THE DIALOG EDITOR

INTRODUCTION

This chapter explains how to use Resorcerer's Dialog Editor, which lets you edit all aspects of Macintosh windows, dialogs, alerts, Control Panel dialogs, and the items in them.

If you are not already familiar with general resource editing, see the "Editing Resources" chapter earlier in this manual.

TOPICS COVERED

- Dialog Editor resource sets
- Opening an existing dialog
- Creating a new dialog
- Duplicating an open dialog
- Closing a dialog
- Changing dialog attributes
- Using the overview window
- Setting window colors
- Changing a dialog's resource IDs

- Creating a new item
- Dialog items
- Dialog item numbers
- Editing dialog items
- Changing item information
- Control Panel editing
- Setting item colors and text styles

- Trying a dialog out
- Starter code for your dialog
- Printing a dialog
- Decompiling a dialog
- Editor preferences
- Dialog Editor menu commands

DIALOG EDITOR RESOURCE SETS

Dialogs are windows that have sets of standard items such as buttons, lists, or checkboxes displayed within them. The set of items displayed is kept in an *item list*, which can be empty.

Note: Unless otherwise indicated, whenever we refer to a *dialog* in this chapter, we mean to include alerts, windows, and Control Panel dialogs as well. The Editor considers windows to be special cases of dialogs that cannot have any items in them.

Rather than construct dialogs and alerts from scratch every time an application presents one on the screen, the application typically keeps the information on how to construct the dialog in a variety of related resources. The program then asks the Mac's Window or Dialog Manager to build the dialog for it using the template data kept in this set of resources.

Resorcerer's Dialog Editor is responsible for editing five major resources and their related minor resources:

DIALOG TYPE	MAJOR	MINOR RESOURCES
Dialog	‘DLOG’	‘DITL’, ‘dctb’, ‘ictb’, ‘DLGX’
Alert	‘ALRT’	‘DITL’, ‘actb’, ‘ictb’, ‘DLGX’
Window	‘WIND’	‘wctb’
Control Panel	‘nrct’	‘DITL’, ‘dctb’, ‘ictb’, ‘DLGX’
Dialog Item List	‘DITL’	‘ictb’, ‘DLGX’

Each set of resources is characterized by the set's major resource type: ‘DLOG’ for dialogs, ‘ALRT’ for alerts, ‘DITL’ for dialog item lists, ‘WIND’ for windows, and ‘nrct’ (Rectangle Lists) for Control Panel dialogs. With the exception of ‘nrct’ and ‘DITL’ resources, these major resources are often known as *template* resources, since they contain template data that the Dialog or Window Manager uses to create the dialog, alert, or window. Associated with each major resource are its minor resources, which contain information on the items within the dialog, their colors, styles, and types.

In addition, the Editor can create and maintain a separate resource of type ‘DLGX’ in which it keeps extra information about each dialog and its items.

This extension resource lets the Editor support certain standard items and behaviors that were not so standard when dialog items were first designed, such as pop-up menus and lists.

Note: The ‘DLGX’ resource is an optional Resorcerer extension that the Mac’s Dialog Manager neither recognizes nor requires to run your dialog. You should consider it a private Resorcerer resource whose presence helps Resorcerer edit your dialogs. If you do not want the Editor to create these ‘DLGX’ resources, the **Preferences…** command lets you turn this feature off.

The Editor edits each resource set in an integrated WYSIWYG (“What You See Is What You Get”) fashion. Each set of resources is collected together as a single entity while you edit them, and then separated back into the various resources when you’ve finished.

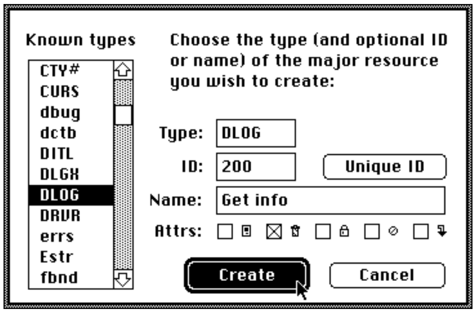
For instance, when you open a ‘DLOG’ resource, a simulation of the dialog is created and displayed on the desktop, using the information in the ‘DLOG’ resource, its related ‘DITL’ and ‘DLGX’ resources, and its optional color and style tables, the ‘dctb’ and ‘ictb’ resources. Or when you open an ‘nrct’ resource to display a Control Panel dialog, the Editor simulates the look of the control panel for you so you can concentrate on the exact placement of the items in both the ‘nrct’ rectangle list as well as the associated ‘DITL’ item list resource.

Since the Editor opens all resources in the resource set simultaneously, related information in the separate resources is kept completely in sync during common editing operations. For instance, the items in a dialog’s ‘DITL’ occur in a particular order, and any style or color information for those items, which is kept in a separate ‘ictb’ resource, must be in the same order. The Editor maintains the consistency between the two resources, even when you cut and paste items between dialogs in different files.

Note: To find out the types, IDs, and other attributes of the current resource set, choose **Resource Info** from the **Resource** menu. This opens a resource information dialog for each resource in the current resource set for editing. For more on this, see the section “Changing a Dialog’s Resource IDs” later in this chapter.

CREATING A NEW DIALOG

You can create a new dialog, alert, window, or Control Panel dialog by using the **New** button in the File Window or the **New** command in the **Resource** menu. Resorcerer then asks the Dialog Editor to build a default empty resource set. For more on how the **New** command works, see the earlier "Editing Resources" chapter.



Choose the type (and optional ID or name) of the major resource you wish to create:

Type: DLOG

ID: 200

Name: Get info

Attrs: □ ☒ □ □ □

In assigning a new resource ID, the Dialog Editor looks in your file for the first free ID number that has not been assigned to any resource of the same types as are in the resource set you are creating. For instance, if you ask that 'DLOG' 405 be created, but 'DITL' 405 already exists because there is an 'ALRT' 405 that references it, the Dialog Editor will search upwards from 405 until it finds an ID that is not used anywhere in the file for every type in the resource set. This guarantees that all 'DITL' resources are referenced by either a 'DLOG' or an 'ALRT', but not both. It also guarantees that the same ID is used for all resources in the set. Once a free ID is found, Resorcerer assigns it to all resources in the new set.

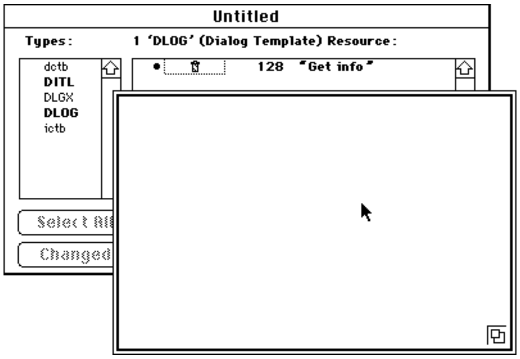
Note: The **Resource ID Preferences** section of the **Preferences** command in the **Edit** menu lets you assign the base ID for any type from which to begin allocating resource ID numbers. Many programmers develop the habit of letting their dialog IDs begin at, say, 200, while assigning IDs of, say, 400 and higher to alerts. This convention helps you tell which 'DITL's are which in the Resources List.

When the major resource is a 'DLOG', the minor 'DITL' resource is created with no items in it; the optional minor 'dctb' resource is created with standard black and white colors in it; and the optional minor 'ictb' resource is created with no entries, to match the empty 'DITL'. The Editor also creates its own 'DLGX' resource in which to hold extra information about the dialog and its items.

When the major resource is an 'ALRT', the same happens except that instead of creating a minor 'dctb' resource, the Editor creates an 'actb' resource with the same default colors.

When the major resource is a ‘WIND’, only a minor ‘wctb’ resource is created, since windows don’t have associated minor ‘DITL’ resources. Like the other color tables, the ‘wctb’ is filled with the default black and white colors.

Finally, when the major resource is an ‘nrct’, the associated ‘DITL’ is created with no items in it. Currently, Control Panel dialogs don’t support color or item styles; however the Dialog Editor still creates a temporary ‘dctb’ resource. The Editor does create its own ‘DLGX’ resource.



Resorcerer temporarily adds the new resources to the current file, and then opens the dialog editing window on the desktop. It looks like a standard modal dialog window for dialogs and alerts; a standard document window with GoAway box for windows; and a simulated Control Panel dialog for Control Panel dialogs.

When you open an item list by itself, the Editor temporarily creates a standard document window in which to display the items from the ‘DITL’ and associated resource. The Editor computes the size of the window by starting with a rectangle the size of your main screen and shrinking it to fit around all the visible items.

If you cancel the new dialog editing session using **Cancel Edit Session** in the **Resource** menu, all resources in the resource set that were added to the file are thrown away.

OPENING AN EXISTING DIALOG

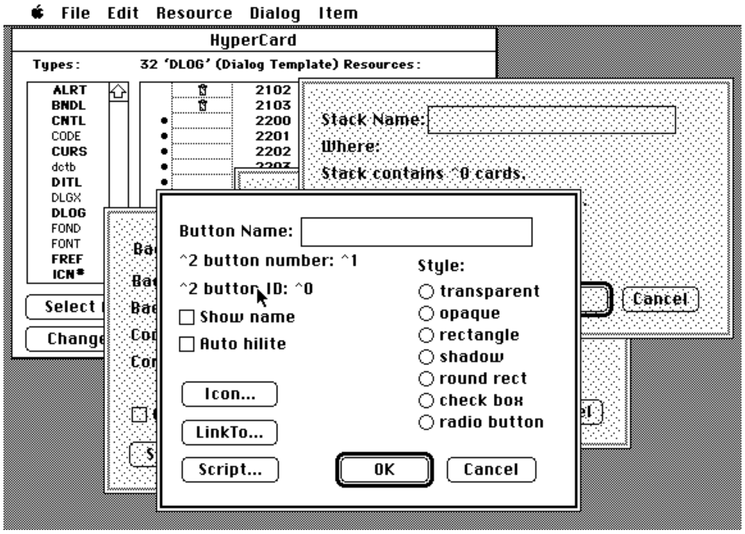
To open an existing dialog, open its major resource (usually a ‘DLOG’ or ‘ALRT’). Resorcerer gathers all existing related minor resources and presents them to the Dialog Editor as the resource set to edit. If any minor resource is missing, the Dialog Editor automatically creates a default-valued one and asks Resorcerer to temporarily install it in the file. You can see its entry in the File Window once the resource set is opened. If you do not edit any part of the opened dialog that would be saved in one of these temporarily created resources, Resorcerer automatically deletes the optional resource when you close the resource set.

Note: If you **Save** the file while the dialog is open for editing, all resources in the set will be saved. You may want to close the dialog first before saving the file.

The optional minor resources are the ‘dctb’, ‘actb’, ‘wctb’, and ‘ictb’ color table resources. The ‘DITL’ resource, although minor, is not considered optional. The ‘DLGX’ resource depends on whether you have asked the Editor to keep extra information in it or not. For more information, see the “Editor Preferences” section later in this chapter.

BROWSING DIALOGS

You can have as many dialogs open as memory and the speed of your system permit, although usually you will only want to edit one or two simultaneously. Because the Window Manager slows down, between 10 and 20 dialogs simultaneously open are the most that you or your machine will be interested in.



The easiest way to open all the dialogs or alerts in a file is to fill the Resources List with ‘ALRT’s or ‘DLOG’s, use the Shift key to extend the selection across all the ones you want to open (or click on the **Select All** button), and then click on the **Open** button.

After opening lots of dialog windows, you can use Resorcerer’s **Send Behind** command in the **Edit** menu to rotate through all your open windows, or **Close Resource Set** in the **Resource** menu to close ones you are not interested in, or **Close All This Type** to close them all.

Sorcery: ⌘ B is the keyboard equivalent for **Send Behind**, but only when no items in the dialog are selected.
⌘ W is the keyboard equivalent for **Close Resource Set**.
⌘ . (period) will halt the opening of further selected resources if you've asked to open too many for the Mac to handle before the system slows down unacceptably.
If no further opening is going on, ⌘ . will **Cancel** the frontmost editing window.

THE DIALOG EDITING WINDOW

For each of the various major resource types, the Editor opens an editing window appropriate to the major resource type and draws the items, if any, in it. This window represents the various resources in the resource set that you have opened for editing. A modal dialog is edited in a modal dialog window, a non-modal dialog in a standard document window, etc., so that you can see exactly how the dialog will look when it is eventually used in your application.

The initial position of the window is copied directly from the major 'DLOG', 'ALRT', or 'WIND' resource.

The title of an editing window is taken from the resource *data*, rather than being formed from the major resource's ID, name, and file, as happens with other resource editing windows. There are no extraneous controls or tools to get in the way of the illusion that you are editing the actual dialog with its own window title on the desktop.

Regardless, however, of the type of editing windows being displayed, the Editor treats all dialog editing windows as non-modal document windows. You can drag them, grow them, and have any number of them open simultaneously, even if visually they appear to be modal dialog or alert windows.

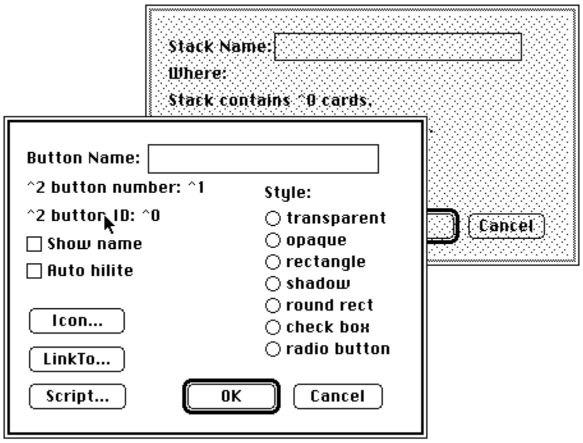
Sorcery: If you want to view a dialog by itself on the desktop, you can choose **Hide File Windows** from Resorcerer's **File** menu to get rid of any distracting file windows.

Note: The Mac normally does not let you switch to another application window if the frontmost window is a modal dialog window. When you are editing a modal dialog, however, it only looks like a modal dialog window; it is not one internally, so you can switch without any problem.

However, if the window your modal dialog is displayed in is based on your own *custom* Window Definition Function (a 'WDEF' using the standard modal variation code of 1), then MultiFinder will *not* let you switch directly to another background application when your dialog is the frontmost window. You can click on the File Window to which the dialog belongs (or choose **File Info** from the **File** menu), though, and then switch out from there after the File Window has become the frontmost window.

DIMMING INACTIVE DIALOGS

The Dialog Editor marks inactive editing windows (those that are behind the front window) by overwriting their content regions with a light gray dusting of bits. This helps you better remember that they are windows that behave differently from how they may look, and is especially important when editing more than one modal dialog at a time, since their window frames on black and white screens are not designed to indicate whether they are active or inactive.



Note: You can disable the dimming feature using the **Preferences...** command in the Dialog Editor's first menu, in case you want to view two dialogs side by side, or you're taking screen snapshots, or if you just don't like it. Dimming inactive dialogs forces lots more update drawing activity which, on a standard Mac Plus system, may take too much time for complicated dialogs.

WHY RESORCERER'S WINDOWS CAN DIFFER FROM THE STANDARD

The Dialog Editor's editing windows must look like standard Macintosh dialog windows, so Resorcerer chooses to represent *its own* dialog and alert windows (including most other Editor's windows) using slightly different drag bars and borders from those in the standard Mac interface. It does this only for screen depths less than 8 bits. This helps you keep track of which windows, dialogs, and alerts are Resorcerer's and which are your editable representations. All of Resorcerer's document windows have a gray patterned drag bar (usually colored light blue on color Mac systems), and all of its alerts and modal dialogs are presented in a modal dialog window with a gray patterned border (colored blue for dialogs or red for alerts on color systems).

STRANGE OR HIDDEN SITUATIONS WHEN OPENING A DIALOG

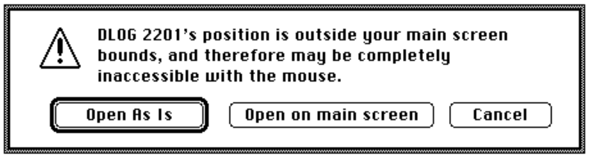
Due to the WYSIWYG nature of the Dialog Editor, a variety of strange situations can occur while opening and constructing a dialog and its items. The Editor checks for these situations and alerts you to the problems and how to overcome them. The situations checked are:

Window bounds are empty

The position of the dialog, window, or alert, as found in the major resource's template data, is empty: that is, the width or height of the rectangle specifying the position is 0 or negative. If you want to continue to open the dialog, the Editor will assign a default position for viewing. This changes any template data.

Window bounds are off the main screen

The position of the dialog, window, or alert, as found in the major resource's template data, is outside of your main screen bounds. This can happen if the data has been purposefully edited to make it so, or if the dialog was saved after being dragged onto a second screen while being edited. The Editor asks you if you want to adjust the position to bring it back onto the main screen, or to open it as is.



DLOG 2201's position is outside your main screen bounds, and therefore may be completely inaccessible with the mouse.

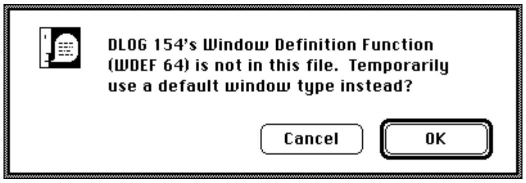
Missing DITL

When a dialog (‘DLOG’) or alert (‘ALRT’) is opened, its item list (‘DITL’) minor resource is expected to be present and from the same file as well. If it is not, the Editor lets you know, and creates an empty list for you.

Typically, however, the item list resource *is* in the same file. The real problem is usually that the major resource’s template data contains an improper resource ID refering to its associated ‘DITL’ resource. Double-click on the window background or choose **Set Dialog Info** (or **Set Alert Info** if you’re editing an alert) to edit the resource ID in the template data directly.

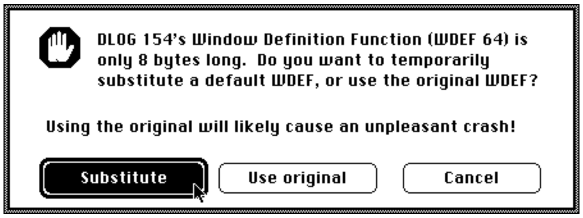
WDEF too small or not found

Some dialogs and windows use custom Window Definition Functions (‘WDEF’ code resources) to display themselves. The resource ID of these functions is kept as part of the dialog or window template data. Before displaying a dialog window via its associated ‘WDEF’, the Editor first ascertains the ‘WDEF’s presence in the same file as the dialog resource, and checks its size as well. If the ‘WDEF’ is not present, and is not the standard system default ‘WDEF’, the Editor substitutes a default window type so that you can continue to edit the contents of the dialog.



DLOG 154’s Window Definition Function (WDEF 64) is not in this file. Temporarily use a default window type instead?

Similarly, if the ‘WDEF’ looks suspiciously small, you can choose to substitute the system default. Unless you really know what you’re doing, you’ll mostly likely want to substitute the default; otherwise, the Window Manager will likely crash. Certain applications use very small ‘WDEF’s that contain application-installed jump vectors back into the application code. This works fine for the particular application in question, but Resorcerer cannot similarly install the same jump address.

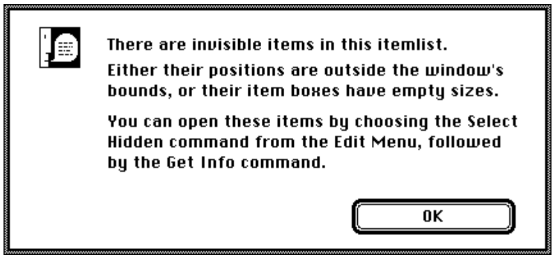


DLOG 154’s Window Definition Function (WDEF 64) is only 8 bytes long. Do you want to temporarily substitute a default WDEF, or use the original WDEF?

Using the original will likely cause an unpleasant crash!

If you always want the opportunity to substitute the standard ‘WDEF’ for displaying a window that depends on a custom ‘WDEF’, you can set the minimum size threshold to any number larger than the size of your custom ‘WDEF’ resource. This may be necessary if, for instance, the custom ‘WDEF’ is misbehaving or otherwise interacting poorly with Resorcerer (which, after all, is not the application the ‘WDEF’ was designed to work with). The minimum size is settable when you use the Editor’s **Preferences...** command.

Hidden items



There are invisible items in this itemlist.

Either their positions are outside the window's bounds, or their item boxes have empty sizes.

You can open these items by choosing the Select Hidden command from the Edit Menu, followed by the Get Info command.

