXE) file, you'll make it easier to associate the header file with the application.

Editing a Symbol in the Header Editor

To edit a Symbol in the Header Editor:

> Make the Symbol field active and type the symbol into the field. For an existing symbol, you can select the symbol from the list box, and it will automatically be copied into the Symbol field. Once you type the symbol, make the Value field active and type the resource's value.

> Click **To Top**, **Put**, or **To End** to position the symbol in the header file.

Symbol Field

The first character of a symbol must be alphabetic. All additional characters can be alphanumeric. The maximum length of the symbol depends on the source code language.

The symbol must be unique. If you enter the name of an existing symbol, the value you specify in the Value field replaces the value previously defined for the symbol -- even if the existing symbol isn't currently visible in the list box.

Value Field

Enter the value for the resource the symbol will substitute for. For example, if you're defining a symbol for a menu item with ID 101, enter 101 as the value to associate with the symbol. The value must be numeric.

To enter a long number, suffix the number with an L.

Values don't have to be unique within a header file because different resources in an application can use the same ID values.

For example, a menu item on a menu might have value 101, and a button on a dialog might also have value 101. Windows won't confuse the two values because it interprets the values within the context of the resource being used.

You can't enter macros or expressions in the Value field. The Header Editor ignores any lines it doesn't understand.

To Top

To Top places the symbol at the top of the header file. This ensures that the symbol will be displayed in the associated resource editor, even if another symbol in the header file is defined for the same value.

Put

Put can be used to change the value of the currently selected symbol, without changing the symbol's position in the header file.

For example, if you select a symbol from the table, its symbol and associated value are displayed in the Symbol and Value fields.

If you change the value and click on **Put**, the new value replaces the previous value, without changing the symbol's position.

If you type a unique symbol into the Symbol field and click on **Put**, **Put** functions the same as **To End**.

To End

To End places the symbol at the end of the header file. This ensures that the symbol won't be displayed in the associated resource editor if another symbol is defined for the same value. You might do this to test whether the value is unique in the header file, or to prevent the new symbol from affecting your work in the associated resource editor.

For example, if you just defined symbol CW_FILE_NEW for menu item 101 in the Menu Editor, you could place the symbol at the bottom of the header file, then see whether CW_FILE_NEW is the symbol displayed for menu item 101 in the Menu Editor. If it is, you know value 101 is unique in the header file. If it isn't, you know there's at least one other symbol defined for value 101 in the file.

Or you might be working in the String Editor when you decide to define symbol CW_FILE_NEW for menu item 101. If you also have a string with the defined symbol value 101, you want to be sure the symbol for string 101 is above the symbol for menu item 101 in the header file. Otherwise, the String Editor will display the wrong symbol for the string resource that has value 101.

Editing a Symbol: Accelerator, String, and Menu Editors

To edit a Symbol in the Accelerator, String, and Menu editors:

> Select the Symbol field from the appropriate row in the table, then add or edit the symbol.

> Press ENTER, or use the mouse or keyboard keys to move the field indicator to another field. This opens the Undefined Symbol dialog.

> Type the value you want the symbol to substitute for. This will most likely be the value for the current Symbol field, but the dialog lets you change your mind and associate the symbol with a different value.

> Click **Put Top** or **Put End** to position the symbol in the header file. **Put Top** and **Put End** are equivalent to **To Top** and **To End** in the Header editor.

Editing a Symbol: Dialog Editor

To edit a Symbol in the Dialog Editor:

> Open the control's Attributes box by double-clicking the control. In the Attributes box, the Item ID field displays the control's value. If there's a symbol defined for that value in the header file, the symbol is displayed instead.

> Type the new symbol into the Item ID field, then click the **OK** button. This opens the Undefined Symbol dialog.

> Type the value you want the symbol to substitute for, then click **Put Top** or **Put End** to position the symbol in the header file. **Put Top** and **Put End** are equivalent to **To Top** and **To End** in the header file.

Note. You can't enter alpha characters in the Item ID field unless a header file is open.

Deleting a Symbol

To delete a symbol, select it in the Header Editor's list box, then click the **Delete** button. This deletes both the symbol and its associated value from the header file.

Be sure to select the correct symbol. The Header Editor doesn't open a dialog to confirm you want to delete it.

Moving a Symbol

To move a symbol within the header file, select it in the Header Editor's list box, then click either **To Top**, or **To End**.

Navigating the Header Editor

TAB	Moves to the next field or button.
SHIFT+TAB	Moves to the previous field or button.
Up arrow	In the list box, selects the symbol in the row above the currently selected symbol.
	In the Symbol or Value fields, moves the cursor one character to the left.
Left arrow	Same as up arrow.
Down arrow	In the list box, selects the symbol in the row below the currently selected symbol.
	In the Symbol or Value fields, moves the cursor one character to the right.
Right arrow	Same as down arrow.
HOME	Selects the first symbol in the header file.
END	Selects the last symbol in the header file.

Typing in the Symbol or Value Fields

To replace the text in either the Symbol or Value fields, select the text, then begin typing or press the DELETE key.

To preserve the text in either the Symbol or Value fields, move the cursor in the field, using the mouse, an arrow key, or the HOME or END key. You can then edit some portion of the text.

SHIFT+right arrow	Selects the character one position to the right.
SHIFT+left arrow	Selects the character one position to the left.
SHIFT+END	Selects all characters from the current cursor position to the end of the line.
SHIFT+HOME	Selects all characters from the current cursor position to the beginning of the line.

Menu Editor

Fields: Menu Editor

Break
Check
Help
Menu Text
Separator
Style
Symbol
Value

Files: Menu Editor

The Menu Editor can edit menu resources from resource (RES) files, executable (EXE) files, and dynamic link libraries (DLLs).

The Menu Editor can save a menu resource directly into a new or existing RES file, or into an existing EXE file or DLL. It can also save the same menu specification into a resource script (RC) file.

The Menu Editor can open a header file (also called an include file) to correlate symbolic constants from the file with the menu resource. This adds a Symbol column to the menu table, and opens the Header Editor. You can edit symbol values with the Header Editor, or edit them directly from the menu table.

Menu Text

Menu Styles and Attributes

Keys Used in the Menu Editor

Menu Editor: Basics

The Menu Editor lets you create and edit menu resources. This means you can determine both the content and appearance of any menu in a Windows application.

The Menu Editor also lets you interactively test changes you make to a menu, without having to compile and run the application.

Menu Table

When you first open the Menu Editor, it contains a menu table with two columns of edit fields.

The Menu Text column lets you enter or edit menu text, which is the text displayed for menu items.

The Value column lets you enter or edit a value for the current item.

If a header file is open, an additional Symbol column shows the symbolic name, if any, for each menu item.

Movement Buttons

Above the menu table are four movement buttons.

The Left and Right buttons let you change a menu item's level in the menu hierarchy.

The Up and Down buttons let you change a menu item's row in the Text field, and therefore its position on a menu.

Style and Attribute Fields

Below the menu table is a set of fields that let you define menu styles and attributes. These fields let you add the following a menu:

Separator lines
Check marks
Text styles
Columns
A help item

Defining Menu Text

In the menu table, the Menu Text and Value fields, and when a header file is open, the Symbol field, are fields you can edit.

To define menu text, enter the text in the Menu Text field, then assign a value to the new menu item.

Break Field

The Break field lets you use breaks in the menu to align menu items in columns.

Check Field

The Check field lets you use a check mark to indicate a menu item's status.

Help Field

The Help field lets you assign the help attribute to a menu item.

Menu Text Field

The Menu Text field displays the menu text that appears in the actual menu. Enter the text exactly as you want it to be displayed on the menu. The text can contain any alpha-numeric characters, and can be up to 255 characters long. Windows uses the & character to define and underscore activation keys, and \t for a tab.

Separator Field

The Separator field lets you add a separator line to a popup menu.

Style Field

The Style field lets you define a style for a menu item's text.

Symbol Field

The Symbol field displays symbol whose ID number in the header file matches the value for the current menu item. You can use this symbol in the application's source code to refer to the menu item.

Value Field

The Value field displays the ID number for the menu item specified in the Menu Text field. This is the number the item returns when it's selected, and it's the number used to identify the item in the application's source code. As an alternative to using the value, you can define a symbol for the menu item in the Symbol column.

Defining Menu Levels

To create a new popup level, you indent text, using the right movement button.

To delete a popup level, use the left movement button to eliminate its indentation level from the menu text.

To move an item up or down in the menu hierarchy:

> Activate any of the item's fields.

> Click the left movement button to move it up in the hierarchy, or the right movement button to move it down.

Alternatively, hold down the ALT key a press the left or right arrow on your keyboard.

Either method shifts text for the current item left or right one place.

If the item has popup menus under it, all the popup menus are moved with it.

Changing the Position of Menu Items

To change the position of a menu item, activate any of the item's fields, then use the up and down movement buttons.


Click the up movement button to move it up in the order.


Click the down movement button to move it down.

Alternatively, hold down the ALT key and press the up or down arrow on your keyboard. Either method moves text for the current item up or down one place.

An item can be moved only to a position at its current level in the menu hierarchy. If there isn't a position available at its current level, the item isn't moved.

If the item has popup menus under it, all the popup menus are moved with it.

Setting an Activation Key

To define an activation key:

> Type an ampersand (&) in front of the corresponding letter in the menu text. For example, the text **&Save** defines S as the activation key for **Save**. The text **Save &As** defines A as the activation key for **Save As**.
>
> The & doesn't have to precede the first letter in a word. For example, **Cu&t** defines T as the activation key for **Cut**.

In the actual menu, the ampersand is translated to an underscore under the activation key.

Inserting Tabs in Menu Text

You can insert a tab mark in menu text by typing **\t** where you want the tab to go in the text. For example, in the text **Clear\tDel**, the tab will be inserted before the string **Del**.

Tabs are most useful when you're identifying accelerator keys for a menu option. The tab ensures that the text identifying accelerator keys is aligned in a column.

Windows automatically determines the appropriate width and spacing for the column, based on the menu text.