Many dialogs keep various items hidden until some user action causes them to be shown (see, for instance, the “cards” in Resorcerer’s own **Preferences** dialog for a grand example). When you edit a dialog with hidden items, however, they cannot be seen and can easily be forgotten. Thus, the first time any just-opened dialog with hidden items in it becomes the front window, you are alerted to their presence. If the dialogs you are editing always have hidden items in them, you can ask the Editor to skip the alert, or just to beep rather than put up the entire alert. For more on this, see the **Preferences...** command documented later in this chapter.

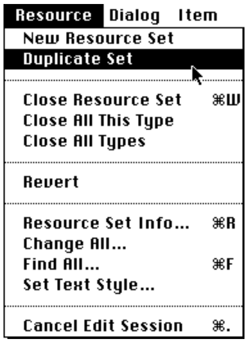
The three kinds of hidden items are: 1) items whose non-empty bounding boxes are just outside the bounds of the dialog window; 2) items whose non-empty bounding boxes have been formally hidden, as in the HideDItem toolbox command, which offsets the X coordinates of the item bounding boxes by 16384 pixels; and 3) items with empty or inverted bounding boxes anywhere inside or outside the dialog window bounds.

You can select hidden items using **Select Hidden** in the **Edit** menu, and once selected, you can open each hidden item’s information dialog to check the actual data. If you know that the items have been formally hidden, the **Show Items** command in the **Items** menu will bring them back into view after they have been selected.

Missing references to CNTL, ICON, PICT, or MENU resources

Dialog items that refer to other resources have the resource IDs of these other resources incorporated into their data. The Editor checks to see that the referenced resources are in fact in the same file as the dialog being opened, and alerts you if it can't find them. You can turn this automatic checking off using the appropriate checkbox in the Editor's **Preferences...** command.

DUPLICATING AN OPEN DIALOG



To duplicate the entire resource set for a dialog that is the frontmost window, choose **Duplicate Set** from the **Resource** menu. The new resources are added to the same file as the originals came from.

The Editor does not duplicate resources that are only referred to by items in the dialog's item list, such as 'PICT's, 'ICON's, 'CNTL's, and 'MENU's; only their ID references in the item list are. For example, a Picture item in the duplicated dialog's item list will have the same 'PICT' resource ID in its data as the original. The 'PICT' itself will remain unduplicated.

All data in the resource set is duplicated with the exception of the dialog, alert, or window positions, which are offset from the original, so that the duplicated dialog doesn't completely cover the original. You must drag the duplicate session window back to the same position as the original if you want both dialogs to have the same positions (see the **Note** below).

In order to create a duplicate item list ('DITL') resource, all item numbers in the dialog you are editing must be valid (for more on this, see the "Item numbers" section later in this chapter). If they are not, the Editor will alert you to the problem, and give you the option of automatically renumbering before continuing the duplication, or allowing you to cancel so that you can fix the item numbers by hand.

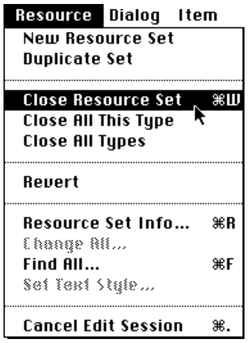
Note: To duplicate a closed dialog, select the major resource ('DLOG', 'ALRT', 'nrct', 'WIND') in the Resources List of your File Window, and choose either the **Edit** menu's **Duplicate Selection** or the **Resource** menu's **Duplicate Set.** When duplicating a closed resource, any window position will remain the same.

For more on duplicating closed resources, see the "Editing Resources" chapter earlier in the manual.

CLOSING YOUR DIALOG

SAVING YOUR CHANGES

Because the Dialog Editor edits dialogs in their own windows, there may not be a GoAway box as part of the editing session's window. If there is, simply click in it as you would other resource editing windows to initiate closing all the resources represented by the window. If there is no GoAway box, you will have to use **Close Resource Set** or one of its related commands in the **Resource** menu.



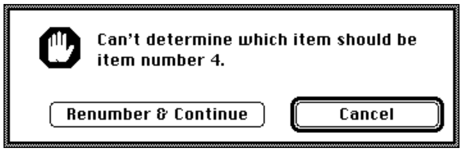
Sorcery: ⌘W is the keyboard equivalent of **Close Resource Set**.

If you have made any changes to any of the resources in the resource set, you will be prompted to confirm them, unless you have unchecked **Confirm resource saves** in the **Confirm Preferences** section of Resorcererer's **Preferences** dialog (for more on this, see the "Preferences" chapter later in the manual).

If the dialog, alert, or window color table ('dctb', 'actb', or 'wctb') resources or the item color table ('ictb') resource were missing when you opened the dialog, they were created for you to fill out the resource set while the dialog was open for editing. Because these are optional, if you have not set any new information that would be recorded into these two minor resources, they will be deleted as you close the resource set.

The Dialog Editor also keeps extra information about your dialog in a separate resource of type 'DLGX'. This resource may also have been created when you opened the dialog, and will be saved at this time (for more on the 'DLGX' resource, see the "Dialog Editor resource sets" section earlier in this chapter).

In order to build the item list ('DITL') resource, all item numbers in the dialog you are editing must be valid (for more on this, see the "Item numbers" section later in this chapter). If they are not, the Editor will alert you to the problem, and give you the option

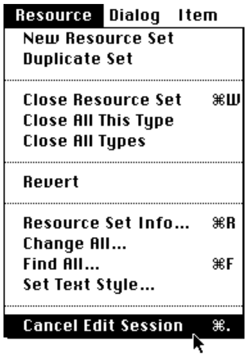

Can't determine which item should be item number 4.

of automatically renumbering before continuing. You can **Cancel** so that you can fix the item numbers by hand before saving. You should choose **Renumber & Continue** *only if you don't care* about the order of items in the 'DITL'. Although the Editor attempts to maintain the existing relative item ordering, there are always going to be situations where it must make arbitrary numbering decisions.

DISCARDING OR REVERTING YOUR CHANGES

If after editing a dialog you decide to throw away any changes you've made since you opened it, choose **Cancel Editing Session** from the **Resource** menu. Although most resource editing session windows contain a **Cancel** button for this purpose, the Dialog Editor cannot due to the WYSIWYG nature of its design.

Sorcery: ⌘ . (period) is the keyboard equivalent of **Cancel Editing Session**.



If you have made any changes, Resorcerer will ask you to confirm that you want to discard them, unless you have unchecked the **Confirm resource cancels and reverts** preference in the **Confirm Preferences** card of Resorcerer's **Preferences** dialog.

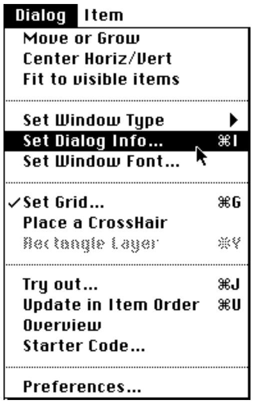
When you choose **Revert**, Resorcerer cancels the session and re-opens the resource set after throwing the changes away.

CHANGING DIALOG ATTRIBUTES

Depending on the type of major resource you are editing, the first menu to the right of Resorcerer's **Resource** menu is named either **Dialog**, **Alert**, **Window**, or **ControlPanel**

In the first three cases, the major resource is a template containing the information about the dialog, alert, or window that an application needs for building the dialog, alert, or window. Control Panels are not represented by template data, and so are not considered here; see "Control Panel Editing" in a later section of this chapter.

If you have opened an item list by itself, the Editor uses the **Dialog** menu, since the item list is displayed in an tempoarary dialog window.



For each of these menus, there is a **Set Dialog Info**, **Set Alert Info**, or a **Set Window Info** command, each of which opens a window that lets you directly edit all fields of the respective template resource data. In addition, it displays the first five standard window colors taken from the associated color table resource. As you change values of the data in this dialog, the actual window, dialog, or alert window is updated appropriately, with the exception that while you're editing window position data the window won't change if you specify a negative size.

Sorcery: Double-clicking in the background area of the editing window is the equivalent of choosing the appropriate **Set Info** command for whichever type of major resource is being edited. For windows, this area is the entire content region of the editing window; for dialogs and alerts, this area is any part of the content region of the editing window that is not covered by a dialog item.

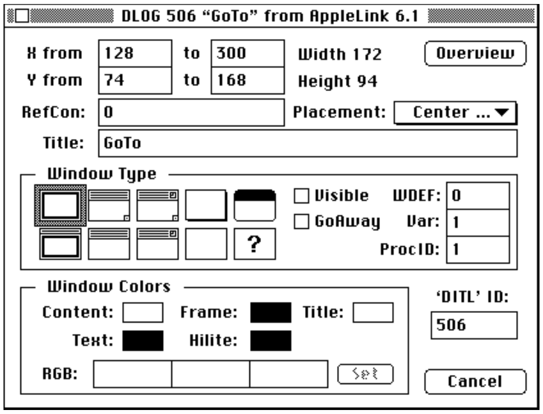
⌘I will also bring up the appropriate **Set Info** dialog, but only if there are no dialog items currently selected (otherwise it opens the items' own information windows).

⌘-double-clicking in the background opens the Overview Window for your dialog. For more on this, see the "Using the Overview Window" section later in this chapter.

DIALOG AND WINDOW TEMPLATE DATA

Both window ('WIND') and dialog ('DLOG') resources contain the same data, with the exception that the dialog data contains one extra field for specifying the dialog or alert's item list ('DITL') resource ID.

To open the template data for a dialog or window, double-click anywhere in the background area of the open editing window, or choose **Set Dialog Info** from the **Dialog** menu. The Editor opens a non-modal dialog window in which all of the various fields in the template are viewable and editable. These fields are more fully documented in the Window and Dialog Manager chapters of *Inside Macintosh*. In addition, you can set the dialog's standard five window colors here (for more on this, see the "Setting Window Colors" section later in this chapter).



DLOG 506 “GoTo” from AppleLink 6.1

X from 128 to 300 Width 172

Y from 74 to 168 Height 94

RefCon: 0

Placement: Center ...

Title: GoTo

Window Type

□ Visible

□ GoAway

WDEF: 0

Var: 1

ProcID: 1

Window Colors

Content: Frame: Title:

Text: Hilite:

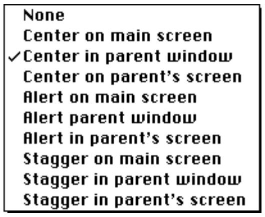
RGB:

‘DITL’ ID: 506

The bounding rectangle’s coordinates are visible, along with the width and height of the window. When you change any coordinate, the change will automatically be reflected in the actual editing window as well, unless one or the other coordinate becomes non-positive. You can also move or grow the window directly (see the following section, “Moving the editing window”). When you do, its coordinates are automatically updated if the template data window is open in the background.

When you need to get a better idea of your dialog’s position with respect to a variety of standard screen sizes, click in the **Overview** button to bring up the Overview window for your dialog. Any changes you make using the Overview window are automatically reflected numerically in the Dialog data window.

Under System 7, dialog, alert, and window resources can have an extra data field that tells the Window Manager to override the window coordinates in the rectangle, and to position the window automatically in one of a variety of ways. The values this field can take on are available in the **Position** popup menu.



To continue to place the window at the absolute coordinates, set the popup to **None**. The **center** variations center the window both horizontally and vertically with respect to the main screen, the screen of the parent window, or the parent window. The **alert** variants place the window with a fifth of the leftover space above the alert and the rest below. The **stagger** variants place the window at a staggered offset from the upper left corner of the main screen, the screen of the parent window, or the parent window.

If **Go Away** is checked, the dialog or window will have a GoAway box if the window type supports it (e.g., document or accessory windows).

If **Visible** is checked, the window will be made visible when your application calls upon the Mac Toolbox to read and build the window using the template resource. While editing the dialog, however, this property is ignored.

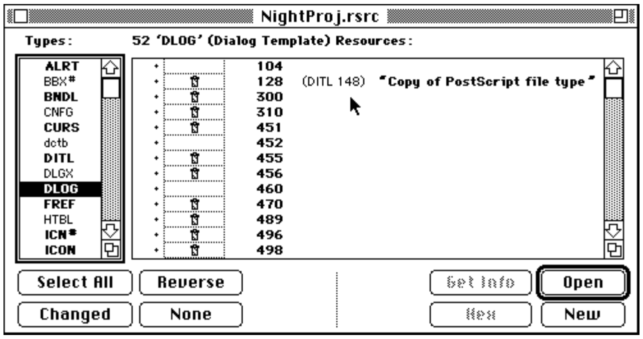
Note: If you are not using the automatic centering fields, or you want to provide a good user interface under System 6, it is best to make your dialog windows invisible so that your application can center the window first. It should then explicitly make the window visible after centering it (otherwise, the window will jump around on the screen when you call `GetNewDialog`).

The **RefCon** field is available to hold any numerical value at all. You can enter either a hexadecimal (preceded by a '$') or decimal constant in this field.

The **DITL Res. ID** field, which is displayed only for dialog (not window) template data, contains the resource ID of the minor item list ('DITL') resource that specifies the items to draw in the dialog. The items displayed are taken from whatever 'DITL' resource having the given ID is found in the same file as the major resource. If none is found, the Editor will let you know that it is missing, and display no items.

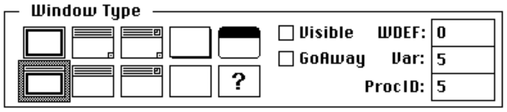
Note: Although you can share the items in a 'DITL' resource among more than one dialog, we highly discourage this practice. Resorcerer requires that all open resources be represented by one and only one resource editing window.

In general, it is a good idea to keep the resource ID of the major 'DLOG' or 'ALRT' resource the same as that of the 'DITL' to which the major resource refers. If the two IDs are not the same, the Resources List displays the 'DITL' resource ID alongside the 'DLOG' or 'ALRT' to let you know that the IDs are out of sync.



104
128 (DITL 148) "Copy of PostScript file type"
300
310
451
452
455
456
460
470
489
496
498

The **Window Type** panel lets you specify the type of window to use for your dialog. The window type icons on the left let you choose from among the standard Macintosh window types; the numerical boxes on the right let you explictly set the resource ID and variation code of the Window Definition Function ('WDEF') resource to use to display the window. These two numbers are combined into a single 16-bit quantity, called the *procID*, which contains the 'WDEF' resource ID in its upper 12 bits, and the variation code in its lower 4 bits. You can type or paste this number directly into the procID field, or enter the two numbers themselves into their respective fields. If the 'WDEF' whose ID you have entered is not available to display the dialog window, then a standard document window is temporarily substituted while you edit the dialog.



Window Type

□ Visible

□ GoAway

WDEF: 0

Var: 5

ProcID: 5

Note: The standard 'WDEF' variation code of 5 is defined for moveable modal windows, which are undefined in Mac systems less than 7.0. If the Editor sees that you are running under an earlier system than 7.0, it simulates the Moveable Modal window; otherwise, it calls upon the system 'WDEF' to display the window.

Note: For standard desk accessory type windows, the roundedness of the corners is determined from the variation code.

Note: You can change a dialog's window type by choosing one of the entries in the **Set Window Type** sub-menu in the **Dialog** menu. This lets you quickly make the change without having to open and then close the template data.

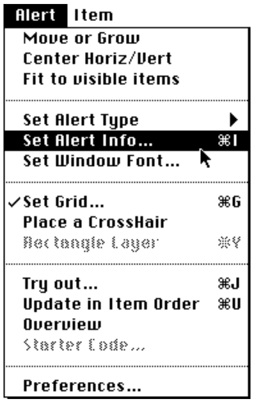
The title of the window or dialog is settable in the **Title** field, which can contain from 0 to 255 characters of title.

Below all these fields, the standard window colors are displayed. For more on these, see "Setting Window Colors" later in this chapter.

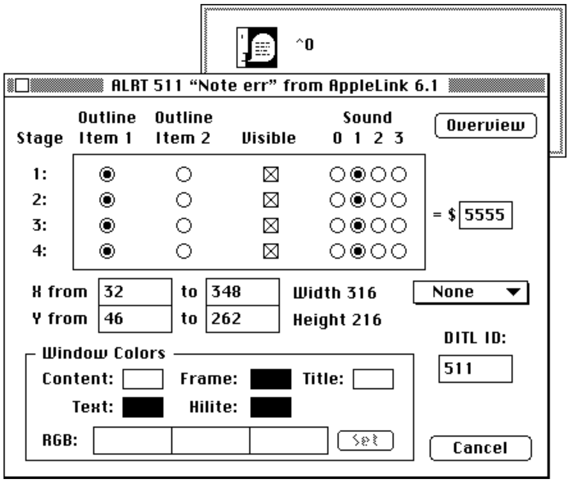
To discard any changes you've made to the data since opening the template data window, click in the **Cancel** button.

Sorcery: ⌘W is the keyboard equivalent of **Close Resource Item**.
⌘. (period) is the keyboard equivalent of **Cancel**.

ALERT TEMPLATE DATA



To open your alert template data for viewing, either double-click on the alert window's background, or choose **Set Alert Info** from the **Alert** menu. The non-modal dialog opened presents three groups of information: the *stages* word and all its bits, the 'DITL' resource ID for the items in the alert, and the alert window's bounding rectangle, along with its width and height.



Stage	Outline Item 1	Outline Item 2	Visible	Sound 0	1	2	3
1:	◉	○	☒	○	◉	○	○
2:	◉	○	☒	○	◉	○	○
3:	◉	○	☒	○	◉	○	○
4:	◉	○	☒	○	◉	○	○

= $ 5555

X from 32 to 348 Width 316
Y from 46 to 262 Height 216

None

DITL ID: 511

Window Colors
Content: □ Frame: ■ Title: □
Text: ■ Hilite: ■
RGB:

The bounding rectangle's coordinates are visible, along with the width and height of the alert window. When you change any coordinate, the change will automatically be reflected in the actual editing window as well, unless one or the other coordinate becomes non-positive. You can also move or grow the window directly (see the following section, "Moving the editing window"). When you do, its coordinates are automatically updated if the template data window is open in the background.

If you need to get a better idea of your alert's position with respect to a variety of standard screen sizes, click in the **Overview** button to bring up the Overview window for your dialog. Any changes you make from the Overview window are automatically reflected numerically in the Alert data window.

The stage information consists of the bits in a 16-bit word, which is presented to you in a panel containing a matrix of radio buttons and checkboxes, as well as in a standard editing box with the hex value so you can cut and paste the information as a single number.

Every alert has four stages, corresponding to the number of times the alert has been called consecutively by its application. At each stage, the Dialog Manager will look at the alert's template data to determine whether to automatically highlight with a bold border the first or second item in the alert's item list as the default button item; whether the alert should be visible or not; and what sound, if any, to use.

The column of radio button pairs for each stage lets you set which item you want to be the default item, indicated by drawing an extra thick outline around its border.

The Dialog Editor looks at the stages word to see which button (item 1 or 2) to display with the default outline surrounding its border. If both items 1 and 2 are the default button at different stages, the two items are both shown with the outlined border. However, the border is drawn in gray, not black, to indicate the mixed state of affairs.

Note: In general, it is not good user interface practice to change the default button at different stages of an alert.

The **Visible** checkbox column lets you set whether to show the alert or not when you raise it. Typically, all boxes are checked.

During the raising of an alert, for each stage the sound number is the argument to the current `ErrorSound` procedure. This procedure causes the standard system error beep to be sounded the number of times specified by this number, from 0 to 3, with a value of 0 meaning silence.

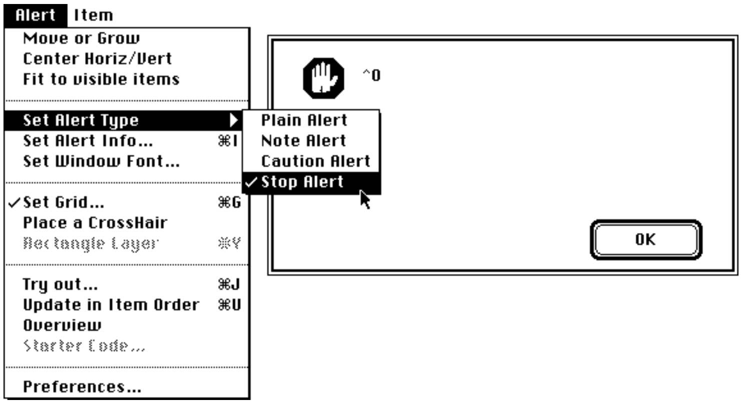
The standard alert window colors are displayed below the template data. For more on these, see 'Setting Window Colors" later in this chapter.

When you have finished setting the alert template data and/or alert colors, click in the GoAway box of the dialog window.

To discard the changes you've made, click the **Cancel** button.

SETTING THE ALERT TYPE

When editing a dialog or window, the **Dialog** or **Window** menu contains a hierarchical sub-menu, entitled **Set Window Type**, that lets you quickly choose among the standard window types without having to open and close the template information dialog. Alerts, however, are always displayed in the same type of window, the standard modal dialog window. The window type is not part of the template data as it is in a general dialog or window template. Therefore, the **Set Window Type** sub-menu is of no use and is dedicated instead to choosing which alert icon, if any, you want displayed in the alert's upper left corner.