Adding Separator Lines to a Popup Menu

To add separator lines to a menu:

> Insert a blank row in the menu table where you want the separator line to appear. To do so

- Determine which item will immediately precede the separator line, then activate any of its fields.

- Press CTRL+ENTER. This inserts a blank row beneath the current item.

> Enter Yes in the Separator field. This defines the row as a separator line. All the other style and attribute fields for the row become inactive.

A separator line is a horizontal line that divides menu items into visual categories. Use separator lines to separate menu items into functional categories.

Separator lines are optional on a menu and serve no function beyond providing visual categories for the menu items.

Using a Check Mark to Indicate an Item's Status

Items on a popup menu can provide independent options, each of which can be turned on or off. They can also provide a set of mutually exclusive options, only one of which can be on at any given time.

In these cases, you can use a check mark to indicate whether an option is in effect.

To use a check mark for a menu item, enter **Yes** in its Check field.

Defining a Style for a Menu Item's Text

The Menu Editor lets you use different styles of text to reflect a menu item's current status. For example, you can display **Cut**, **Copy**, and **Paste** as gray text when these options are unavailable.

To specify a menu item's style, enter ACTIVE, INACTIVE, or GRAY in its Style field.

The source code defines the conditions under which an item is active, inactive, or grayed. When conditions are met in the application for an inactive or grayed menu item to become available, the source code changes the item's style to active.

ACTIVE indicates a menu item is active and returns its defined value when selected. An active item is displayed in the normal style.

INACTIVE indicates a menu item is initially inactive and doesn't return its defined value when selected. Inactive items are displayed in the normal style.

Inactive items can be useful as you develop your application. For example, if you don't have code written for an item's function, but you still want to display it as it will appear in the final application, you can define the item as INACTIVE. When you select the inactive item in the application, it won't cause problems because it doesn't return a value.

GRAY indicates the item is initially grayed and doesn't return a value when selected. Grayed items are displayed as gray text, which flags them as currently unavailable.

Aligning Menu Items in Columns

To align menu items in columns, you can define breaks for the items.

To define breaks on a popup menu, determine which item should be the first item in a new column, then enter NONE, COLUMN, or BAR in its Break field.

Items below the break item on the menu are aligned in its column, unless you define another break.

Windows lets you define two types of breaks: a *column* break and a *bar* break.

A column break aligns items in a column.

A bar break aligns items in a column, and also separates columns with a vertical bar.

NONE indicates the current item doesn't begin a new column.

COLUMN indicates the current item begins a new column. All text below this item in the Text field is aligned in the new column, unless you set another break.

BAR works the same as COLUMN, except that it inserts a vertical bar between columns to further emphasize the break.

Assigning the Help Attribute to a Menu Item

The Menu Editor lets you assign the Windows help attribute to a top-level item on a menu. Only one item can have the help attribute, and the item must be at the highest level.

To assign the help attribute to a menu item, enter Yes in its Help field.

Items positioned below the help item on the menu are also right-justified on the menu bar. To avoid this, make the help item the last top-level item in the menu.

The help attribute determines that the item will be right-justified on the menu bar. This is useful when you want to separate the Help menu from other menus on the menu bar.

Testing Menus

Above the menu table is a special test window. The test window lets you interactively test your menu.

Use the menus in the Test Window exactly as you would use them in the actual application.

When you make a selection in the Test Window, its defined value and symbol are displayed in the window's client area.

The Test Window is automatically refreshed each time you click it. Thus, every time you use it, it reflects the current status of your work.

Navigating a Menu Table

CTRL+ENTER	Adds a new row to the menu table.
TAB	Moves one column to the right
SHIFT+TAB	Moves one column to the left.
CTRL+TAB	Moves the active field between the menu table and the style and attribute fields.
Up arrow	Moves the active field up one row.
Down arrow	Moves the active field down one row.
Right arrow	Moves the cursor one position to the right in the active field.
Left arrow	Moves the cursor one position to the left in the active field.
HOME	Moves the cursor to the beginning of the active field.
END	Moves the cursor to the end of the active field.

Typing in the Menu Table

If you activate a field from the Menu Text or Value columns, all text in the field is selected.

- To replace the text, begin typing, or press the DELETE key.
- To preserve the text, move the cursor in the field, using the mouse, an arrow key, or the HOME or END key. You can then edit some portion of the text.

SHIFT+right arrow	Selects the character one position to the right.
SHIFT+left arrow	Selects the character one position to the left.
SHIFT+END	Selects all characters from the current cursor position to the end of the line.
SHIFT+HOME	Selects all characters from the current cursor position to the beginning of the line.

Selection Fields in the Menu Table

The style and attribute fields below the menu table let you select one option from a set of available options:

To cycle through available selections in the field, press the SPACEBAR.

Alternatively, type the first letter of the selection. For example, in a Yes/No field, if Yes is the current value, press the N key to cycle to No.

String Editor

Fields: String Editor

String Text

Symbol

Value

Keys Used in the String Editor

Menu Commands

File

Edit

Header

Options

String Editor: Basics

The String Editor lets you create and edit string resources. String resources are strings used by an application for the text displayed in

- Error messages.
- Status messages or other general system messages.
- Caption text in a window.

The string resources are loaded as needed from an executable file. The more strings you define as resources, the easier it will be to translate displayed text to another language, or customize the text for different uses.

The String Editor stores strings in a *string table*. The maximum size of a string table permitted in Windows is 64K. There can be only one defined string table for an executable (EXE) file.

Files: String Editor

The String Editor can edit string resources from resource (RES) files, executable (EXE) files, and dynamic link libraries (DLLs).

The String Editor can save a string resource directly into a new or existing RES file, or into an existing EXE file or DLL.

The String Editor can open a header file (also called an include file) to correlate symbolic constants from the file with the string resource. This adds a Symbol column to the string table, and opens the Header Editor. You can edit symbol values with the Header Editor, or edit them directly from the string table.

String Text Field

The String Text field displays the string text that was inserted for the row. Enter the strings without quotation marks; if a string is enclosed in quotes, the quotes are treated as part of the string.

String text can be up to 255 characters long.

Symbol Field

The Symbol field displays the symbol whose ID number in the header file matches the value for the current string text. To load the string text into your application, you can refer to this symbol in the application's source code.

Value Field

The Value field displays the ID number for the string in the String Text field. To load the string text into your application, refer to this number in the application's source code. Alternatively, you can define a symbol for the string.

The String Editor automatically assigns values to the rows in a string table. The values are sequential integers, beginning with 0. Rows cannot be added or deleted from the table, nor can a row value be changed. Thus, you cannot activate a field in the Value column.

The ID values of string text can be the same as the ID's used in other resources, such as for the controls in a dialog box. This is because Windows interprets the ID value within the context of the resource currently being used in the application.

Windows loads string resources in 16-string segments. For example, when you ask for string 14, Windows loads strings 0 through 15.

To reduce the number of loads Windows has to perform, group strings that are common to a particular function into 16-string segments. This allows Windows to load related strings together, and also discard them together when the application no longer needs them.

Navigating a String Table

TAB	Moves one column to the right. If there isn't a column to the right, moves to the next row of the table.
ENTER	Same as TAB.
SHIFT+TAB	Moves one column to the left.
Up arrow	Moves the active field up one row.
Down arrow,	Moves the active field down one row.
Right arrow	Moves the cursor one position to the right.
Left arrow	Moves the cursor one position to the left.
HOME	Moves the cursor to the beginning of the field.
END	Moves the cursor to the end of the field.
PGDN	Pages down through screens in the string table, without skipping any screens.
CTRL+PGDN	Pages down through screens in the string table, omitting screens that don't have defined string text. For example, if the current screen displays rows **0** through **8** and value **999** is the next value to have defined string text, pressing CTRL+PGDN pages to row **999**.
PGUP	Pages up through screens in the string table, without skipping any screens.
CTRL+PGUP	Same as CTRL+PGDN, except that it pages up rather than down.
CTRL+L	Goes directly to a specified line. For example, to go directly to line **999**, press CTRL+L, then enter **999** in the resulting dialog's entry field.

Because you can't add rows to a string table, ENTER advances the cursor to the next row of the table. It doesn't insert a blank row or split string text.

Typing in the String Table

If you activate a field from the String Text column, or when a header file is opened, the Symbol column, all text in the field is selected.

- To replace the text, begin typing, or press the DELETE key.

- To preserve the text, move the cursor in the field, using the mouse, an arrow key, or the HOME or END key. You can then edit some portion of the text.

SHIFT+right arrow	Selects the character one position to the right.
SHIFT+left arrow	Selects the character one position to the left.
SHIFT+END	Selects all characters from the current cursor position to the end of the line.
SHIFT+HOME	Selects all characters from the current cursor position to the beginning of the line.

An activation key is a key used to select a menu item.

For example, assuming F is the activation key for **File** and S is the activation key for **Save**, you can choose **File Save** by pressing the ALT key to activate the menu bar, F to choose the **File** menu, and then S to choose **Save**. The activation keys for each menu work only when the menu is active.

Note. Windows automatically defines the ALT key as the activation key for the menu bar.

An executable (EXE) file contains all an application's resources plus compiled program code. This means EXE files store all resource types.

You can directly edit the resources in an EXE file, and save modified resources back into the file. This is useful for making cosmetic changes to applications, even if you don't have the source code, RES, or RC files.

A resource (RES) file is a compiled RC file containing any type or combination of resources in binary format. You can directly edit RES files in the Resource Toolkit, and save modified resources back into the file.

You can also save a resource into a new RES file. Typically, you would create one RES file for each application, and store all the application's resources in that file.

To use the resources in your application, you have to add them to your application's executable (EXE) file. In the Resource Toolkit, you can accomplish this by using the Resource Manager to copy resources from the RES file into the EXE file.

A dynamic link library (DLL) is an executable module containing resources plus functions that can be called by Windows applications. A DLL is similar to a run-time library, except that it's linked to a program at run time rather than when conventional application files are linked. The library is activated when another module calls one of its functions, thus allowing it to be used by multiple applications or programs. This means common routines can be consolidated into a single module rather than duplicated in each program that uses them.