Plain alerts do not have an icon in the upper left. Note, Caution, and Stop alerts show the respective system alert icons in the upper left corner.

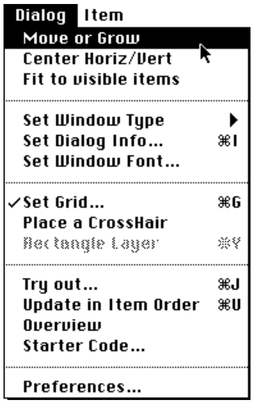
This alert type information is not part of the alert template data, and is for display purposes only, to let you get a better idea of the layout of the alert. The Editor, however, does maintain which type of alert you have chosen, keeping the information in its own 'DLGX' resource so that when you open the alert later it can again display the proper icon that represents its type.

If you open an existing alert whose type is unknown because there is no matching 'DLGX' resource, the Editor sets it to be a Note Alert, unless some item in its item list overlaps the icon area in the upper left corner of the alert window. In this case, the Editor takes it to be a Plain Alert.

MOVING AND GROWING THE EDITING WINDOW

Alert, dialog, and window template resources all contain data specifying where on the desktop their windows should be placed. The Editor uses this data, called the boundsRect, to draw and position the editing window when you open the dialog. When you move or grow an editing window, this changes the resource data to reflect its new position and/or size (see **Sorcery** below, for an exception to this).

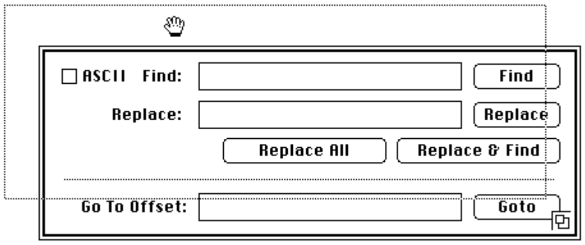
Since dialog windows come in a variety of different types, and because some of these types don't have drag bars or grow icons, the Dialog Editor lets you move or grow any editing window using a special move/grow mechanism which you can invoke by choosing the **Move or Grow** command in the Editor's first menu. This temporarily installs a grow icon in the frontmost dialog's lower right corner, and changes the cursor into a hand.



Note: The grow icon is not normally displayed since it would not normally appear in a modal dialog with items in it, and since it might interfere visually with an item that appears in the window corner (such as an **OK** or **Cancel** button). However, for editing windows with no items in them, the Editor displays the grow icon automatically.

The first time you use **Move or Grow**, the Editor will give you a hint explaining how to use a shortcut for the command (see **Sorcery** below).

Click the hand anywhere within your dialog window outside of the grow icon, and drag the mouse while holding the button. You will drag the window to wherever you want to place it on the desktop. (Of course, if the window already happens to have a drag bar, you can use the usual method directly.) Once you've moved the window to its new position, the cursor returns to its previous state.



You can also move a window by dragging its image in the Overview window. For more on this, see the “Using the Overview Window” section following this one.

When the cursor is within the grow icon in the lower right, it turns into an arrow to indicate that you can click and drag that corner of the window to a new position in order to grow or shrink the session window’s size. After you’ve finished, the grow icon disappears automatically and you can return to editing the dialog’s contents.

Control Panel sessions are displayed in a mock Control Panel dialog window, whose size is taken from the standard system Control Panel dialog, as documented in *Inside Macintosh*. This size is not part of any resource data that you can edit, so that changing the size of the editing window does not change any resource data. Its only purpose might be to give yourself more room in the session window for editing.

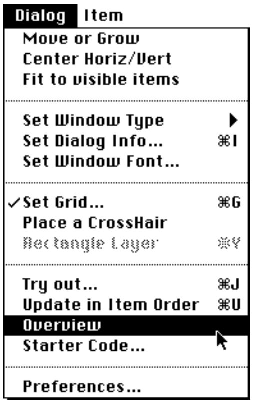
Note: Remember, though, that now that the Macintosh screen can come in many different sizes (and numbers, too), smart programming practice discourages relying on the absolute placement of dialogs and alerts from the resource data. For instance, Resorcerer has over 100 separate dialogs and alerts of its own. Before presenting any of them to you, the program ignores the absolute position in the resource data and calculates where they should appear based both on the size of your Mac screen, the number of screens, and other context. Generally, you shouldn’t worry about the exact placement of a dialog or alert window, only the relative height and width.

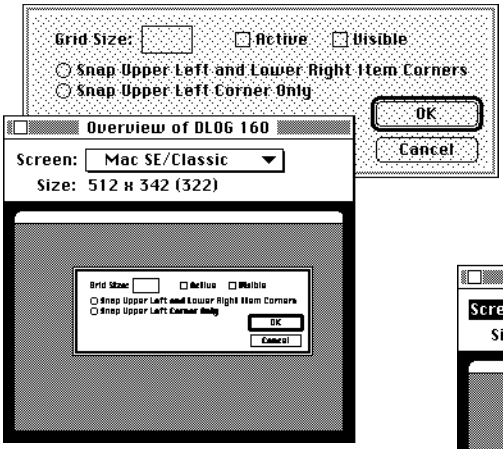
Sorcery: Pressing and holding the Space Bar places you directly into **Move or Grow** mode for as long as you hold the Space Bar down. This lets you perform multiple moves or grows without needing to use the menu command.

If you hold the ⌘ key down while holding the space bar, the template data in the resource is *not* changed. When you enlarge the window without changing the template data, the template data bounds become visible as gray lines on the right and bottom borders of the bounding rectangle.

USING THE OVERVIEW WINDOW

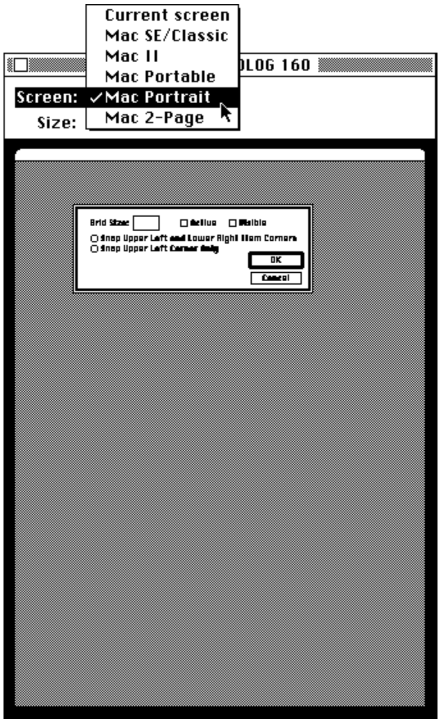
The **Overview** command in the Editor's first menu opens an Overview window, which displays a half size replica of your dialog inside a half-size replica of any of a number of Mac screens. The placement of the half-size dialog is the same within the miniature screen as the actual dialog's position is on your main screen.




Grid Size:
□ Active
□ Visible
○ Snap Upper Left and Lower Right Item Corners
○ Snap Upper Left Corner Only
OK
Cancel
Overview of DLOG 160

Screen: Mac SE/Classic
Size: 512 x 342 (322)

At the top of the Overview window is a popup menu that displays the names of a variety of common Mac screen sizes. If set to **Current screen**, the overview will automatically adjust size when you drag it to another screen. Below the popup is the actual size of the screen being replicated. The *usable* height of the screen (that is, the height of the screen minus the height of the menu bar) is displayed in parentheses. Each time you choose a new screen from the popup, the window resizes itself to display the half size replica of that screen, treated as if it were your main screen. This lets you check to make sure that your dialog window is not too large to fit on some of the smaller Mac screens.



You can change the coordinates of your dialog's position by clicking on and dragging the half-size replica within the Overview screen. If you want to see the coordinates numerically, choose **Set Dialog Info...** first to open the data window.

Sorcery: To bring your dialog window to the front, double-click on its replica in the Overview window.

To center your dialog with respect to your currently chosen replica screen, choose **Center Horiz/Vert** from the Editor's first menu. If your dialog is not already centered horizontally, it is moved to be so. Otherwise, the command either centers the dialog using Apple's 1/3 - 2/3 vertical spacing rule or centers it both vertically and horizontally, depending on its current position. The 1/3 - 2/3 rules says that after subtracting the height of the window from the height of the screen, the window should be placed in such a way that 1/3 of the leftover vertical space is above the window, and 2/3 of the leftover vertical space is below.

If you would like to add more entries to the Overview window's list of possible screens, you can by editing some of Resorcerer's own resources. For more on this, see the "Configuring Resorcerer" chapter at the end of the manual.

Note: The Overview window simulates in miniature the basic look of the standard Mac window types. It does not display any title bar information. If you are using a custom 'WDEF', it will only show you the framed bounding boxes of the custom window's content and structure regions.

SETTING WINDOW COLORS

When you open a dialog resource set, the Editor automatically looks for and loads any accompanying dialog, alert, or window color table ('dctb', 'actb', or 'wctb') minor resource that has the same resource ID as the major resource. If it can't find the accompanying color table, then the Editor creates a default-valued one and *temporarily* adds it to your file.

Each of these color table resources has the same internal format, which is that of a standard color lookup table ('clut') resource. When you use the Dialog Editor (as opposed to the more general Color Lookup Table Editor) to edit these resources, it lets you view and change only the first five entries in the table, since these are the only entries that standard Mac windows use. In nearly all applications, this is quite sufficient; however, if you are using a custom Window Definition Function ('WDEF') that requires more than the first five entries in the color table, you will have to edit the color table resource by itself if you want to access any colors past the first five.

To view or change the standard window colors associated with any dialog, choose **Set Dialog Info...**, **Set Alert Info...**, or **Set Window Info...** from the Editor's first menu, depending on whether you are editing a dialog, alert, or window. Or simply double-click in the window's background. The Editor opens the template information window that also displays the five standard colors, labeled according to which parts of the window the colors affect.

These parts are:

- The window's background or content area,
- The window's frame,
- The window's title bar background color,
- The foreground text color, and
- The highlight color when the window is active.

Each of these colors is displayed in a box, any one of which you can select when you click on it. The selected color box has an extra border surrounding it. The components of the selected color are shown in their own individual edit boxes below, and a **Set** button lets you invoke the Standard Color Picker to change the selected color.

You can **Copy** and **Paste** between the selected color box and any other color box in other Resorcerer editors. This makes color coordination much easier, for instance, between window title bars and menus.

Note: You cannot view the colors on a non-color Mac, although you can still edit the component values.

Sorcery: Double-clicking directly on a color box is equivalent to clicking the **Set** button.

To keep the Editor from disposing of any automatically created default color table resource, you must make an editing change to some color in it. If you don't want a change to any color, but still need to save the color table, click once within any of the three component edit boxes (this causes the Editor to mark the resource as changed).

CHANGING A DIALOG'S RESOURCE IDS

The information about dialogs and alerts is kept in multiple resources whose resource IDs need to conform to certain rules. If the resource IDs of the resource set get out of synchronization with each other, you can get unexpected results.

The best rule of thumb is to make sure that all minor resources have the same resource ID as the dialog's major resource, and that this ID be unique among the set of all dialog ('DLOG') and alert ('ALRT') resources. The Dialog Editor always enforces this rule when creating new dialogs or alerts. The link between the dialog or alert template data and its item list ('DITL') resource is not by convention the same as the major resource ID, but is explicitly encoded into the template data (see the previous section, "Changing dialog data"). The Dialog Manager also looks for a dialog or alert color table ('dctb' or 'actb') minor resource with the same ID as the dialog or alert; and it looks for the item color table ('ictb') with the same resource ID as the item list ('DITL') resource.

If you have set Resorcerer's **Include related resources in selections** preference (for more on this, see the "Preferences" chapter), Resorcerer will renumber all of a dialog's minor resources when you assign a new ID to the major resource. To do this, select the major resource ('DLOG', 'ALRT', 'WIND', etc.) in the Resources List of its File Window, and click on the **Get Info** button. For more on this, see the "Editing Resource Information" section of the "Editing Resources" chapter.

If you want to set the IDs of the individual resources by hand, first turn the **Include related resources in selections** preference off (otherwise the automatic renumbering features will get in your way). If the dialog is not open for editing, you can Option-click on the **Get Info** button to open information windows for all existing resources in the resource set. If you are editing the dialog, the **Resource Info** command in the **Resource** menu changes its name to **Resource Set Info** to indicate that choosing it will open resource information windows for all resources in the set.

When all of the resources in the set have had their IDs changed, you should open the dialog template information for editing, and replace the 'DITL' resource ID field value with the resource ID you've used for it elsewhere. Close your dialog and you should be all done.

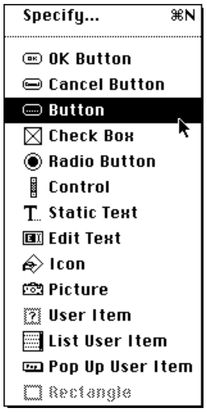
DIALOG ITEMS

The information that the Dialog Manager uses to create the items in a dialog is kept in the 'DITL' (Dialog Item List) resource, with optional color and style information for control and text items kept in an 'ictb' resource sharing the same ID as the 'DITL'.

In addition, Resorcerer's Dialog Editor keeps extra information about the dialog and its items in its own 'DLGX' resource, which the Mac's Dialog Manager knows nothing about. This resource helps the Dialog Editor extend the definition of a standard dialog item to include Pop-up Menu and List items. Other helpful editing information is kept in the 'DLGX' resource also.

All items in an item list resource contain data that encodes the item type, its position in the dialog, and whether or not it is an enabled item. In addition, there is data that is specific to each item type.

BUTTON ITEMS



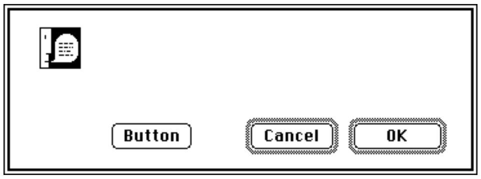
The Editor supports three types of Button items:

OK buttons,
Cancel buttons,
and all other buttons.

The Mac's Dialog Manager makes no distinctions among buttons; they are all stored in exactly the same manner in the item list resource data. However, the two most common buttons are the ones whose titles are "OK" and "Cancel", and by distinguishing these from all others the Editor can more accurately draw and, more importantly, generate code that runs the dialog for you.

OK buttons are drawn as if they were always the default item in the dialog; that is, with a bold border around them. You can give an **OK** button a different title than "OK" if you want by opening its item information dialog (see the later section, "Changing item information").

When you edit an alert, either item 1 or item 2 is considered the default item, and the item should be a button. The alert template data's *stages* word encodes which item is the default. If it says that both items are



the default during different stages, then the Editor displays both with the default bold outline, only in gray instead of black to let you know about the mixed state of the stages.

Note: **OK** and **Cancel** buttons should generally be the first and second items in the dialog's item list, respectively, especially in alert item lists. That is, their item numbers should be 1 and 2. The Editor will notify you if you create an **OK** or **Cancel** button with a higher item number.

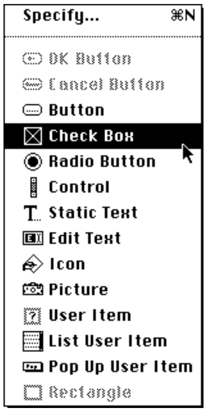
You can set the colors in which the buttons are displayed when you choose **Colors & Text Styles...** in the **Item** menu. For more on this, see the "Setting item colors and text styles" section later in this chapter.

To change the title of a button, double-click on it (or select it and choose **Get Item Info** from the **Item** menu) to open its item information.

You can conform the sizes of any selection of buttons to that of the widest button in the group with the **Use Best Size** command in the **Item** menu.

Note: The Dialog Manager always uses the system font to draw the title text of buttons. If you want a button to use the current window font, you must make the item a Control item, which references a Control Template ('CNTL') resource which in turn uses the standard system 'CDEF', but with the *useWFont* bit incorporated into its procID. For more on this, which also applies to Radio Button and Checkbox items, see "The Control Editor" chapter.

CHECKBOX ITEMS



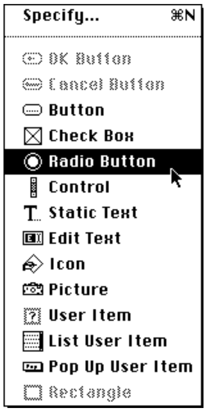
The Checkbox item is a standard Dialog Manager item type that represents a Boolean (on/off) value.

You can set the display colors of the Checkbox item using **Colors & Text Styles…** from the **Item** menu.

To set the title of a checkbox, double-click on it (or select it and choose **Get Item Info** from the **Item** menu) to open its item information.

As with buttons, if you want the Dialog Manager to draw the title text of your checkboxes in any font other than the system font, you must use Control items instead. See the **Note** above for Button items, and the section below on Control items for more on how to do this.

RADIO BUTTON ITEMS



The Radio Button item is a standard Dialog Manager item type. Radio buttons generally are created in groups of two or more; the Dialog Editor maintains a group number for every radio button. This number helps the Editor to create starter code for you and is kept in the associated ‘DLGX’ resource. The group number is displayed along with the item number (see the next section, “Dialog Item Numbers”).

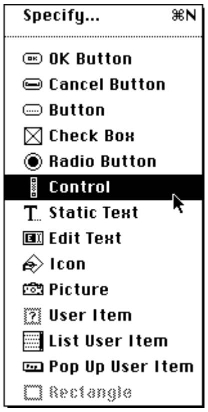
You can set the colors of Radio Button items using **Colors & Text Styles…** in the **Item** menu, described later in this chapter.

To change the title of a radio button, double-click on it (or select it and choose **Get Item Info** from the **Item** menu) to open its item information.

As with Button items, if you want the Dialog Manager to draw the title text of your radio buttons in any font other than the system font, you must use Control items instead. See the **Note** above for Button items, and the section below on Control items, for more on how to do this.

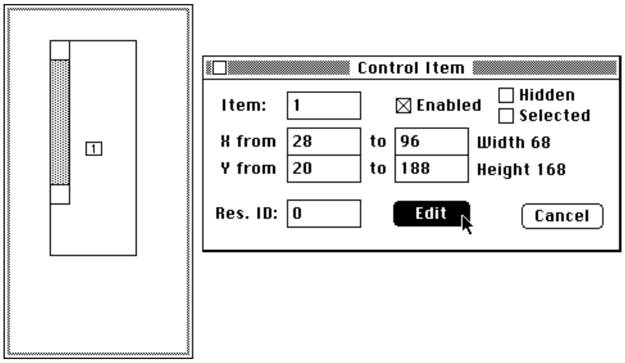
CONTROL ITEMS

A Control item is a more general dialog item that is actually a reference to another resource called a *control template* resource (see “The Control Editor” chapter for more on these). The Dialog Manager looks up the referenced resource (of type ‘CNTL’) and uses the data in it to build a Control item in the dialog. The data can specify standard controls such as buttons, radio buttons, checkboxes, or scroll bars; or it can specify a custom control, which you would need to implement as a custom Control Definition Function (‘CDEF’) resource.



Note: You must use Control items if you want buttons, checkboxes, or radio buttons to be displayed in anything other than the system font. To take advantage of the current window font, the control template must also have its *useWFont* bit set in the ‘CDEF’ `procID` field.

When the Mac’s Dialog Manager builds a control via the control template, it places the control in the dialog using the upper left corner of the Control item’s position. The size of the control, however, is not taken from the item list data, but from the ‘CNTL’ data. For this reason, the Editor displays Control items for you in a way that shows you what both bounding boxes are. The Dialog Editor also looks up the **Visible** field from the template; if it’s not set, the Editor will display the string “CNTL not visible” in the bounding box of the item.



Item: 1 ☒ Enabled □ Hidden □ Selected
X from 28 to 96 Width 68
Y from 20 to 188 Height 168
Res. ID: 0

To edit the referenced ‘CNTL’ resource, open the Control item’s information window, and click on the **Edit** button (for more on this, see the later section, “Editing item information”). If the referenced ‘CNTL’ doesn’t yet exist, this will create a new one with a bounding box the same as the Control item that references it.

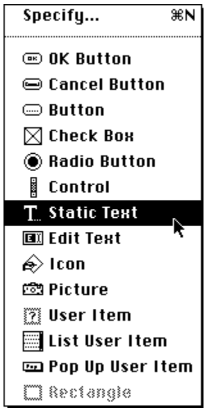
Sorcery: Option-double-clicking on the Control item will open or create the referenced ‘CNTL’ resource directly.

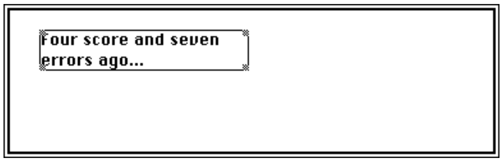
You can use the **Item** menu’s **Use Best Size** command to conform the item’s definition box to the control template’s bounding box, or you can ask the editor to link the two bounds automatically (for more on this, see the “Dialog Editor Preferences” section later in this chapter).

Note: The Dialog Editor does not resolve, build, and display any custom controls while you are editing items, unless the controls are for the standard system buttons or scroll bars. To see any referenced custom controls, you can use the **Try Out...** command, which actually runs your dialog through the Mac’s Dialog Manager. This also lets you interact with any controls normally. For more on this, see the section, “Trying a dialog out” later in this chapter.

STATIC TEXT ITEMS

Static Text items place text in a dialog, usually as a prompt, label, or other identifying string. The text is shown in the standard system font by default; however, you can change the font style either explicitly when you use **Colors & Text Styles...** in the **Item** menu, or implicitly when you choose **Set Window Font...** from the Editor's first menu. Explicitly setting the text style means that the style information is kept in an entry in the item color table ('ictb') minor resource, which the Mac's Dialog Manager recognizes.




Four score and seven errors ago...

Implicitly setting the text style means that your application will change the dialog window's text style prior to letting the Dialog Manager draw the text using whatever the current font is set to. This method is not recommended now that explicitly setting the styles is supported.

Note: The Dialog Manager often does not deal correctly with dialogs whose window font has been set to something other than the system font. In particular, Edit Text items in your item list have been known to cause the Dialog Manager to change the window font back to the system font. Thus, if you want to ensure that all Static Text items in your dialog will appear in some non-standard font, and for some reason you do not want to use the 'ictb' resource, then your Static Text items should appear in the item list earlier than the Edit Text items.

You will save yourself a lot of trouble if you just use the **Colors & Text Styles...** command to put the style information in the 'ictb' resource data, rather than changing the dialog window font in your application. For more on this, see the later section on the **Colors & Text Styles...** command .

The Editor keeps the window font settings you make in its own ‘DLGX’ resource, which the Dialog Manager knows nothing about. Style settings you make with **Colors & Text Styles...**, however, are stored in the standard ‘ictb’ resource which the Dialog Manager will use if it is present.

The **Use Best Size** command will change the height of all selected Static Text items to the integral number of lines required to display all the text in each item, given each item’s current width and whatever word breaks are found in the text content.

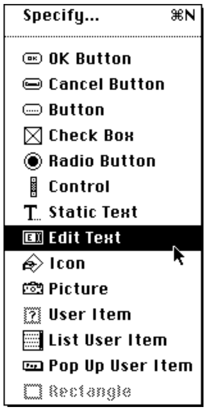
Note: The text in any one Static Text item should not exceed 240 characters.

EDIT TEXT ITEMS

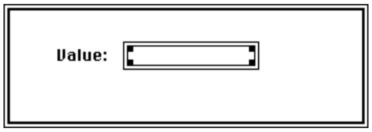
Edit Text items display editable text for the user to interact with. Normally, the application fills in the field with the text, if any, to be displayed; you can, however, specify some initial text in the item data.

The framed bounds of an Edit Text item, as displayed by the Dialog Manager, are actually 3 pixels outside of the item definition box kept in the ‘DITL’ data.

As with Static Text items, the text in a given Edit Text item is subject to the text style information that you can set with **Colors & Text Styles...** in the **Item** menu. If you haven’t already, please read the **Note**s in the previous “Static Text Items” section.

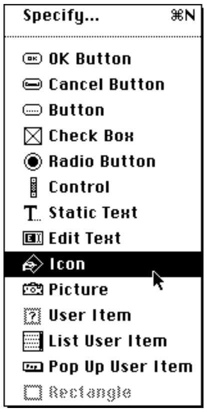


The **Use Best Size** command will change the height of all selected Edit Text items to the integral number of lines required to display all the text in each item, given each item’s current width and whatever word breaks are found in the text content.



Note: The initial text (stored in the ‘DITL’) in any one Edit Text item should not exceed 240 characters.

ICON ITEMS

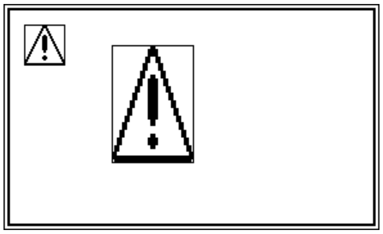


An Icon item is a standard Dialog Manager item that displays an icon within its bounds. The item data contains the resource ID of a ‘cicn’ or an ‘ICON’ resource elsewhere in your file (a ‘cicn’ resource will override any ‘ICON’ resource with the same ID). If the referenced icon can’t be found, the Editor shows the Icon item in gray. An exception is made for missing ‘ICON’ resources with IDs 0, 1, or 2, which are the standard alert icons always found in the System file.

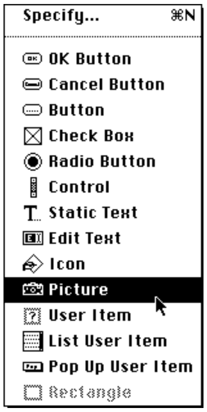
To edit the referenced resource, double-click on the Icon item to open its information window, and click on the **Edit** button there. If the resource doesn’t exist, a new one will be created for you.

Sorcery: Option-double-clicking on the Icon Item will open or create the referenced resource directly.

If the Icon item’s definition box is not the same size as the icon (32 by 32 for ‘ICON’s, and any size for ‘cicn’s), then the icon image will be stretched to fit when it is displayed. The **Item** menu’s **Use Best Size** conforms the item bounding box to the icon size.



PICTURE ITEMS



The Picture item is a standard Dialog Manager item that displays a given picture resource. The item data contains the resource ID of the ‘PICT’ resource elsewhere in the file.

Picture items are considered transparent. That is, the Editor leaves it up to the picture that is drawn whether or not to erase the background. Pictures that are screen snapshots or other kinds of bitmaps will generally completely fill (and thus erase) the background. Pictures that contain simple sequences of QuickDraw commands may leave areas underneath them in your dialog untouched.

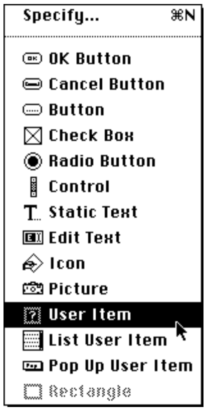
Like Icon items, if the size of the Picture item’s bounding box differs from the picture’s frame size, the picture will be stretched or shrunk appropriately. To keep the picture image from being stretched, you can use the **Use Best Size** command to conform the item bounding box to the picture frame.

To edit the referenced ‘PICT’ resource, double-click on the Picture item to open its information window, and click on the **Edit** button there. If the ‘PICT’ resource doesn’t exist, a new one will be created by letting you take a screen snapshot the same size as the Picture item’s bounding box.

Sorcery: Option-double-clicking on the Picture Item will open or allow you to screen copy the referenced resource directly.

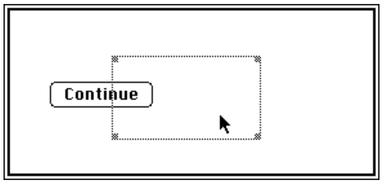
USER ITEMS

The Dialog Manager supports a generic item called a User item. Each User item consists only of an enabled or disabled bounding box. How User items look and how they interact with the user are up to the application, which must draw their content, entertain any mouse clicks, etc..



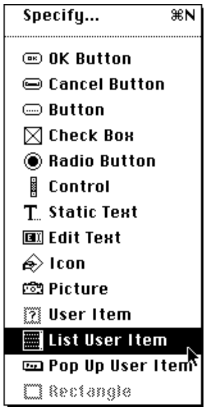
The Editor displays each User item by framing (in gray) its bounding box without erasing its interior. When your application is running the dialog, the appearance of User items in your dialog is entirely up to the application. However, when you choose **Try Out...** to run your dialog through the Dialog Manager, the Editor will frame the user items (in black) to show you their position.

You can create a User item for purely graphic purposes, such as to place a divider line somewhere in the dialog, or to designate a rectangular panel, or cluster, to be framed. The latter use is quite common, and for this reason the Editor presents User items to you as transparent, so you can see any items that they cover.



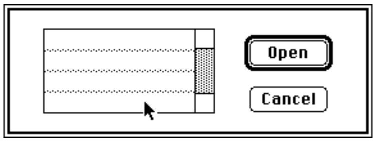
Note: The internal format of a User item within a 'DITL' resource has been ambiguously documented in Apple's *Inside Macintosh* for many years. Because of the way the Dialog Manager works, it is possible to associate private data with each User item directly in the 'DITL' in a way that makes it invisible to the Dialog Manager. Some programs, such as AppMaker™, have taken advantage of this. The Dialog Editor does not support directly editing any extra User item data; however, if it finds any private data in a User item when it scans the dialog's item list, it will maintain the data and write it back out intact. Resorcerer recognizes AppMaker-generated List and Popup user items.

LIST USER ITEMS



Unfortunately, the Dialog Manager does not know about lists the way it knows about buttons, icons, controls, etc. Thus, to implement a list in your dialog, the best you can do is to mark where in the dialog you want it to be shown. Generally, you do this by creating a User item whose bounds your dialog code looks up and provides to the Mac's List Manager.

Lists are important pieces of the Mac interface, though, so the Dialog Editor special-cases this type of User item for you. This lets the Editor display the basic look of the list. More importantly, since it knows which User items are supposed to be lists, the Editor can generate proper List Manager starter source code for you.



Only 1-dimensional lists are supported. The list's cell size is displayed by delineating cells with gray lines, two pixels thick, so that cell boundaries fall in the *middle* of the gray lines.

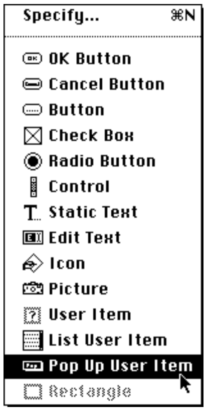
To make the size of the List's item definition box a multiple of the cell size, you can choose **Use Best Size** from the Item menu. The upper half of the bottom-most divider line is normally suppressed when the List is properly sized to an integral number of cells.

List User Item information such as cell size is kept by the Editor in its own 'DLGX' resource. There is no provision for recording this type of item in the standard 'DITL' resource, other than as a generic User item.

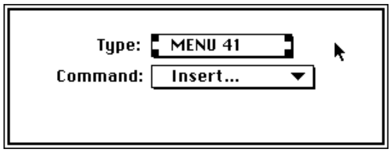
Note: The item definition box of the List User item is 1 pixel inside the rectangular frame that surrounds the list; it corresponds to the content area of the list, including the scroll bar. For best results, the width or height of the item definition box should be a multiple of the cell width or height, respectively.

POPUP MENU USER ITEMS

Another important use for User items is to designate the position of a pop-up menu in your dialog. Like lists, pop-up menus have become such important pieces of the Mac interface that the Dialog Editor special-cases this kind of User item for you. This lets the Editor display the basic look of a standard pop-up menu. More importantly, since it knows which User items are supposed to be pop-up menus, the Editor can generate lots of useful pop-up menu management code for you.



The Editor displays Pop-up items in one of two ways, depending on whether the ‘MENU’ resource referenced in the item’s information can be found in the same file as your dialog. If it can be found and the ‘MENU’ appears to be a standard text menu, the Editor draws the text of the first menu command; if not, it just draws the ‘MENU’ resource ID.



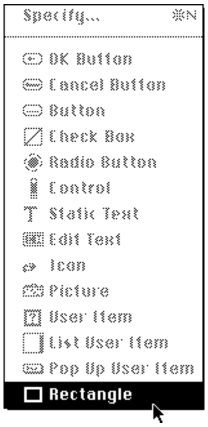
Type: MENU 41
Command: Insert...

The item definition box of the Pop-up Menu User item is one pixel inside the rectangular frame that surrounds the pop-up menu. The **Use Best Size** command in the **Item** menu looks up the menu referred to by each selected Pop-up Menu item, calculates the menu size if it appears to be a standard menu, and sets the Pop-up User item’s width to the width of the menu.

The ‘MENU’ resource ID that the Pop-up User Item keeps as its item information is stored by the Editor in its own ‘DLGX’ resource, since there is no provision for recording this type of item in the standard ‘DITL’ resource.

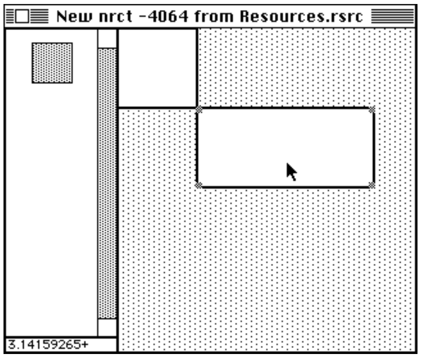
To test the pop-up menu, use the **Try out...** command.

CONTROL PANEL RECTANGLE ITEMS



Rectangle items are not standard dialog items; they are the rectangles stored in the ‘nrct’ major resource that you use to refer to and to open a Control Panel dialog. The Dialog Editor treats Rectangle items much like other standard dialog items while you are editing, with the exception that they are edited in their own *layer*. This is because in a Control Panel dialog that conforms to Apple’s guidelines, the rectangles should completely cover all the usual items in the dialog. By placing the Rectangle items in a different layer, the Editor lets you edit them without any interference from the dialog’s other items. Similarly, it means you can edit the standard dialog items without interference from the Rectangles.

To switch layers, use the **Rectangle Layer / Dialog Item Layer** command in the **ControlPanel** menu.



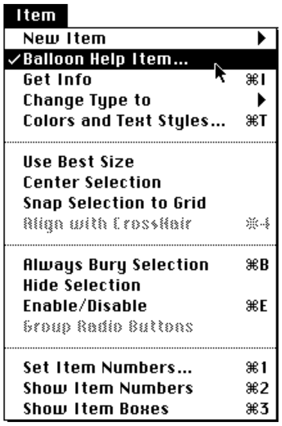
Rectangle items are drawn using a two-pixel thick frame. In order to make adjacent rectangles in the Control Panel look good, the item bounding boxes need to overlap by one pixel rather than just touch each other. The Editor will automatically align the left and top edges of any Rectangle items that touch the left or top edge of the light gray area where they appear in a Control Panel dialog.

Sorcery: You can use the Shift-Arrow keys to bump a Rectangle item’s position by one pixel to precisely align the items.

Rectangle items kept in the ‘nrct’ resource have no provision for the enabled or disabled property; the Editor therefore always displays them as disabled, since they are purely used for graphic purposes.

For more on Control Panel dialogs, see the “Control Panel Editing” section later in this chapter.

BALLOON HELP ITEMS

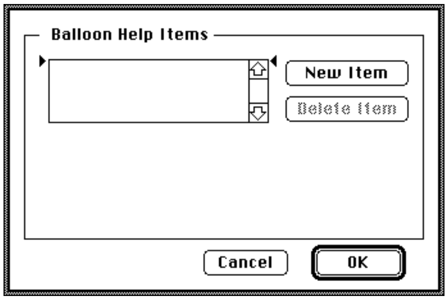


Starting with System 7.0, a dialog or alert can have associated on-line help that the System 7.0 Help Manager displays automatically when the user asks for help. This information is kept in a resource whose type and ID are recorded directly in a special item, called a Balloon Help item, which is usually appended to the list of standard items in your dialog. The Help Item is invisible as far as the dialog and the user is concerned, and its presence in the ‘DITL’ is completely ignored by the Dialog Manager. Its sole purpose is to link another resource (usually of type ‘hdlg’ or ‘hrct’) to your dialog explicitly so the Help Manager can find it when the dialog is running.

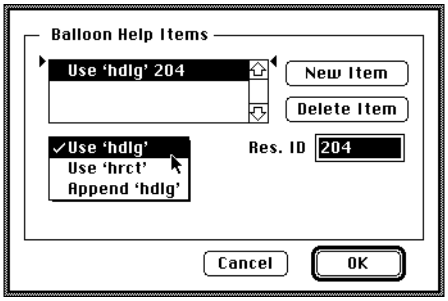
Note: When the Editor initially scans an item list (‘DITL’) resource, it removes all Help Items it finds and keeps them in a separate list. Help Items have no graphical representation, and their presence even in the middle in a ‘DITL’ does not affect how the Dialog Manager reports item numbers. When a changed ‘DITL’ is written back out, the list of Help Items is appended or inserted back into the item list.

Since Balloon Help Items are invisible, you can tell if a dialog’s item list has at least one by seeing if the **Balloon Help...** command in the **Item** menu is checked or not. Choosing this command lets you create, delete, and edit any set of Help Items (there can occasionally be more than one, and the data structure does not limit how many there can be).

To add a Help Item to your dialog, choose **Balloon Help...** and click on the **New Item** button in the Balloon Help Items editing dialog. The new item is inserted at the position of the list insertion caret, and is selected for editing.



Help Items have three variations. In all cases, the item specifies a resource ID of some help resource, either an ‘hdlg’ or an ‘hrct’. Using the popup menu below the list, you can set which of the three types of Help Item you want the selected item to be.



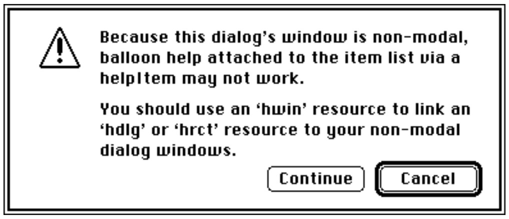
Res. ID 204

In the third “Append ‘hdlg’” case, an additional piece of information is needed, the item number offset. This is useful for stand-alone item lists that are meant to be appended to other item lists at run-time. The most common time this happens is in the Standard Print Job dialog.

To delete a Help Item, select it in the list, and click on the **Delete** button.

When you are done, click on the **OK** button to install your changes, or click on the **Cancel** button to discard them.

Note: In both System 7.0 and 7.1,and possibly later, the Help Manager has problems displaying balloons that are attached to non-modal dialogs using help items. Consequently, the Editor warns you about this bug if the dialog’s window type is not a standard modal window.



Because this dialog’s window is non-modal, balloon help attached to the item list via a helpItem may not work.

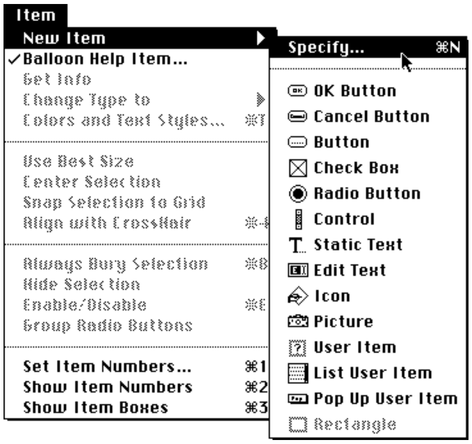
You should use an ‘hwin’ resource to link an ‘hdlg’ or ‘hrct’ resource to your non-modal dialog windows.

Note: Resorcerer’s Data Editor can create and edit all of the standard Balloon Help resources. The templates describing these resources should be in the “Balloon Help Templates” file in Resorcerer’s templates folder.

For more information on how the Balloon Help Manager uses Help Items, see the Help Manager chapter of *Inside Macintosh, vol. VI.*

CREATING A NEW ITEM

The first entry in the **Item** menu, **New Item**, is a hierarchical sub-menu that lets you create a new item in one of two ways. You can choose the **Specify…** command, in which case the Editor brings up a dialog that lets you set the item type and most of the information pertaining to the item; or you can choose from among the item types listed in the rest of the sub-menu, letting the Editor create a default-valued item immediately. In the latter case, you will usually need to open the item's information later on, in order to fill in some of its values.

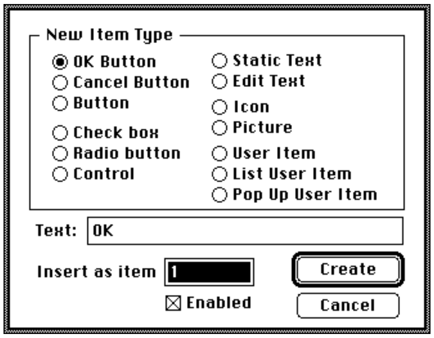


In either case, the Editor assigns the next free item number to the item, as well as a default position, and places it in your dialog window for you to drag to its final destination. The item is left selected.

It is also possible to create new items by duplicating existing items, or by pasting items from another dialog. For more on this, see the "Editing Dialog Items" section later in this chapter.

SPECIFYING A NEW ITEM

When you chose **Specify…** from the **New Item** sub-menu, the Editor displays a dialog that lets you choose the type of dialog item you want to create.