Routines in a DLL can be modified without relinking the DLL to the programs that call it. This contrasts with normal object libraries, which have to be relinked whenever their routines are modified.

A resource script (RC) file contains an application's resources in text format. RC files can be generated by the Resource Toolkit, or created with a text editor.

RC files identify resources by name or number, and define their attributes, styles, and other information relevant to the individual resources. To use the resources defined in the RC file, you have to compile the file to create a resource (RES) file, and then add the RES file to an executable (EXE) file.

RC files are useful for printed archives of your resources, or for compatibility with Microsoft's RC compiler. In most cases, they're not necessary since the Resource Toolkit lets you directly edit EXE and RES files, eliminating the need for script files.

A dialog resource script (DLG) file contains dialog resources in text format. DLG files can be generated by the Resource Toolkit, or created with a text editor.

DLG files can be included in an RC file using **rc include** statements, or can be manually inserted into an RC file.

DLG files are useful for printed archives of your dialog resources, or for compatibility with Microsoft's RC compiler. In most cases, they're not necessary since the Resource Toolkit lets you directly edit EXE and RES files, eliminating the need for script files.

Windows lets you assign symbolic names to the numbers used in an application. This lets you work with meaningful names rather than a confusing list of numbers.

To assign the symbolic names, you construct a header file to associate *symbols* with the numbers; the header file can be either a C-style header (H) file or a Pascal include (INC) file. You then let the language compiler and the resource editor translate the symbol back into the identifying number.

The Accelerator, Dialog, Menu, and String editors provide access to header files and also let you create or edit the symbols defined in those files.

A bitmap resource describes a picture but has no immediate window manager function. For example, a copyright dialog might display a product's logo. The logo identifies the product, but it doesn't serve a functional purpose in the dialog.

Cursors are specialized bitmaps that define the on-screen location of the mouse pointer.

To represent changes in the mouse's function, it's useful to use different cursors. For example, when the mouse is used to locate an insertion position for text, the cursor might be shaped like an I-beam. When the mouse is used to move an item on the screen, the cursor might be shaped like a hand.

Icons are specialized bitmaps that represent applications or actions within an application. For example, if you minimize a Windows application, such as the Resource Toolkit, an icon can be displayed on screen to represent the minimized program.

Resolution information for various device drivers is stored in the file WRT.DAT. This information is required for creating a new image for a cursor or icon.

To guarantee that each cursor or icon you create will look good on all display devices, you can create one image for each record in file WRT.DAT. This shouldn't be necessary, however. If an image isn't defined for a particular device, Windows selects the image that will look best on the current device.

Generally, you need one image for each unique cursor or icon dimension. If you create only 16-color images for icons, monochrome devices will dither all colors except black and white.

For more information on file WRT.DAT, see your user's guide.

Bitmap (BMP) files are stored in bitmap format and can contain only a single bitmap resource. Once the file is created, it can be referenced in an RC file.

Cursor (CUR) files are stored in bitmap format and can contain only a single cursor resource. Once the file is created, it can be referenced in an RC file.

Icon (ICO) files are stored in bitmap format and can contain only a single icon resource. Once the file is created, it can be referenced in an RC file. In addition, you can use it in the Windows Program Manager to change the icon displayed for an application.

A dialog box is a special window containing controls, such as list boxes, push buttons, and edit fields. Controls are the elements that compose the dialog's user interface.

Menu Commands

File

Edit

Header

Menu Commands

File

Edit

Header

File Menu

The **File** menu lets you create new files, work with existing files, or define resource attributes.

New

Open

Save

Save As

Save Into

Resource Attributes

Accelerator, Dialog, Menu, and String editors

New Header

Open Header

New

New clears the editor and opens a new, untitled file. Use it to clear the current resource out of the editor and enter a new resource.

- If the resource in the editor has been saved, the new file is opened immediately.
- If the resource in the editor hasn't been saved, a dialog opens to ask whether you want to save changes made to the file:
    - Click **Yes** to save the changes. This opens a dialog so you can name the file.
    - Click **No** to discard the changes.

Open

Open clears the editor and loads an existing file into it. Use it to clear the current resource out of the editor and retrieve contents from an existing resource.

When you choose **Open**, the File Open dialog pops up:

> Select the type of file you want by clicking the appropriate radio button in the File Type field. For example, to see a list of EXE files, click the EXE radio button.

> If the file you want isn't in the current directory, change to the directory where it's stored.

> In the Files box, double-click the file you want, or select it and click the dialog's **OK** button.

> Select the individual resource you want to work with.

If the resource in the editor has been saved, the existing file is opened immediately.

If the resource in the editor hasn't been saved, a dialog opens to ask whether you want to save changes made to the file:

- Click **Yes** to save the changes. This opens a dialog so you can name the file.ts

- Click **No** to discard the changes.

Save

Save saves the current resource under its original name and into its original file. Use it to save the resource, then continue working with it.

If the resource wasn't originally from a file but was created in a new file, **Save** works the same as **Save As**.

To save the resource under its original name but into an existing file that isn't the original file, use **Save Into**.