New Item Type
● OK Button ○ Static Text
○ Cancel Button ○ Edit Text
○ Button ○ Icon
○ Check box ○ Picture
○ Radio button ○ User Item
○ Control ○ List User Item
○ Pop Up User Item

Text: OK

Insert as item 1

☒ Enabled

Click on the radio button that corresponds to the type you want. For each item type, the dialog displays certain fields in the bottom left part of the window that let you fill in additional information for the item.

To set the item to be enabled or disabled, check or uncheck the **Enabled** checkbox in the lower left of the dialog.

For Button, Checkbox, and RadioButton items, you can enter the title of the item. The OK and Cancel button types fill in with the appropriate titles; these choices may be disabled, however, if the dialog already has an OK or Cancel button (see **Sorcery** below, also).

For Control, Icon, Picture, and Pop-up Menu User items, you can enter the resource ID of the referenced ('CNTL', 'ICON', 'PICT', or 'MENU') resource. In addition, a **Choose Icon ID** button lets you get a quick look at all the 'ICON' resources in the current file.

For Static or Edit text items, you can enter the text to be displayed.

For List User items, you can enter the cell size (in pixels) of the one-dimensional list.

For standard User items, there is no further information to be set.

Initially, the value of the **Insert as item** field is set to the first free item number in your dialog's list of items. For a well-formed item list (that is, one without any missing item numbers), this means that your new item will be appended to the list.

If you want to insert the item at some other position in the list, change the value of the **Insert as item** field to indicate which item number you want your new item to have. All items whose numbers are the same or higher than the number you specify will have their item numbers (positions) in the list bumped by one.

When you have finished specifying the information for your new item, click on the **Create** button to create it, or click on the **Cancel** button to do nothing.

Note: You cannot use the **New Item Specify...** dialog to create new Rectangle items for Control Panel dialogs. To do that, you must choose the **Rectangle** command in the sub-menu, which is only enabled if you are currently editing the Rectangle Layer (see "Control Panel Editing" for more on this).

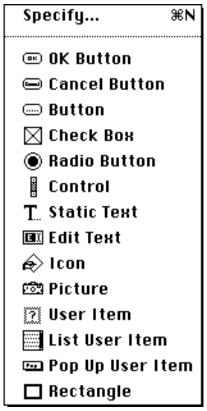
Sorcery: The TAB key cycles through the item type radio buttons. ⌘N is the keyboard equivalent for **New Item Specify...**.

⌘ . (period) is the keyboard equivalent of the **Cancel** button in the dialog.

When you open the **New Item Specify…** dialog, the initial item type is the first enabled radio button. Thus, to create and add an OK button and then a Cancel button (which are the first two radio button types) to an empty new dialog, type ⌘N followed by a Return, and then another ⌘N followed by a Return.

CREATING A DEFAULT-VALUED ITEM

To create an item of a given type, choose one of the commands in the rest of the **New Item** sub-menu. Each one creates a default-valued item of the same type as the command name.



OK Button creates the special enabled OK button with the title "OK". The button is placed in the lower right part of the dialog window, near where standard OK buttons might appear. If there is already an OK button in the dialog, this choice will be disabled.

Cancel Button creates the special enabled button with the title "Cancel". The button is placed in the lower right of the dialog window, near where standard Cancel buttons might appear. If there is already a Cancel button in the dialog, this choice will be disabled.

Sorcery: The positions and titles of the OK and Cancel buttons are copied from items 1 and 2 of a sample alert in Resorcerer's own resources, which you can edit to suit your tastes or language. To do this, run a copy of Resorcerer on itself, and edit the 'ALRT' whose resource name is "Sample Alert".

Button creates a standard enabled Button item with the title "Button".

Check Box creates a standard enabled Checkbox item with the title "Checkbox".

Radio Button creates a standard enabled Radio Button item with the title "Radio button".

Control creates a standard enabled Control item with a resource ID reference set to 0, and with dimensions of a vertical scroll bar.

Static Text creates a standard disabled Static Text item with the text "Static text" in it.

Edit Text creates a standard enabled Edit Text item with no text in it.

Icon creates a standard disabled Icon item with the resource ID reference set to 0.

Picture creates a standard disabled Picture item with the resource ID reference set to 0.

User Item creates a standard enabled User item.

List User Item creates an enabled List User item with a default cell size of 16 pixels high.

Pop-up User Item creates an enabled Pop-up Menu item with its resource ID reference set to 0.

Rectangle creates a Rectangle item for the Rectangle Layer of a Control Panel dialog. When editing standard dialogs or alerts, or the Dialog Item Layer of a Control Panel dialog, this choice is disabled.

DIALOG ITEM NUMBERS

Each item in an item list is assigned an *item number*, which is determined by the *position* at which the item is found within the item list (‘DITL’) resource (or the ‘nrct’ resource for Rectangle items). The item numbers are important, since they are what the Dialog Manager uses to identify to your application what item your user has interacted with in any given dialog.

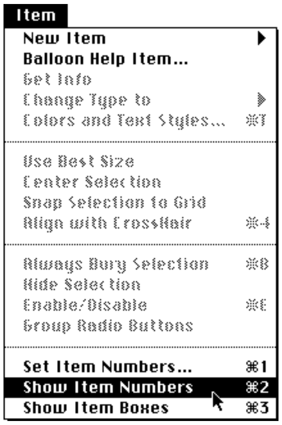
Unfortunately, item numbers are not part of any explicit data kept in the resource. They really only represent the ordering of items. The Dialog Editor temporarily turns these implicit numbers into explicit data that you can view and change while the dialog is open. When you close the editing window, the item numbers are used to determine the order in which items should be entered back into the ‘DITL’ resource. The Editor makes sure that there aren’t any mistakes in the item numbers such as duplicate, missing, or undefined values, and it will let you know if there are.

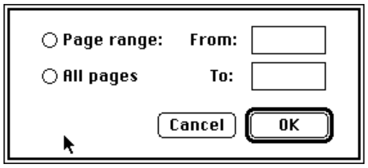
Rectangle items reside in a separate list from the standard dialog items, and hence have their own sequence of item numbers that exists in parallel with the standard dialog item numbers, beginning with the number 1. In most other respects, though, they behave as other dialog items do. In general, the order that the rectangles appear in the ‘nrct’ resource is immaterial.

Balloon Help items, although kept in the standard ‘DITL’ resource, do not affect the item numbering of standard dialog items in the list, even if a Help Item is found in the middle of the list. The Dialog Manager completely ignores them when reporting item numbers to your application.

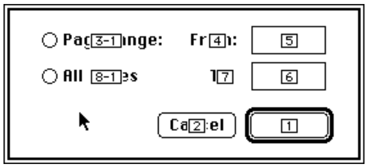
SHOWING ITEM NUMBERS

Normally, the Editor suppresses showing you each item’s number so that you can concentrate on the look of your dialog layout. You can tell the Editor to superimpose each item’s number on top of each item for easy viewing and setting. To do this, choose **Show Item Numbers** from the **Item** menu. Or you can use the Editor’s own **Preferences...** command in its first menu to automatically show item numbers whenever you open any dialog.





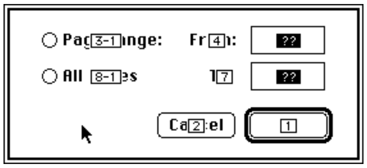
Normally, each item number is displayed in the center of its item's bounding box. However, you can choose to have the number placed in any corner or along the edge of the item's bounding box. For more on this, see the "Dialog Editor Preferences" section later in this chapter.



Note: When showing item numbers, any Radio Button items that have been grouped (see the **Group Radio Buttons** command in the **Item** menu) will also display the group number to the right of the item number.

Show Item Numbers mode is temporarily turned on for you when you ask to renumber items with **Set Item Numbers…** in the **Item** menu. They are made invisible again when you finish setting all item numbers (see the next few sections for more on this).

INVALID ITEM NUMBERS



During editing, it is possible for an item's number to become invalid. To make them easy to spot when item numbers are showing, the Editor displays invalid item numbers as a pair of white question marks on a black background.

The most common way that an item number becomes invalid occurs when you ask to renumber items (explained in the next few pages). Another time item numbers can become invalid is when you select an item whose number is in the middle of the item list and then you delete or cut that item. Depending on the action you ask the Editor to take with respect to higher item numbers, the Editor may invalidate all items that have item numbers higher than the lowest item number cut or cleared, thereby encouraging you to renumber them. (For more on deleting items, see the "Editing Dialog Items" section in this chapter).

Note: You can explicitly invalidate an item by setting its item number to 0.

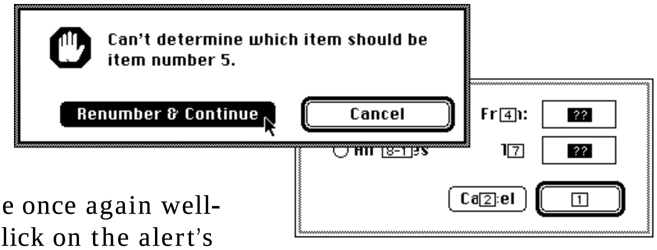
WHEN ITEM NUMBERS MUST BE VALID

During most editing, the Dialog Editor imposes no restriction on an item's number. Two items can have the same number; there can be holes in the sequence of numbers; and there can be invalid item numbers.

There are times, however, when the Editor requires that the numbers be valid. These times are

- before you can close and save the dialog and its item list,
- before you can duplicate the dialog,
- before you can generate starter code for the dialog, and
- before you can use the **Try out...** command.

Prior to attempting to perform any of these operations, the Editor ensures that the set of item numbers is in good order (literally). If it is not, the Editor lets you know which item is causing the problem. When it does alert you to a problem, you will usually want to **Cancel** whatever you started to do, and fix the item numbers yourself.



Can't determine which item should be item number 5.

However, you are also given the option of having the Editor re-order the items in such a way that the numbers become once again well-defined. To do so, click on the alert's **Renumber & Continue** button. How the Editor does this is completely arbitrary, although it does attempt to keep valid item numbers the same.

If you are using the Editor just to prototype a user-interface for which no application exists, you don't need to care about item numbers. If you are maintaining a dialog that already belongs to an application, you should not change any item's number unless you know exactly what you are doing. In the cases where the dialog remains open after an automatic renumbering, you can choose **Undo Renumber** to reset the invalid items to their previous values.

CHANGING INDIVIDUAL ITEM NUMBERS

The simplest way to assign a new item number to an item is to first ensure that **Show Item Numbers** mode is turned on and then select the single item to which you want to assign the new item number. Once it's selected, simply type the digits of the number you want the item to have. The digits accumulate as long as you type them within a short time of each other. If you make a mistake, either choose **Undo** to restore the previous number, or wait a moment and try typing again.

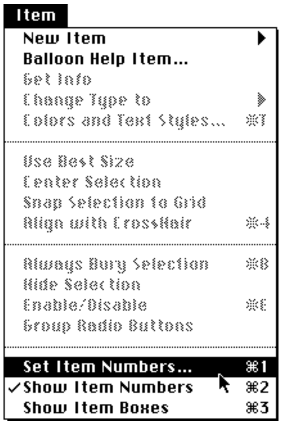
You can also assign a new item number when you open any item to edit the data associated with it (for more on this, see the "Changing item information" section later in this chapter), since the item's number is also editable in its information window.

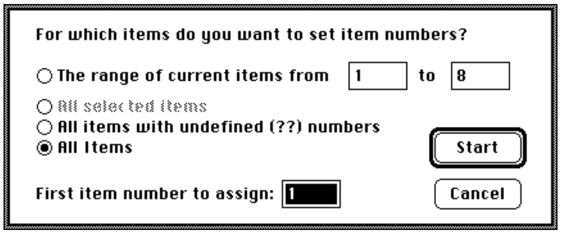
Note: If the number you assign to the item is 0 or negative, the item number is considered invalid, and drawn as a pair of white question marks within a black box.

For more than a few items, however, setting individual items can get tedious, and it is easier to use **Set Item Numbers...** from the **Item** menu. The next section explains how to use this command.

SETTING GROUPS OF ITEM NUMBERS ALL AT ONCE

To rearrange the item numbers of any of a variety of groups of items, choose **Set Item Numbers...** from the **Item** menu. The Editor asks you to specify which group of items you want to rearrange, and what the starting item number value should be.





For which items do you want to set item numbers?

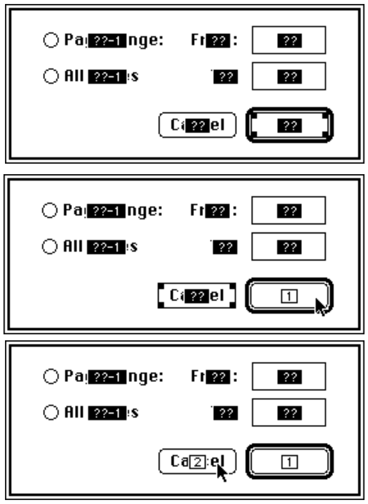
○ The range of current items from 1 to 8
○ All selected items
○ All items with undefined (??) numbers
● All Items

First item number to assign: 1

The groups you can specify are:

- all items,
- only items in the current selection,
- only items with already invalid item numbers, or
- only items whose valid numbers are in a given range.

Click on the **Start** button to begin renumbering. The first time you do this, the Editor will give you a hint explaining what to do next.



The Editor selects and invalidates the items to be renumbered, automatically turns **Show Item Numbers** mode on so you can see what you are doing, and then waits for you to reset all invalid item numbers. To reset an invalid item number, hold the ⌘ key down and click the mouse on the item as if you were selecting it. The next item number in the sequence is assigned to the item.

After assigning a new number to an item, the item is deselected and, if it covers any other item, it is sent it to the rear of the editing display list (i.e., it is temporarily buried).

If you make a mistake, you can choose **Undo** to restore the item numbers to what they were prior to choosing **Set Item Numbers…**.

During renumbering, you can perform most usual item editing tasks. In particular, you can hide and show items, which may be important if an item that you want to renumber is hidden outside the bounds of the dialog window where you can’t click on it.

You can tell when all items are renumbered by letting go of the ⌘ key. If you have just ⌘-clicked on the last invalid item, letting go of the ⌘ key turns **Show Item Numbers** mode off. Thus, if item numbers are still visible after you let go, there must still be an invalid item somewhere waiting to be assigned the next valid number.

RESETTING INVALID ITEM NUMBERS

At any time during editing, you can assign the next free item number to an item that has an invalid item number. **Show Item Numbers** mode must be turned on for this to work.

Hold the ⌘ key down and click the mouse on the item as if you were selecting it. If item numbers are showing, the new item number will be visible. If not, the item’s number will not change.

Sorcery: In addition to renumbering an invalid item number, ⌘-clicking on an item will send the item to the rear of the editing display list (if the item covers some other item), and will deselect it. If you want to keep the item selected, hold the Shift and ⌘ keys while clicking.

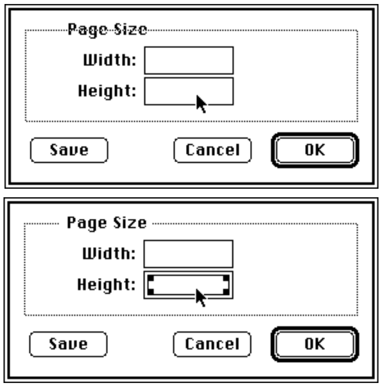
EDITING DIALOG ITEMS

The Dialog Editor keeps the list of items in your dialog in an internal form that is similar to how the Mac manages windows on the desktop. The items are always displayed from back to front, so that front-most items are fully visible if they overlap some other items in the list. Usually an item in front of another will obscure the background item; however, some items, such as User items, can be transparent.

While you are editing, the display list has no relationship to the order in which items are finally stored in the item list resource data. You can always rearrange the display list into that order when you want, using the **Update in Item Order** command, and the **Try Out...** command lets the Dialog Manager show you exactly what it would do with the items. During editing, though, you should not have to worry about the eventual order, which is determined solely by the item numbers of the items.

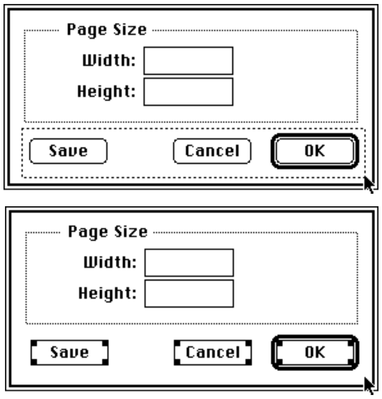
SELECTING ITEMS

To select an item in your dialog, click once within the confines of its item definition box. The Editor brings the item to the front of the item display list, and the box and its handles become visible to indicate that the item is selected. All other items will be deselected unless you are holding the Shift key down, in which case the item will be added to the current selection.



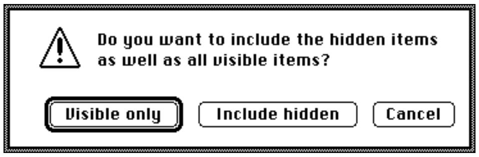
Note: The *item definition box* is the rectangle that is recorded in the ‘DITL’ resource data. In most instances, this box is the same as the bounding box of the item. The bounding box, however, can be larger for OK Buttons, Edit Text items, List User items, and PopUpUser items. In these items, the selection handles appear inset from the actual item corners.

To select a group of items, click the mouse in the dialog window background and drag the marquee (“marching ants”) rectangle out to some point. All dialog items whose item definition boxes intersect the selection rectangle will be selected, and all that don’t intersect will be deselected. If you are holding the Shift key down, the items will be added to the current selection.



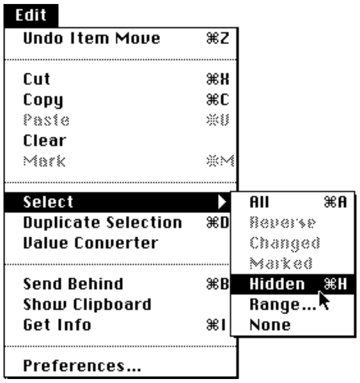
Note: Items that have been buried (see “Burying and Unburying Items” later in this section) are not selected by the selection rectangle unless they are completely *enclosed* by it (as opposed to just intersecting it).

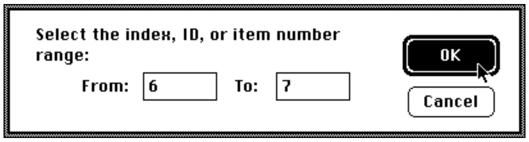
To select all items, choose **Select All** from the **Select** sub-menu in Resorcerer’s **Edit** menu. If there are any hidden items, the Editor will ask you if you intend to include them or not.


Do you want to include the hidden items as well as all visible items?

To select an item that is covered by another item, you must either move the covering item out of the way by dragging it or hiding it, or bury it so that it is behind the item you want to select in the display list. The easiest way to do this is to c-click on the covering item(s) until the covered item gets to the top.

To select all hidden items, choose **Select Hidden** from the **Select** sub-menu in Resorcerer’s **Edit** menu. Hidden items are items whose item definition boxes are empty, or are outside the bounds of the dialog window. Unless you have asked the Editor not to, it will warn you when you open a dialog that has any hidden items in it (for more on this see the “Editor Preferences” section later in this chapter).





Select the index, ID, or item number range:

From: 6 To: 7

To select a range of items by item number, choose **Select Range...** from the **Select** sub-menu in Resorcerer's **Edi**t menu. Enter the first number in the range, the last number in the range if different from the first, and then click OK. To select all items with invalid item numbers, use the range from item 0 to 0.

To select all items whose item information, such as titles or text, contains a match for a given piece of text, use Resorcerer's **Find All...** in the **Resource** menu (for more on this, see the earlier "Editing Resources" chapter).

You can also select an item when you check the **Selected** checkbox in the item's information window. For more on this, see the "Changing item information" section later in this chapter.

DESELECTING ITEMS

Typically, items are deselected whenever some other items are selected.

To deselect a specific selected item, hold the Shift key down and click once on the item. This deselects it without affecting any other items.

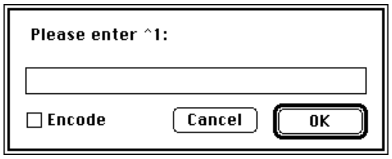
To deselect all items, click once in the background area of the dialog window, or choose **None** from the **Select** sub-menu in the **Edit** menu.

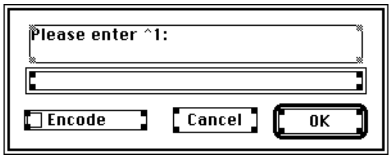
You can also deselect an item when you uncheck the **Selected** checkbox in the item's information window. For more on this, see the "Changing item information" section after this one. This is useful when you want to deselect a single hidden item while keeping it hidden.

Items marked for automatic burial are usually sent to the rear of the item display list when they are deselected; however, when Shift-clicking on such an item to deselect it, the Editor will constrain it to remain at the front of the display list.

ITEM HANDLES

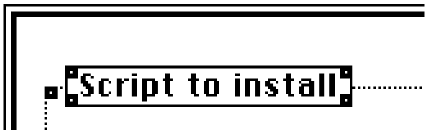
When you select an item, four *grow handles* appear at each corner of the item definition box. These handles are small squares that let you manipulate any of the four corners of the definition box with the mouse. In addition, they graphically indicate whether the item is enabled or not, and whether it is marked for automatic burying during deselection.

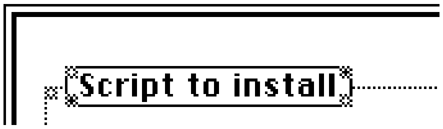




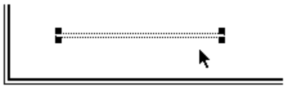
If the item is an enabled item, the handles are drawn in black. If the item is disabled, they are drawn in gray. For more on this, see the next section.

If the item is marked for automatic burying during deselection, a small dot is drawn in the center of the handle. For more on burying items, see the "Burying items automatically" section below.



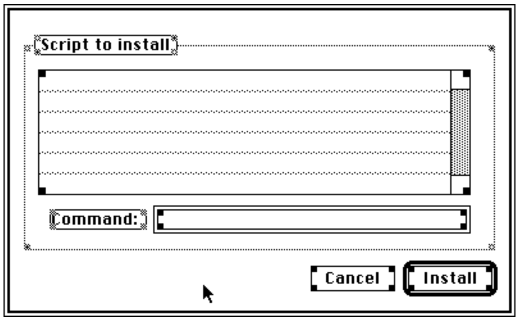


Normally, the grow handles are confined within the bounds of the item definition box. However, if the item's definition box is so narrow or thin that the handles would overlap, they begin to push outside the bounds to avoid overlapping so that you can always manipulate any corner unambiguously. Usually this applies only to very thin User or Picture items.



ENABLED VS. DISABLED ITEMS

Every item in a dialog's item list can either be *enabled* or *disabled*. During the time the Dialog Manager is running your dialog, enabled items are ones that report user interactions back to your program; the Dialog Manager takes care of disabled items internally.

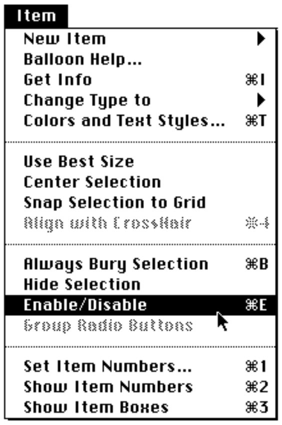


Generally, you want Static Text, Icon, and Picture items disabled. User items being used to define graphic bounds can usually stay disabled; other types of User items should be enabled if they represent something the user is going to interact with. Edit Text items should be enabled if your program wants to monitor the user's editing activity in the item. Edit text items still work if they are disabled, but only in the simple standard way.

Note: The user can always interact with Edit Text items in your dialog. If the Edit Text item is enabled, then the Dialog Manager informs your application of the latest interaction (e.g. a character was typed, the mouse was clicked, etc.) with that Edit Text item. If the item is disabled, your application must explicitly read the contents of the Edit Text field to know if the field has changed any.

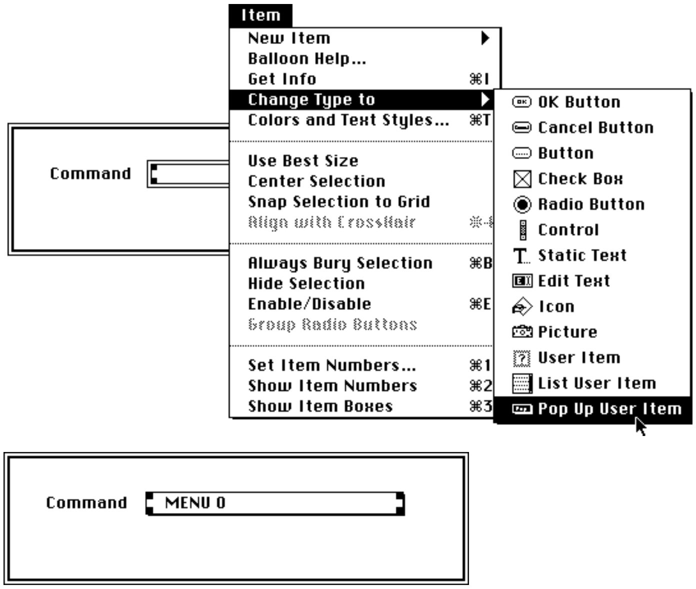
Enable/Disable in the **Item** menu lets you set the enabled bit of each selected item. If any item in the selection is disabled, then all items are enabled. If all items are already enabled, then the command will disable all selected items.

Enabled items have black selection handles. Disabled items have gray selection handles.



CHANGING AN ITEM'S TYPE

To change one or more items from one type to another, select the items to be changed, and choose the new type from the sub-menu of the **Change Type To** command in the **Item** menu. The types are changed without changing any item numbers, and if both original and final items have title text, it remains the same also.



Usually you will want to change the type of an item during the modification of an existing dialog, such as after deciding that the values in a particular Edit text field would be better represented in a PopUp menu. This lets you safely change a given item without disturbing the ordering of the rest of the item list. Another use for the **Change Type To** command is when you are using the Editor for the first time on a dialog you've built using some other dialog editor. Typically, if certain User items in the dialog are meant to represent List or Pop-up Menu items, you will want to convert the generic user items into the specific ones to get a better feel for how the dialog looks.

Arbitrarily changing one item to another will generally lose information that you may have to re-enter. For instance, if you decide to change a Button item into an Icon item, the text of the button title may be thrown out. Furthermore, the newly changed Icon item will probably not reference the proper 'ICON' resource in its data.

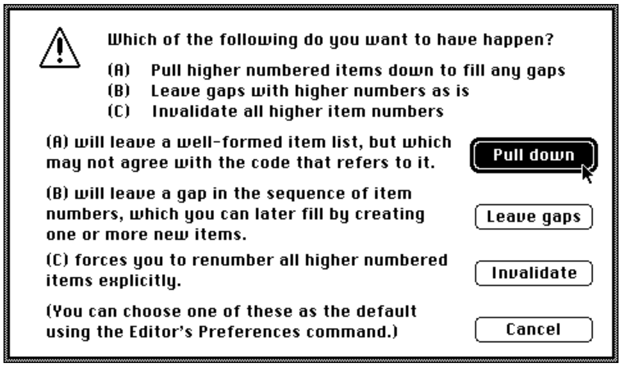
DELETING AND CUTTING DIALOG ITEMS

To delete one or more items, select them and tap the Delete key or choose **Clear** from the **Edit** menu. To cut one or more items, select them and choose **Cut** from the **Edit** menu. Cut items are placed in the clipboard so that you can paste them into some other dialog.

The information that is cut includes all related color and text style information kept in the item color table (‘ictb’) resource.

When you delete a selection of items, the Editor must decide what to do about all remaining items in your dialog whose item numbers were higher than the lowest numbered item you deleted. Remember that although item numbers eventually represent the ordering of items in the item list resource, while you are editing a dialog, the item numbers of each item are completely editable as if they were explicit data.

The Dialog Editor supports four possible courses of action. Generally, you will prefer one of these actions over the others, and you can set your preferred action using the **Preferences** command (for more on this see the “Editor Preferences” section later in this chapter). Initially, however, the Editor asks you which action you want:



Which of the following do you want to have happen?
(A) Pull higher numbered items down to fill any gaps
(B) Leave gaps with higher numbers as is
(C) Invalidate all higher item numbers

(A) will leave a well-formed item list, but which may not agree with the code that refers to it.

(B) will leave a gap in the sequence of item numbers, which you can later fill by creating one or more new items.

(C) forces you to renumber all higher numbered items explicitly.

(You can choose one of these as the default using the Editor’s Preferences command.)

- The most common action is to pull all higher numbered items down. That is, if you delete item 6 from your list, the old item 7 will become the new item 6, the old item 8 will become the new item 7, etc. This leaves your items well-numbered (ordered) if they were well-numbered in the first place. On the other hand, your smaller item list almost certainly is now out of sync with the application code that refers to it and which still expects the old item 7 to be the same when you run the dialog. Since the item list remains well-formed, the Editor cannot warn you about anything you might have forgotten.

- Less commonly, you might want to leave all item numbers of higher numbered items alone. This leaves a gap in the set of numbered items that you will have to fill at some later time. For example, if you cut item 6, the old item 7 remains numbered as item 7, etc.; there simply is *no* item 6 for the time being. If you forget to create a new item 6 to fill the gap, the Editor will alert you that the item is missing.

  When you create a new item, the Editor will attempt to fill the first hole in the list of items. Or you might want to fill the gap by reassigning the item number of some other item in the list. You can do this by either opening its item information and changing the item number there; or by selecting the other item and typing its new item number.

- Least commonly, when you delete an item, the Editor can invalidate the numbers of all items with item numbers higher than the lowest numbered deleted item, in order to encourage you to renumber them explicitly. This option is occasionally useful if you currently don't care about all later items in the list, and you want to put off thinking about what the actual ordering is eventually going to be. For more on invalid item numbers, see the previous "Item Numbers" section in this chapter.

- The last possibility is to ask the user which of the above three actions to take. This lets you decide what to do on a case-by-case basis. Because pulling higher-numbered items down in the list is the most common action for experienced users, the **Pull Down** button is the default button when the Editor asks you what to do.

COPYING AND DUPLICATING ITEMS

To copy one or more items to the clipboard, select them and choose **Copy** from the **Edit** menu. You can then paste them into any other dialog open for editing. If any item has custom text styles or colors (kept in the minor ‘ictb’ resource), these are copied or duplicated as well.

To duplicate one or more items, select them and choose **Duplicate Selection** from the **Edit** menu. The Editor will place the duplicated item(s) at an offset from the items duplicated. It deselects the items just duplicated, and selects the newly created ones.

Note: If you duplicate an item, drag it to another position, and then duplicate it again, the offset of the second duplicate is taken to be the same as the distance you dragged the first duplicate. This makes it a bit easier to space a set of checkboxes or radiobuttons that you create by duplicating an original.

The copied or duplicated information includes all related color and text style information kept in the item color table (‘ictb’) resource.

To duplicate an entire dialog (which includes all the items in it), use **Duplicate Resource Set** in the **Resource** menu. This is explained in the earlier “Duplicating an Open Dialog” section.

PASTING DIALOG ITEMS

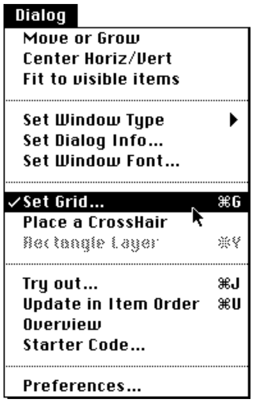
After cutting or copying a selection of items from a dialog to the clipboard, you can then paste them into another (or the same) dialog.

When the selection of items is pasted, the Editor appends the items to the current item list. The item numbers of the pasted items will be adjusted to reflect their new positions in the destination item list; however, if any of the item numbers were invalid, duplicated, or missing, these properties will remain after they are pasted.

THE UNDERLYING GRID

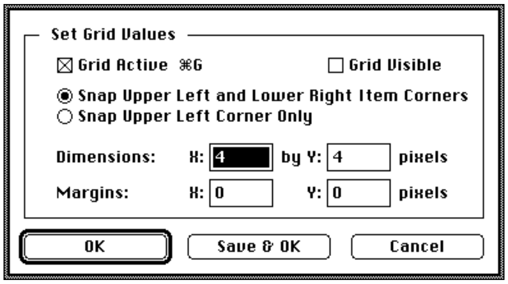
The Editor maintains an underlying coordinate grid that can be much coarser than the coordinates (which are in pixels) of the item positions. If you have enabled the grid, coordinate positions for items are snapped to the grid lines whenever you move or grow an item.

To see what your current grid settings are, choose **Set Grid...** from the Editor's first menu. This lets you set various grid parameters. The **OK** button makes the changes only to the front-most dialog window; the **Save & OK** button also installs the settings as your preference.



You can set different grid **Dimensions** for both the horizontal and vertical parts of the grid. A horizontal value of 4 or 8 and a vertical value of 8 or 16 are usually the best.

The origin of the grid is taken from the **Margins** X and Y coordinates.



Set Grid Values

☒ Grid Active ⌘G □ Grid Visible

● Snap Upper Left and Lower Right Item Corners
○ Snap Upper Left Corner Only

Dimensions: X: 4 by Y: 4 pixels
Margins: X: 0 Y: 0 pixels

To make the grid visible, check the **Visible** checkbox. Grid lines are drawn into the background of the dialog window, in blue on color systems. The margin is drawn in dark gray, or green on color systems.

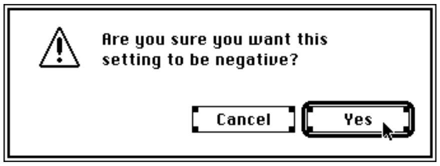
To turn the grid on or off, check or uncheck the **Grid active** checkbox, whose keyboard equivalent is the same as the **Set Grid...** command.

Usually, you only want to snap the upper left corner of an item's definition box to the grid; occasionally you will want all corners of items to conform to the grid lines. Whichever is the case, choose the appropriate radio button to set the Editor behavior.

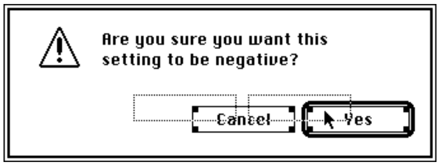
The X and Y grid sizes are saved in the dialog's 'DLGX' resource so that every dialog you edit can "remember" the settings you've chosen. Newly created dialogs are assigned your current preferred grid settings.

MOVING ITEMS

To move one or more items, click on them to select them and, while holding the mouse down, drag any one of the selected items to a new position within the dialog window. When you let go of the mouse, the items move to the new position, which may be adjusted to conform to the current grid settings. During dragging, an outline of each item's defining box follows the motion of your mouse.



Are you sure you want this setting to be negative?

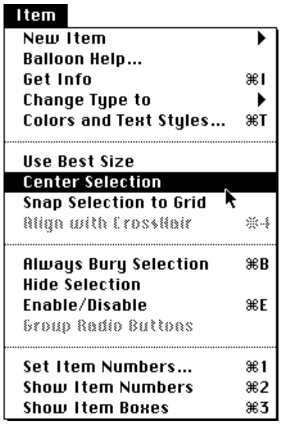


Are you sure you want this setting to be negative?

Note: The Editor will not let you drag any item completely outside the window bounds of the dialog window. However, if you are dragging the bounding box as if it were an item, some items internal to the group may be dragged completely outside the window bounds.

To constrain the movement of your selection of items, hold the Shift key down. If the first detected motion of your move is vertical, no horizontal change is allowed, and vice-versa.

To center an item in the dialog window, select it and choose **Center Selection** from the **Item** menu. If its item definition box is not centered horizontally already, it will be; otherwise, it is centered vertically. If more than one item is selected, the bounding box of all selected items is centered.

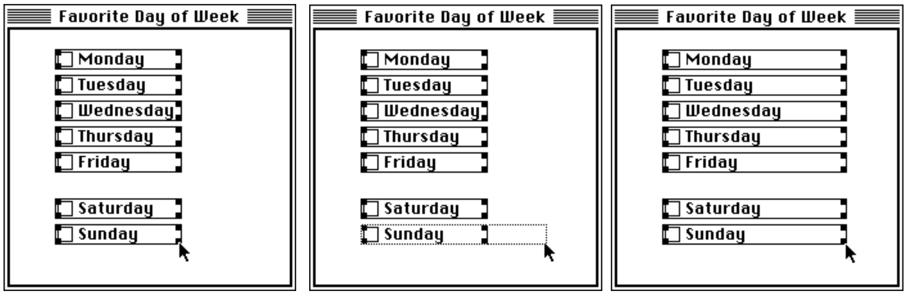


To move a selection of items one grid unit at a time, tap the arrow keys to bump the items one grid unit in the same direction as indicated by the key. The new position is not subject to being snapped to the underlying grid. For more on the arrow key mechanism, see "Growing items" below. To constrain the move to one pixel at a time, hold the Shift key while tapping the arrow key.

You can also move an item by hiding it or aligning it with the crosshair. See the following sections for more on how to use these features.

GROWING ITEMS

You can grow one or more items from any corner. First select the items you want to grow. Click the mouse on the handle in the corner you want to grow (the opposite corner will remain anchored), and drag the corner to its new position. This new position is conformed to the current grid settings, and all selected items are grown by the same amount.



To constrain the growth to one axis or the other, hold the Shift key down. If the first detected motion of your mouse is vertical, no horizontal change is allowed, and vice-versa.

The Dialog Editor supports a number of different ways of growing more than one item simultaneously:

Fine Adjustment

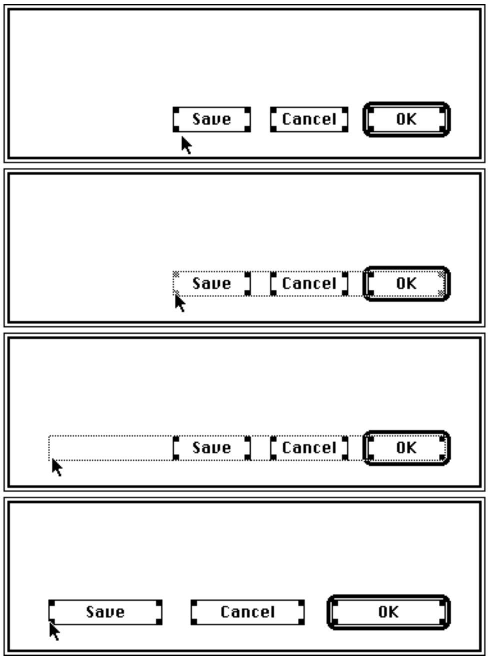
Fine adjustment lets you use the arrow keys on your keyboard to grow all selected items one grid unit or one pixel at a time. If the mouse is directly over some selected item's handle, that corner and all analagous corners in the other selected items grow by one grid unit if the Shift key is not down, or one pixel if the Shift key is down, in the direction of the arrow key used. Tapping on any arrow key continues to affect the position of the item corners until you move the mouse from its original position, or do any other action other than tapping an arrow key.

Anchored Growing

Anchored growing (illustrated in the above figures) is the way many object editors (e.g. MacDraw®) work. To grow more than one item, select the items to be grown and grow any one of them. The Editor applies the proportionate change in size of the item you grow to all other selected items. For instance, if you extend the lower right corner of one item of the selection, all other selected items will be extended from their lower right corners, while their upper left corners remain anchored.

Group Stretching

The problem with anchored growing is that quite often you will want to change the positions of adjacent items so that they don't overlap after being grown, and so that the inter-item spacing changes proportionately as well.

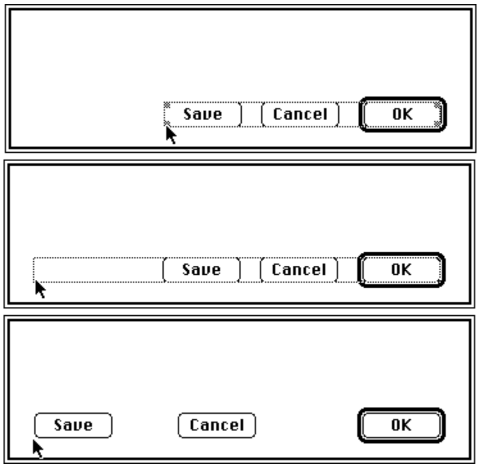


To stretch a group of items, select the ones you want to stretch and group them by holding down the Option key (see the next section, "Grouping items", for more on this). The group's bounding box becomes visible, with its own handles, which you can click on and drag to grow the group as a whole. This is very useful for growing groups of related items, particularly groups of buttons.

Suppose, for example, you want to change the sizes of three buttons in a dialog. Select them all, hold the Option key down, and click and drag one of the handles in their bounding box's corners to stretch all three buttons. The buttons closest to your mouse move so as to maintain a proportional spacing between the two.

Position Magnification

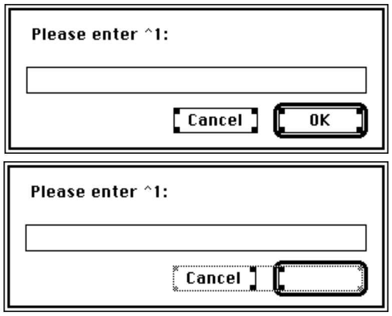
Group stretching generally doesn't make a lot of sense if your selection of items contains many different types. If it does, then you will likely want to magnify their positions while constraining the individual item sizes to stay constant. This lets you *explode* the items in a dialog, which is only rarely useful.