Save As

Save As saves the current resource under a different name. Generally, use **Save As** to create a new file. The types of files you can create are BMP (Bitmap Editor), CUR (Cursor Editor), DLG (Dialog Editor), ICO (Icon Editor), RC (Accelerator, Menu, and String editors), and RES files (all editors).

When you click **Save As**, the File Save As dialog opens so you can name the file.

> Select the type of file you want to save by clicking the appropriate radio button in the File Type field. For example, to save a RES file, click the RES radio button.

> To store the file in a directory other than the current directory, change the directory.

> Either create a new file, or replace a resource of the same name within one of the existing files that are displayed in the Files box.

> Click the dialog's **OK** button.

If you created a new file name, this creates the file on disk.

If you're replacing a resource from an existing file, this opens a dialog to confirm the replacement:

- **OK** replaces the existing resource.
- **Cancel** cancels the save operation.

Caution. If you click **OK**, the modified file is saved on disk, replacing the original file. Don't select **OK** unless you're sure you want to replace the resource in the existing file.

When saving a resource into a RES file, **Save As** lets you save the resource into a new or existing RES file. If you save the resource into an existing RES file, **Save As** functions the same as **Save Into**.

To create a new file:

> Position the cursor in the Filename field and type a name for the file. You don't have to include a file extension since the extension is taken from the File Type radio button.

To replace a resource in one of the files listed in the Files box:

> Select the file. This automatically copies the file's name into the entry field.

Save Into

Save Into saves the current resource into an existing resource (RES) file, executable (EXE) file, or dynamic link library (DLL). In addition, in the Bitmap Editor, **Save Into** can save the bitmap into an existing BMP file; in the Cursor Editor it can save a cursor into an existing CUR file; and in the Icon Editor it can save an icon into an existing ICO file.

Save Into cannot be used to save a resource into a new file. It saves into only an existing file: the resource's original file (if any) or a different file. To save a resource into its original file, it's quicker to use **Save**.

When you click **Save Into**, the File Save Into dialog opens so you can name the file:

> Select the type of file you want to save into by clicking the appropriate radio button in the File Type field. For example, to save into an EXE file, click the EXE radio button.

> If the file you want to save into isn't in the current directory, change the directory.

> Select the file you want. This automatically copies the file's name into the Filename field.

> Click the dialog's **OK** button. This saves the current resource into the selected file.

Saving a new resource into an EXE or DLL file doesn't enable the resource. You must define functionality for the resource in the source code.

In BMP, CUR, or ICO files, **Save Into** writes over the existing resource stored in the file, replacing it with the current resource.

RES, EXE, and DLL files, however, store multiple resources. Thus, you must assign a name to each resource you save into one of these file types.

If you assign the resource a unique name, the new resource is added to the file.

If you assign a name already used by another resource in the file, **Save Into** gives you the option of either replacing the existing resource, or preserving the existing resource by cancelling the save operation. You can then start over and assign a unique name to the current resource.

Resource Attributes

Resource Attributes opens the Resource Attributes dialog, which lets you define attributes for the current resource.

Select the attributes you want to apply to the current resource, then click **OK**. Attributes that don't apply to a particular editor are disabled in the dialog.

The defaults are **Load On Call, Moveable**, and **Discardable** for all resource types except bitmaps. For bitmaps, the defaults are **Load On Call** and **Moveable**. You can select the defaults for the type of resource you're editing by simply clicking the **Default** button rather selecting each attribute separately.

Resource Attributes Dialog

Name	The name of the resource. The name can be a number, or an alphanumeric name. If you change the name in this field, it effectively works like **Save As**: it saves the resource under a new name, without affecting the original resource.
Load On Call	Determines that Windows loads the resource only when it's called from the application. Choose Load On Call unless your application needs the resource immediately after beginning execution.
Preload	Determines that the resource is loaded into memory when the application is started. Choose preload when you know your application needs the resource immediately after it begins execution.
Moveable	Determines that Windows can move this resource to another place in memory if it needs the memory for something else. Typically, bitmaps are moveable but not discardable. This is because Windows allows bitmaps to be modified within the program, and modified resources can't be discarded.
Fixed	Determines that the resource cannot be moved in memory but remains in a fixed place.
Discardable	Determines whether Windows has to keep the resource in memory, even when it isn't being used. When Windows needs more memory, it discards resources defined as discardable until it has the memory it needs. If a resource isn't defined as discardable, Windows won't discard this resource when it needs memory. Typically, icons, cursors, and strings are defined as discardable because they're read-only and Windows can safely discard them. Discardable resources must also be moveable.

New Header

New Header creates a new header file so you can define new constants in the header file and correlate them with the resources you define in the editor.

New Header opens the Header Editor so you can edit the new constants.

In the Menu, String, and Accelerator editors, it also adds a Symbol column to the editor's table.

Open Header

Open Header opens an existing header file so you can correlate constants from the header file with the resources you define in the editor.

Open Header opens the Header Editor so you can edit the new constants.

In the Menu, String, and Accelerator editors, it also adds a Symbol column to the editor's table.