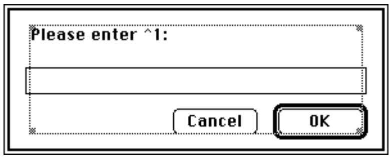
Group the items in your selection as before by holding the Option key down. Then click on one corner of the now-visible bounding box of the selection, and extend the corner to a new position. *After* the start of the drag, but prior to letting go of the mouse, press and hold the Shift key to constrain the item sizes, and then let go of the mouse button.

Note: Items with an edge of their defining box coincident with the edge of the grown bounding box remain anchored to the bounding box's edges, wherever you drag them.

TEMPORARILY GROUPING ITEMS



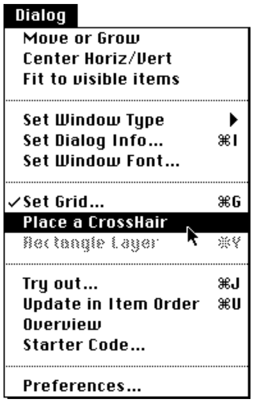
To temporarily group a set of items, select them and hold the Option key down. During the time the key is down, the Editor displays the bounding box of all selected items and treats it as if it were a single selected item. It has its own handles in all four corners; you can move it or grow it in the same way as other actual items (see below).

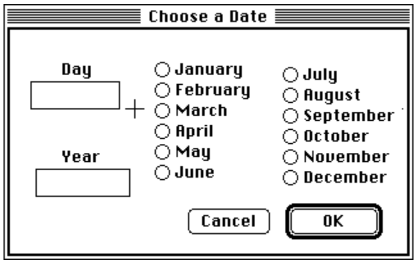


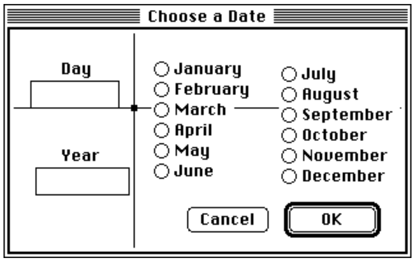
If no items are selected, the Editor displays the bounding box of all items in your dialog, including any hidden ones.

USING THE CROSSHAIR TO ALIGN ITEMS

To drop a crosshair into the background of your editing window, choose **Place a Crosshair** from the Editor's first menu. The mouse cursor becomes a cross to indicate that you should click at the position in the editing window where you want the crosshair to appear.







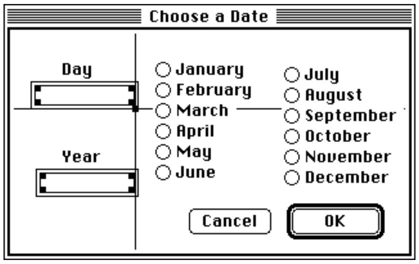
The crosshair consists of a vertical and horizontal line (drawn in red on color systems) that you can use to align any selection of items. Whenever the mouse cursor passes over a visible portion of the crosshair, the cursor turns into a cross to indicate that you can click on that part of the crosshair or portion of it to move it.

To move the vertical part of the crosshair horizontally, click on the vertical line and drag it to the position you want. Similarly, to move the horizontal part of the crosshair vertically, click on it and drag it to the position you want.

To move both horizontal and vertical parts of the crosshair, click on and drag the handle displayed at their intersection.

Note: Normally, the crosshair position is not subject to the underlying grid; however, when you drag it you can constrain its position to the grid by holding down the Shift key.

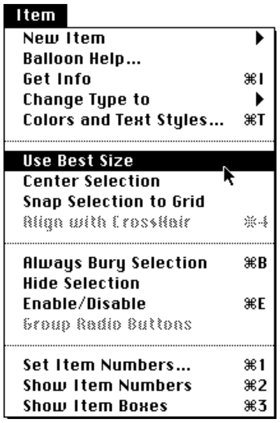
While either part of the crosshair passes through your dialog window, the **Place a Crosshair** command remains checked. To get rid of the crosshair, choose the command again, or pick the crosshair up at its origin and drag it outside the dialog window.



To align an item's vertical edge with the vertical part of the crosshair, place the crosshair where you want it, select the item or items to be aligned, and double-click on the vertical crosshair line (or choose **Align with Crosshair** from the **Item** menu). For each selected item, the right or left item edge that is closest to the vertical part of crosshair will be changed to conform to the coordinates of the crosshair. Similarly, to align the horizontal upper or lower edge of a selection of items with the horizontal part of the crosshair, double-click on the horizontal line.

Sorcery: Normally, items being aligned are grown larger or smaller to bring the item edges into alignment with the crosshair; however, if you want to constrain the items to maintain their sizes, hold the Shift key down before double-clicking.

USING THE BEST SIZE FOR ITEMS



With the exception of generic User items and Control Panel Rectangle items, there is usually a preferable size for an item's definition box, given the item type and its internal data. To change an item's definition box to the best size for that type of item, select the item or items and choose **Use Best Size** from the **Item** menu.

Checkbox, Radio Button, and Button items, including OK and Cancel buttons, are adjusted to the number of lines and width of the widest line in their titles plus some margin on each side of the title. If more than one of the same item type is selected, however, the width of all items of that type is conformed to the width of the widest item. This gives them a consistent look.

Note: Vertical groups of check boxes or radio buttons should usually have bounding boxes aligned on the right as well as left hand sides if you are designing dialogs that are to be run in a reversed script system.

Icon, Picture, Control, and PopUp items all refer to other resources in the file. These other resources contain either implicit or explicit information about their sizes. For instance, black and white icons are always 32 by 32 pixels, whereas general color icons can be any size; pictures contain frame information that specifies the size of the picture; control templates specify the size of the control; and standard menus have a calculated width that depends on the widest item in the menu.

Your dialogs will look the best if the items in the item list that refer to these resources have item definition boxes that are the same sizes as the resources that they refer to.

Static and Edit Text items display text in a given font (usually the System font) using word wrap to fill in as many lines as will fit in the item’s width. In this case, **Use Best Size** keeps the item width the same but increases or decreases the height of the item so that the eventual height is large enough to show all the item’s text. If there is no text in the item, which is very often the case for Edit Text items, then the height of the item is adjusted to the nearest integral number of text lines that will fit.

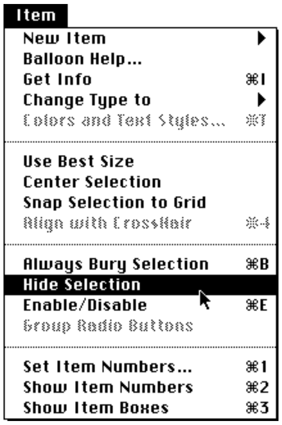
List User items are always adjusted so that the item definition box has a height that is a multiple of the list’s cell size.

HIDING AND SHOWING ITEMS

There are various conditions that can keep an item from being visible in its dialog. In each of these cases, we say the item is *hidden*.

The most obvious condition is when the item definition box is completely outside the bounds of the dialog window in which it would otherwise be displayed. Items that are outside like this can be either *formally* hidden or *informally* hidden. Formally hidden items are ones whose X coordinates have had 16384 added to them, which is what the Mac toolbox routine, `HideDItem`, does to hide an item. Informally hidden items are items with positions just outside the bounds of the dialog window.

Another type of hidden item occurs when the item has an item definition box with non-standard coordinates: either the box is empty or it is one whose “left” coordinate is larger than its “right”, or its “top” coordinate is below the “bottom”. It is possible to create such an item by directly editing its position data in the item’s information window, although with the exception of certain types of User items there are generally no good reasons to do so.



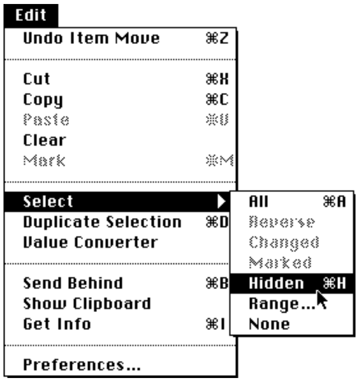
To formally hide an item or group of items, select them and choose **Hide Selection** from the **Item** menu. If the selection already consists of hidden items only, the title of the menu command changes to **Show Selection**.

To informally hide an item or group of items, enlarge the bounds of your dialog window (see “Changing dialoginfo” earlier in this chapter for more on how to grow a dialog window), drag the selection of items outside the old window bounds, and then drag the window bounds back to their old size. The Editor does not let you drag an item outside of the window bounds.

Of course, you can also explicitly set the coordinates of any item by opening its item information window and setting them there so that the item is positioned outside the window bounds. The item information window also contains a checkbox that lets you formally hide the item. For more on this, see the “Changing item information” section later in this chapter.

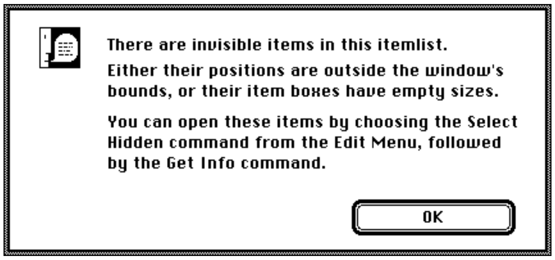
To show a formally or informally hidden item, you need to select it and then choose **Show Selection** from the **Item** menu. Formally hidden items are offset -16384 pixels in the X direction to bring them back within the confines of your dialog window. Informally hidden items are pulled into the window by whatever amount is necessary. An item with an empty or negative box is pulled into the dialog window's bounds, but will still not be able to be seen. In this case you will have to explicitly change its box's position data after opening its item information window.

To select *all* hidden items, use **Select Hidden** in the **Select** sub-menu. Once your hidden items are selected, you can use **Show Selection** to view them; or you can choose **Get Info** from the **Item** menu to open their item information windows.



To show one specific hidden item, you need to select it by itself. If it is the only hidden item, use **Select Hidden** as above. If there are lots of other hidden items, you will probably have to select it by using its item number. To do this, use the **Select Range...** in the **Select** sub-menu to select it by item number. You can then show it or open its item information for editing.

When you open a dialog with hidden items of any of these three types in its item list, the Editor alerts you to their presence since you otherwise might not know about them.



There are invisible items in this itemlist.

Either their positions are outside the window's bounds, or their item boxes have empty sizes.

You can open these items by choosing the Select Hidden command from the Edit Menu, followed by the Get Info command.

Note: The **Preferences** command in the Editor's first menu let's you choose the manner in which the Editor alerts you. For more on this, see the "Editor Preferences" section later on in this chapter.

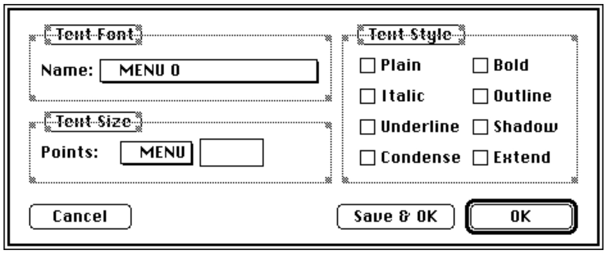
Sorcery: If you are not sure whether a given dialog has any formally or informally hidden items, hold the Option key down to display the bounding box of the current selection, or if there are no selected items, the bounding box of all items in the dialog (see ”Grouping items”). If the bounding box extends past the edges of the dialog window, you know that there are hidden items in that direction.

BURYING AND REBURYING ITEMS

Most of the time, items in your dialogs won’t intersect one another. Sometimes, though, a dialog’s item list has an item that overlaps or completely covers other items. This commonly occurs when an item is intended to be displayed in the background, while an active item lies on top of it. Usually, the background item is a Picture or User item, while the foreground consists of groups of related controls or text items.

One of the most common times that items cover other items is when you employ a User item and a Static Text item to designate the bounds of a graphic rectangular *panel* that surrounds groups of related dialog items. The Static Text item is placed across the top line of the User item in order to give the panel a label. This is sometimes called a *cluster* (especially in MacApp programs).

For complicated dialogs, panels are of great help in organizing the options you present to your user. Many of Resorcerer’s own dialogs use this technique. In the figure here, the three user items overlap the three Static Text panel labels, and completely cover a variety of other items. Because the User items are selected, they are in front of the items they cover. This keeps you from being able to select the covered items with the mouse.



Name: MENU 0

Points: MENU

When a covering item is at the front of the Editor’s display list, you must bury it (that is, send it to the rear of the editing display list) before you can use the mouse to select any items that the front-most item covers. There are two ways to do this: either bury the item temporarily, or mark the item as one you want buried automatically since it presumably is a background item.

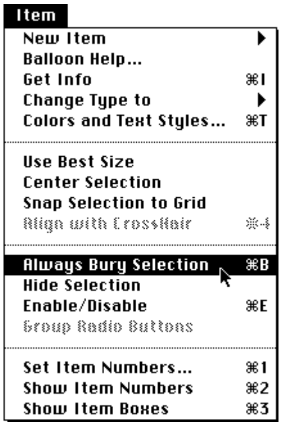
Temporarily burying items

To bury an item temporarily, hold the ⌘ key down and then click the mouse on the item to send the item to the rear of the display list. The items that it covered now become accessible to your mouse clicks. By ⌘–clicking on each new front-most item, you can rotate through a set of overlapping items without changing any of their positions.

Note: If you have **Show Item Numbers** mode turned on, the Editor also uses ⌘–clicking to assign the next free item number to any item that has an invalid item number. If the item is about to be buried as well, the Editor waits a half second before burying, so you can see the new item number before it possibly disappears behind the new item brought to the front of the display list.

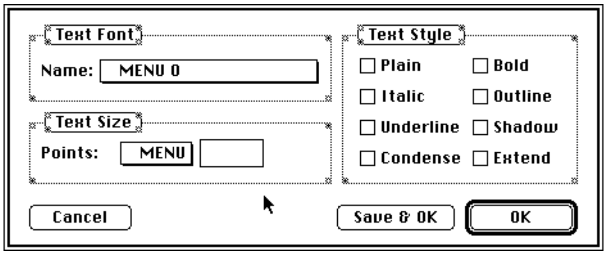
Sorcery: Normally, ⌘-clicking to bury an item also deselects it. You can keep the item selected by holding the Shift key down along with the ⌘ key when you bury the item.

Keeping background items buried



Always Bury Selection in the **Item** menu lets you mark selected items as ones that you want to keep in the background, where they won't cover other items in your dialog while you're editing.

To bury one or more unburied items, select them and choose **Always Bury Selection**. Each selected item is sent to the rear of the display list, deselected, and marked as an item that you are interested in burying again.



When any item is selected, you can tell whether it has been marked for automatic burying by the presence of a small dot in the center of each of the item's corner grow handles. Selected items, whether marked or not, are always brought to the front of the display list.

Once an item has been buried and marked, you can select it by clicking on it, or by using the selection rectangle. However, in the latter case, you must completely *enclose* the buried item with the selection rectangle to select it. This lets you more easily select groups of unburied items that are in front of the buried item.

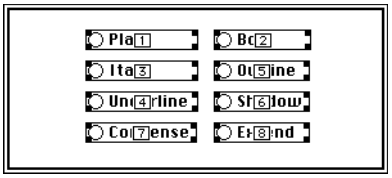
The Editor automatically reburies marked items when they are deselected. The most common time this happens is when you click in the background of the dialog window, which deselects all items.

If *all* items in the selection are marked for burial, the **Always Bury Selection** command in the **Item** menu changes to **Unbury Selection**, so that you can remove the selection from the collection of items marked for automatic burial.

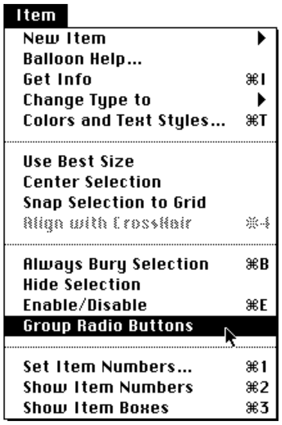
GROUPING AND UNGROUPING RADIO BUTTONS

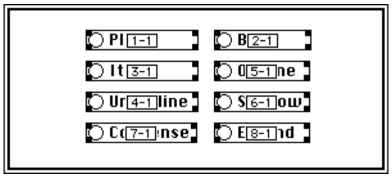
Radio buttons are intended to work together in groups of two or more, where only one button in the group can be on at any one time. When the Dialog Manager runs your dialog, however, it treats radio buttons as unconnected controls, requiring that your application take the responsibility of managing them as a related group.

The Dialog Editor lets you assign radio buttons to different groups so that it can create proper radio button management code for you, and so that it can better simulate the feel of your dialog when you use the **Try Out...** command.



When you create a Radio Button item, it is considered *ungrouped* until you group it. To assign the next free group number to one or more ungrouped Radio Button items, select the items and choose **Group Radio Buttons** from the **Item** menu. The actual group number assigned is unimportant, so long as it distinguishes the new group from all other Radio Buttons in the dialog.





To view radio button group numbers, use **Show Item Numbers** in the **Item** menu. If the Radio Button item is grouped, its group number follows its item number. If it is ungrouped, no group number shows.

If any Radio Button items in the selection are already assigned a group number, the menu command changes to **Ungroup Radio Buttons**.

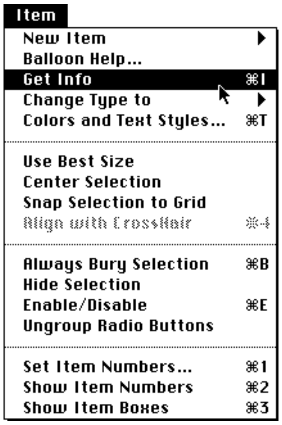
There is no provision in the 'DITL' resource for keeping radio button grouping information. Therefore, the Editor keeps group numbers in the 'DLGX' resource. The numbers are not needed by or accessible to the Dialog Manager.

CHANGING ITEM INFORMATION

Each item in a dialog item list has information associated with it, such as its position, whether it is enabled or not, any resource ID of the resource it references, etc. Some of this information is not normally visible while you are editing the items graphically.

Note: In particular, hidden items can sometimes only be edited by opening their item information windows.

To open the information for any item or group of items, select the group and choose **Get Info** from the **Item** menu.



To open the information for any given item regardless of how many others are selected, double-click on that item.

The Editor displays information for each item opened in a separate non-modal dialog window with a GoAway box and a **Cancel** button. As you make changes to values displayed in the information window, they are reflected back in the dialog. And as you make changes to items in your dialog, their item information windows are updated immediately.

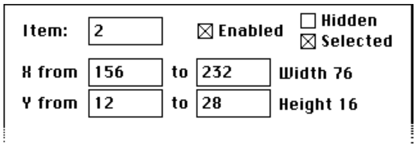
To save the changes you've made to an item since you opened its information window, click in the window's GoAway box. To throw away any changes you've made to an item since you opened its information window, click on the **Cancel** button.

Under various circumstances, the Editor automatically closes your item information windows as if you had asked to save the changes. The most common of these occasions is when you ask the Editor to close the dialog you're editing.

STANDARD ITEM INFORMATION

The information displayed for each opened item always includes the current item number; the coordinates of the item definition box; the width and height of the box as computed from these coordinates; and whether or not the item is selected, hidden, and/or enabled.

To change an item's item number, enter the new number in the appropriate field. A value of 0 (or "??") represents the unknown or invalid item number (for more on this, see the "Dialog Item Numbers" section earlier in this chapter). The number you enter can be equal to another item's number in the short term; however, the Editor will complain later on about any duplicated item numbers when you try to close or try out your dialog.



Item: 2 ☒ Enabled □ Hidden ☒ Selected

X from 156 to 232 Width 76

Y from 12 to 28 Height 16

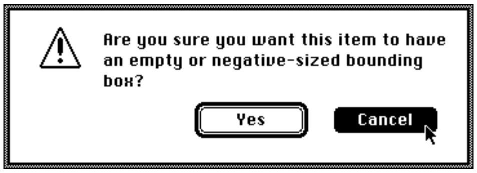
The **Enabled** checkbox enables or disables the item. It is equivalent to using the **Item** menu's **Enable/Disable** command. The Dialog Manager reports user interactions to your application for a dialog's enabled items only.

The **Selected** checkbox toggles the item's selection status back in your dialog, without affecting any other items. You can use this to select and deselect hidden items.

The **Hidden** checkbox hides a visible item when checked, or shows an invisible item when unchecked.

Note: The **Selected** and **Hidden** checkboxes, as well as the editable item number, do not represent actual data to be recorded in the eventual item list resource. These controls are there to make editing easier while you have the dialog and its items open.

As you change the values of the coordinates of the item definition box, the item is redrawn using the new coordinates back in your dialog. The Editor does not keep you from setting the values so that the rectangle is empty or inverted, but it will issue a warning in case it's an error.