Edit Menu

- Undo
- Cut
- Copy
- Paste
- Clear and Clear All
- Select All

Edit Menu

Cut Row

Copy Row

Paste Row

Clear Row

Undo

Undo reverses the last edit modification you made. (**Undo** isn't available in String Editor.)

Cut

Cut removes a selected area and stores it in the Clipboard. The cut information can be retrieved with **Paste**.

Use **Cut** to move a portion of a graphic from one place to another, even to another application.

The cut information remains in the Clipboard until it's replaced by another cut or copy operation in the Resource Toolkit or another application.

Cut Row

Cut Row removes the current row (not selected text) and stores it in the Clipboard. The cut information can be retrieved with **Paste Row**.

Use **Cut Row** to move a row from one line to another. The cut information remains in the Clipboard until it's replaced by another cut or copy operation in the Resource Toolkit or another application.

Copy

Copy copies a selected area and stores it in the Clipboard. The copied information can be retrieved with **Paste**. Use **Copy** to copy a portion of a graphic from one place to another, even to another application.

The copied information remains in the Clipboard until replaced by another cut or copy operation in the Resource Toolkit or another application.

Copy Row

Copy Row copies a the current row (not selected text) and stores it in the Clipboard. The copied information can be retrieved with **Paste Row**.

Use **Copy Row** to copy a row from one line to another. The copied information remains in the Clipboard until replaced by another cut or copy operation in the Resource Toolkit or another application.

Paste

Paste copies the contents of the Clipboard (stored by **Cut**, **Copy**, or another software program) into the editor. Contents of the Clipboard aren't affected. Thus, you can paste them again to another area, or to another application.

Bitmap, Cursor, and Icon editors:

Define the area within which you want to paste. If the area you define isn't the same size as the contents of the Clipboard, a dialog opens so you can choose one of two options:

- Shrink/stretch the contents of the Clipboard so it fits within the area you've selected for pasting.

- Clip contents of the Clipboard. **Paste** then copies whatever fits within the bounded area, measuring from the upper left corner of both the Clipboard and the selected area.

String Editor:

The only text that can be cut, copied, and pasted is text in the String Text field.

Paste Row

Paste Row copies the contents of the Clipboard (stored by **Cut Row** or **Copy Row**) into the editor. Contents of the Clipboard aren't affected. Thus, you can paste them again to another row.

Data in the table won't make sense in a table from a different resource. For example, you can't paste information from a menu table into an accelerator table.

Clear

Clear removes a selected area *without* storing it in the Clipboard. Use **Clear** to erase information from the editor.

Clear All clears the entire work area *without* copying information to the Clipboard.

Clear Row

Clear Row removes all text from the current row *without* copying information to the Clipboard.

Select All

Select All is available only in the Dialog and String Editors.

Dialog Editor: **Select All** selects all controls in the dialog box. To selectively de-select, hold down the SHIFT key and click each control you don't want selected.

String Editor: **Select All** selects all text in the active String Text field.

Images Menu

New	Creates a new image.
Open	Opens an existing image.
Save	Saves an image on disk.
Delete	Deletes an image.

Options Menu

The **Options** menu lets you set the line width for the tools. The widths are identical to those offered by the Line Width tool.

The **Options** menu also lets you turn the Toggle tool on and off, and choose the magnification for the editing area.

Combination mode opens a dialog that lets you determine how colors from Clipboard bitmaps will be pasted to bitmaps in the editing area for pasting.

By default, **Combination mode** pastes bitmap colors exactly as they're stored in the Clipboard. To change pixel color values as they're pasted from the Clipboard to the editor, change the combination of values defined in the dialog. For example, if you reverse the settings of all the dialog's radio buttons, the bitmap will be pasted with the inverse colors to those stored for it in the Clipboard.

Options Menu

The **Options** menu lets you set the line width for the tools. The widths are identical to those offered by the Line Width tool.

The **Options** menu also lets you turn the Toggle tool on and off, and choose the magnification for the editing area.

Options Menu

Next Page

Next Page with Strings

Previous Page

Previous Page with Strings

Goto Line

Next Page

Next Page pages down through screens in the string table, without skipping any screens.

Next Page with Strings

Next Page with Strings pages down through screens in the string table, omitting screens that don't have defined string text. For example, if the current screen displays rows **0** through **8** and value **999** is the next value to have defined string text, **Next Page with Strings** pages to row **999**.

Previous Page

Previous Page pages up through screens in the string table, without skipping any screens.

Previous Page with Strings

Previous Page with Strings pages up through screens in the string table, omitting screens that don't have defined string text. For example, if the current screen displays rows **900** through **908** and value **10** is the first preceding value to have defined string text, **Previous Page with Strings** pages to row **10**.

Goto Line

Goto Line goes directly to a specified line. For example, to go directly to line **999**, choose **Goto Line**, then enter **999** in the resulting dialog's entry field.

Tools Menu

The **Tools** menu can be used as an alternative to using the tools on the Tools palette:

Pencil
Constrained Line and Line
Rectangle and Filled Rectangle
Ellipse and Filled Ellipse
Polygon and Filled Polygon
Drag
Select
Pour
Hot Spot (Cursor Editor only)

Palette Menu

The **Palette** menu lets you get, save, or edit color palettes:

Get Colors

Save Colors

Edit Color

Dialog Menu

Attributes
Show Tab Order
Show Item ID
Show Item Group
Grid Items
Palettes On Bottom
Palettes On Top
Swap Palettes
Show Header

Attributes

Attributes opens the Dialog Attributes box, which lets you define attributes for the dialog.

Show Tab Order

Show Tab Order displays the order in which focus moves through the dialog's controls when you press the TAB key in the application. The order is shown by displaying red numbers in the upper-left corner of each control. The numbers start at 1 and progress sequentially.

To change the tab order by altering the tab number for a control, choose **Set Tabstop** from the **Controls** menu.

Show Item ID

Show Item ID displays the ID of each control. The ID is displayed in blue in the lower-left corner of each control.

To change the ID number, use the control's attributes box.

Show Item Group

Show Item Group represents control groups with patterned rectangles. All controls in the same group show as rectangles of the same pattern.

To define logical groups for the controls, choose **Start Group** from the **Controls** menu.

Grid Items

Grid Items lets you set the grid resolution that determines the placement of controls. For example, setting a grid resolution of 2 means that controls can only be located at even coordinate values within the dialog.

Palettes On Bottom

Palettes On Bottom positions the Tools and Alignment palettes below the editing area in the Dialog Editor.

Palettes On Top

Palettes On Top positions the Tools and Alignment palettes above the editing area in the Dialog Editor.

Swap Palettes

Swap Palettes swaps the locations of the Tools and Alignment palettes.

Show Header

Show Header displays the Header Editor -- assuming you opened a header file by choosing **Open Header** from the **File** menu.

If the Header Editor is currently displayed, this option changes to **Hide Header**, which you can choose to hide the Header Editor.

You can also hide the Header Editor by double-clicking its Control-menu box.

Controls Menu

Attributes
Set Tabstop
Start Group
Move Forward
Move Backward
To Bottom of Tab Order
To Top of Tab Order
Group Together

Attributes

Attributes opens the currently selected control's attributes dialog, which lets you define attributes for the control.

Set Tabstop

Set Tabstop turns on the tab stop bit for the currently selected control. This lets the control receive focus when the tab key is pressed in the application.

Set Tabstop is either on or off. A check next to it indicates it's on.

By default, new controls are defined as tab stops, except for statics, group boxes, icons, and custom controls.

To remove tab stop status from a control currently defined as a tab stop, select the control, then choose **Set Tabstop** from the **Controls** menu to turn off the tab stop status.

Start Group

Start Group defines controls as members of a logical group. A logical group defines a set of controls within which input focus can be moved by pressing the arrow keys.

To define controls as a logical group, select the control you want as the first member of a new group, then choose **Start Group** from the **Controls** menu. This turns **Start Group** on for the selected control. Any new controls placed in the dialog after selecting this become part of this group.

For groups that are already defined, you can change group membership by using **Group Together** on the **Controls** menu.

Start Group is either on or off for a control. If **Start Group** is on for a control and you want to turn it off -- perhaps so you can change the control's group -- select the control, open the **Control** menu, and turn off the **Start Group** option. A check by the menu option indicates it's on.

Move Forward

Move Forward moves the selected control forward one tab stop. For example, if you select a control with a tab order value of 6, and then select **Move Forward**, the control moves to position 5 in the tab order. The control previously in position 5 moves to position 6. The rest of the tab order isn't affected.

Move Backward

Move Backward moves the selected control back one tab stop. For example, if you select a control with a tab order value of 3, and then select **Move Backward**, the control moves to position 4 in the tab order. The control previously in position 4 moves to position 3. The rest of the tab order isn't affected.

To Bottom of Tab Order

To Bottom of Tab Order moves the selected control to the last tab stop in the tab order. All controls in the tab order after the selected control move up one position.

To Top of Tab Order

To Top of Tab Order moves the selected control to the first tab stop in the tab order. All controls in the tab order before the selected control move back one position.

Group Together

Group Together creates a new group. To define control groups, select the controls you want to group together, then choose **Group Together** from the **Controls** menu. This associates all the selected controls into a group and moves them to the top of the tab order.

This operation doesn't work if multiple controls from the group you select in the first step have **Start Group** turned on.

Tools Menu

The **Tools** menu can be used as an alternative to using the tools on the Tools Palette.

Pointer
Dialog and Captioned Dialog
Button
Default Button
Owner Button
Check Box
Radio Button
Edit
Multiline Edit
Edit with Vertical
Edit with Horizontal
Edit with Horizontal/Vertical
Left Aligned Text
Right Aligned Text
Center Aligned Text
Group Box
List Box
Combo Box
Standard Scroll Bar
Scroll Bar
Icon
Custom Control

Alignment Menu

The **Alignment** menu can be used as an alternative to using the tools on the Alignment Palette.

Align Left
Align Right
Align Top
Align Bottom
Center on Horizontal
Center on Vertical
Spread Horizontally
Spread Vertically
Make Same Size