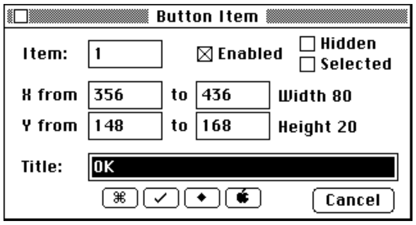


Are you sure you want this item to have an empty or negative-sized bounding box?

If you want to see the exact coordinates of an item while you drag it, open its information window and drag the window off to the side. Click on the dialog window to make it active again. As you drag the dialog item to different positions, you can see the new coordinates in the item information window.

TYPE-SPECIFIC ITEM INFORMATION

Below an item's standard information in the information window the Editor displays various fields and controls that depend on the type of item the information window represents.



Button Item

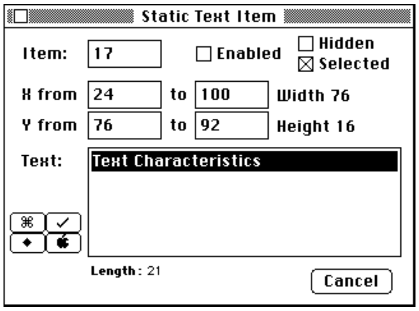
Item: 1 ☒ Enabled □ Hidden □ Selected

X from 356 to 436 Width 80

Y from 148 to 168 Height 20

Title: OK

For Button (including OK and Cancel Button), Radio Button, and Checkbox items, the information window displays a text field in which you can set the title of the control. Below the title field are some buttons that let you enter certain generally untypable characters that are occasionally useful in a control title. Clicking on the button labeled with the character is exactly like typing a key with that character on it.



Static Text Item

Item: 17 □ Enabled □ Hidden ☒ Selected

X from 24 to 100 Width 76

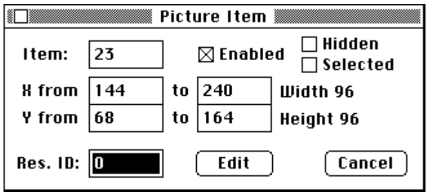
Y from 76 to 92 Height 16

Text: Text Characteristics

Length: 21

For Static Text and Edit Text items, the information window displays a large text field in which you can type the text of the item. Special buttons are available for entering commonly used but untypable characters. The length of the text in bytes is also visible.

Note: The maximum length in characters (bytes) that any single Static or Edit Text item may be is 240.



Picture Item

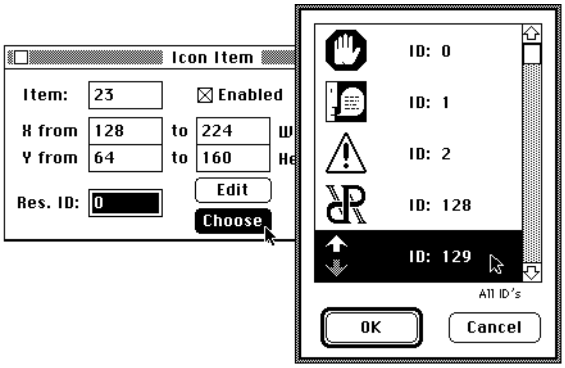
Item: 23 ☒ Enabled □ Hidden □ Selected

X from 144 to 240 Width 96

Y from 68 to 164 Height 96

Res. ID: 0

For Control, Icon, Picture, and PopUp Menu User items, the information window displays a field in which you can enter the resource ID of the referenced resource: 'CNTL' resources for Control items; 'ICON' or 'cicn' resources for Icon items; 'PICT' resources for Picture items; and 'MENU' resources for PopUp Menu User items, respectively. In all cases, the **Edit** button lets you open the referenced resource for editing, or it creates and opens a new resource with the given resource ID if none is found in the same file as your dialog.



Icon Item

Item: 23 ☒ Enabled

X from 128 to 224

Y from 64 to 160

Res. ID: 0

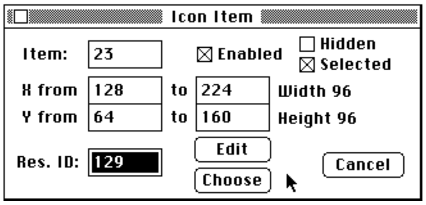
ID: 0

ID: 1

ID: 2

ID: 128

ID: 129



Icon Item

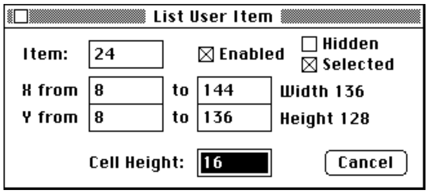
Item: 23 ☒ Enabled □ Hidden ☒ Selected

X from 128 to 224 Width 96

Y from 64 to 160 Height 96

Res. ID: 129

In addition, a **Choose** button is available in the Icon item information window. When you click on it, the Editor displays a list of all 'ICON' resources in your file, along with their IDs, so that you can choose the resource ID of the icon you want the item to display. The listing also includes the three standard system icons and their IDs, so that you don't need to remember which is which. After selecting the icon you want, click in the OK button, or simply double-click on one of the icons to install its ID back in the resource ID field of the item information window.



List User Item

Item: 24 ☒ Enabled □ Hidden ☒ Selected

X from 8 to 144 Width 136

Y from 8 to 136 Height 128

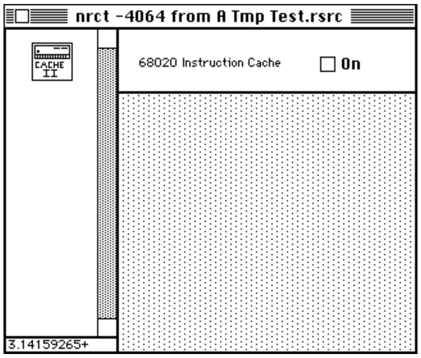
Cell Height: 16

For List User items, the item information window displays a field in which you can enter the list's cell size. This information helps the Editor better display the list according to how you plan to use it in your application.

User items and Rectangle items have no other type-specific information.

Note: Remember that List User items and PopUp User items are kept simply as User items in your dialog's item list ('DITL') resource data. The Editor keeps the item information for these items in its own private 'DLGX' resource.

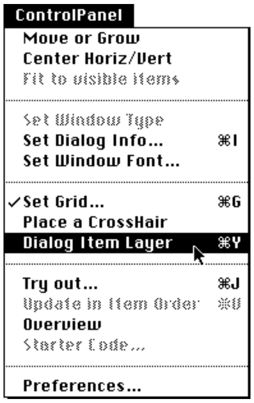
CONTROL PANEL EDITING



To edit a Control Panel dialog, open or create a resource of type ‘nrct’ (Rectangle List) with ID -4064. The Dialog Editor opens the ‘nrct’ along with its accompanying ‘DITL’ -4064, and displays them in a dialog window that simulates the look of the standard Mac (System 6.0) Control Panel desk accessory. Unlike other dialogs, there is no template (e.g. ‘DLOG’) resource that specifies the type of window to display the dialog items in.

The Control Panel desk accessory uses the rectangles in the ‘nrct’ resource to divide the content area of the Control Panel dialog into separate pieces. Each rectangle is framed with a two-pixels-wide border. Thus, adjacent rectangles must *overlap* by 1 pixel in order for them all to seamlessly fit together with a uniform border between them.

Since the Control Panel desk accessory always draws the rectangles in the background, and since by design they are supposed to completely cover the right hand area of the window, the Dialog Editor segregates the rectangles into their own editing *layer*, distinct from the editing layer reserved for the dialog items in the ‘DITL’ resource. This makes using the various selection tools much easier when editing the dialog items.



When you open the Control Panel dialog, the Editor initially sets the current layer to the Rectangle layer. You can tell which layer is current by looking at the title of the simulated Control Panel dialog, which will show you which resource you’re editing: ‘nrct’ -4064 or ‘DITL’ -4064.

To switch to the Dialog Item layer, choose **Dialog Item Layer** from the ControlPanel menu. When you are in the Dialog Item Layer, the menu command changes to **Rectangle Layer**, so you can switch back.

Sorcery: The Editor looks up and displays any ‘ICN#’ -4064 it finds. You can edit this icon by double-clicking on it in the Control Panel editing window.

SETTING ITEM COLORS AND TEXT STYLES

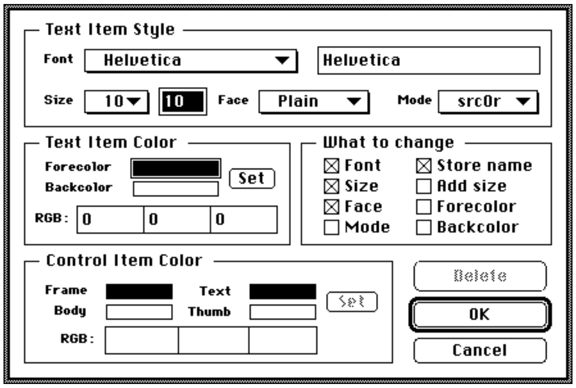
When you edit a dialog, the Editor automatically looks for and loads the 'dctb' or 'actb' (dialog or alert color table) resource that has the same ID as the 'DLOG' or 'ALRT' you're opening. It also looks for and loads the 'ictb' (item color table) resource in which to keep extra color and text style information for any text or standard control items in the dialog's item list. If either of these optional minor resources is not found, the Editor creates default-valued ones, which are *temporarily* added to your file.

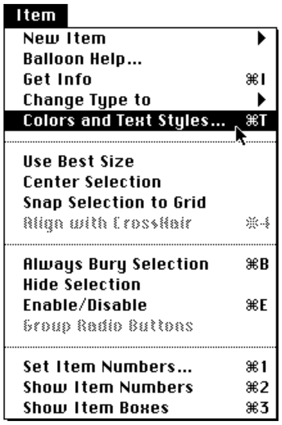
Note: The format of the 'ictb' resource is documented in the Dialog Manager chapter of Apple's *Inside Macintosh*, vol 5, pp. 280-281.

Note: In order for the Dialog Manager to take note of the 'ictb' resource for your dialog, the 'dctb' (or 'actb') resource must also exist.

Basically, the 'ictb' resource is a table of entries, one per dialog item, that allows you to assign different text fonts, sizes, styles, modes, or colors to any dialog's Static or Edit Text items; and to assign colors to the various parts of standard dialog controls, such as Button, Radio Button, or Checkbox items. Other types of items are currently ignored.

To create or change entries in this table for one or more items, select the items first and then choose **Colors & Text Styles...** from the **Item** menu. The command displays a dialog that consists of two parts. The first part, in the upper half of the dialog, applies to all Static and Edit Text items in the selection. The second part, in the lower left of the dialog, applies to all Button, Radio Button, Checkbox, and Control items in the selection.


Text Item Style
Font: Helvetica ▼ Helvetica
Size: 10 ▼ 10 Face: Plain ▼ Mode: srcOr ▼
Text Item Color
Forecolor ■ Backcolor □ Set
RGB: 0 0 0
What to change
☒ Font ☒ Store name
☒ Size □ Add size
☒ Face □ Forecolor
□ Mode □ Backcolor
Control Item Color
Frame ■ Text ■
Body □ Thumb □ Set
RGB:
Delete
OK
Cancel



If the selection has no text items in it, the upper part of the dialog is disabled and grayed out. If the selection has no standard control items in it, then the lower part of the dialog is disabled and grayed out.

In the lower right corner of the dialog are standard **OK** and **Cancel** buttons, and a **Delete** button. The **Delete** button lets you delete any existing entries in the item color table ('ictb') resource for all currently selected items. If there are no entries for any items in the selection, then the **Delete** button is disabled.

SETTING TEXT STYLES

The default behavior of the Mac Dialog Manager is to display Static and Edit Text items in the current font and size (which is usually the System font), using black foreground and white background colors. If you want one or more text items to have a different look, select them and choose **Colors and Text Styles...** from the **Item** menu.

The **Colors and Text Styles** dialog contains three panels, labeled **Text Item Style**, **Text Item Color**, and **What to change**, all of which pertain to the text items in your selection. If there are no text items in your selection, these panels will be disabled. Another panel, labeled **Control Colors**, does not pertain to text items.

Setting Text Item Style

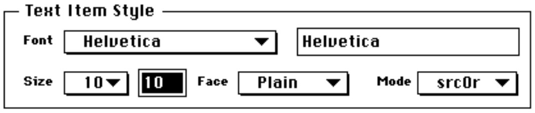
The **Text Item Style** panel lets you specify the font, size, style, and Quickdraw drawing mode that you want the Dialog Manager to use to display the selected Static or Edit Text items.

Note: When specifying a text style, make sure to check the appropriate bits in the **What to Change** panel. The Editor will remind you if you forget.

To set the font, choose from the font names in the **Font** popup menu. These names are the fonts you currently have installed in your System file. If for some reason you need to specify a font that you don't have in your own System file, you can type its name into the accompanying text field if the **Store Name** bit is set, or the font number if not.

Note: Remember that if you expect your program and the dialogs it runs to be widely used, you should not rely on your users having the same fonts installed in *their* System that you may have in yours. The exception to this is font number 1, the application font. Avoid exotic fonts, and avoid using font sizes less than 12 pts (if your user doesn't have the font, the Dialog Manager will default to the System font size, which is usually at least 12 pts).

To set the font size, choose from the standard font sizes in the **Size** popup menu. Sizes that represent real as opposed to scaled fonts are displayed in outline in the menu. If a size is not represented in the menu, you can type your custom size into the accompanying field.



Text Item Style

Font: Helvetica ▼ | Helvetica

Size: 10 ▼ | 10 · Face: Plain ▼ · Mode: srcOr ▼

To set the typeface, choose some combination of styles from the **Face** popup menu. More than one entry in the menu can be checked, but the popup menu only displays the first checked entry when you're not popping it up.

To set the Quickdraw drawing mode, choose from among the standard ones in the **Mode** popup menu, which displays two groups of modes. The first group consists of the standard modes, of which *srcCopy* and *srcOr* are the most common. The second group consists of special color modes for colored or multibit fonts.

Note: The various QuickDraw modes are documented in the QuickDraw chapter of *Inside Macintosh*, vol. I, pp. 156-7, and in the Color QuickDraw chapter of *Inside Macintosh*, vol. 5, pp. 57-61.

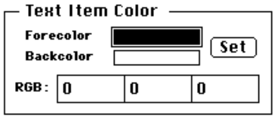
Special Note:

As of this writing, (System 6.0.7) the Dialog Manager has a bug in it such that it does not properly pick up every font name stored at the end of the 'ictb' resource when you have set the **Store Font Name** attribute bit. The symptom of this bug is that the font used to draw certain text items reverts to the system font. The font size, however, remains the same. For instance, if you have three Static Text items that are all set to be drawn in Geneva 9 bold, and you have set the **Store Font Name** bit for each item, only the first item will get drawn in Geneva 9 bold; the other two will be drawn in Chicago 9 bold (which usually looks horrible).

The Editor solves this problem by creating an *expanded* ‘ictb’ resource, as opposed to the standard compact form. Expanded ‘ictb‘s have explicit duplicated entries for each item in the dialog; compact ‘ictb’s allow sets of similar items to refer to and share the same table entry. You can configure the Editor to use one or the other of these methods, depending on whether Apple has fixed the bug in the latest system yet or not (for more on this, see the “Configuring Resorcerer” chapter). Whichever method is currently in effect, you can force the Editor to use the other method by holding the Option key down during those times that the Editor is compiling the ‘ictb’ resource. These times include when you close the editing session, when you are using the **Try Out...** command, and when you are duplicating an open dialog in the front window.

Setting Text Item Color

The **Text Item Color** panel contains two boxes that show you the foreground and background colors that you want the Dialog Manager to use when it displays the text items in your dialog.



To select one or the other color, click once on the box containing the color. It becomes the active color box; the active color box always has an extra rectangular frame drawn around it.

To change a color, select either the foreground or background color box and click the **Set** button. Or simply double-click in the box itself. The Mac’s standard Color Picker dialog lets you choose a new color.

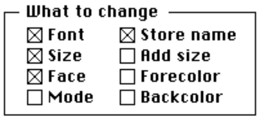
You can also directly view and edit the selected color’s components in the three fields labeled **RGB** along the bottom of the panel.

Note: When specifying a new text color, make sure to check the **Forecolor** or **Backcolor** bits in the **What to Change** panel. The Editor will remind you if you forget.

As with other color display boxes in Resorcerer Editors, once you have selected a color, you can **Copy** the color to the clipboard and then **Paste** it into any other color box.

Setting the What to Change bits

Once you have set a new text style or a new text color in the first two panels, you also need to indicate which of the various styles and colors you want the Dialog Manager to pay attention to, and which you want it to ignore when it draws the items.



What to change

☒ Font ☒ Store name

☒ Size □ Add size

☒ Face □ Forecolor

□ Mode □ Backcolor

Each of the checkboxes in the **What to Change** panel corresponds to a bit in the item color table entry that tells the Dialog Manager what to change (e.g., use the given forecolor instead of the default, use the given font and size instead of the default, etc.) before it draws the item in your dialog.

For text styles, the **Font**, **Size**, **Face**, and **Mode** bits need to be checked for the Dialog Manager to apply the respective changes.

The **Add size** bit indicates that the font size you've specified is to be added to whatever the default size being used is.

The **Store name** bit says that the name of the font you've chosen, rather than the internal font number, should be stored in the item color table resource. In general, you should keep this set, since font numbers are not usually the same on different Mac systems. However, to set a text style to the defualt application font, you will want to turn this bit off, and enter a font number of 1 into font value field to the right of the font popup menu.

For text colors to take effect, the **Forecolor** or **Backcolor** bits should be enabled.

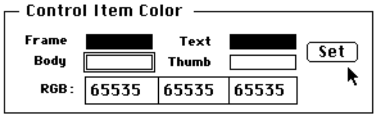
When you click on the **OK** button to apply your choices to all pertinent selected items in your dialog, the Editor checks the **What to Change** settings and checks them against the style and color information. If the changed style or color information is not standard, and a bit that applies to the changed style or color is not set, you are reminded in case you've forgotten to set it.

SETTING CONTROL COLORS

Standard dialog controls (that is, Buttons, Radio Buttons, and Checkboxes) are displayed in black and white, using the system font to draw any title text.

Note: You cannot change the text style of these standard controls (to do that you need to be using Control items), but you can change the colors used to display their various parts.

Select the controls whose colors you want to change and choose **Colors & Text Styles...** from the **Item** menu. The **Control Item Color** panel in the ensuing dialog lets you specify the colors you're interested in. If there are no standard control items in your selection this panel will be disabled.



RGB: 65535 65535 65535

The **Control Item Color** panel displays four boxes that contain the colors of the four standard parts of a control–the *frame*, *body*, *text*, and *thumb*–that you want the Dialog Manager to use when it displays the controls in your dialog.

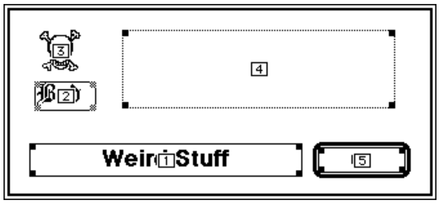
To select one of the colors, click once on the color box. It becomes the active color box, which has an extra rectangular frame drawn around it.

To change a color, select which of the four colors you want to affect and click the **Set** button. Or simply double-click in the box itself. The Mac's standard Color Picker dialog lets you choose a new color.

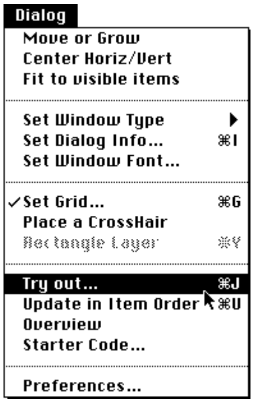
You can also directly view and edit the selected color's components in the three fields labeled **RGB** along the bottom of the panel.

As with other color display boxes in Resorcerer Editors, once you have selected a color you can **Copy** (using ⌘C) the color to the clipboard and then **Paste** it (using ⌘V) into any other color box. This makes it easy to conform colors without having to cut and paste individual color components.

TRYING A DIALOG OUT

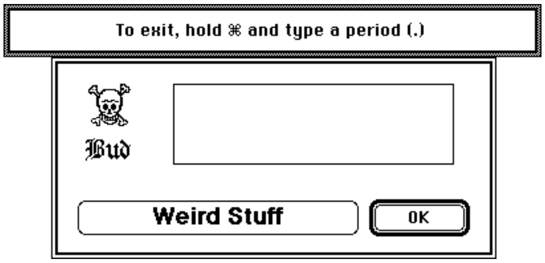


When you are editing a dialog and its related resources, you may want to get a feel for how the dialog will behave when you actually install it in your application. In particular, you may want to try out any custom controls or balloon help. There are also times when the Dialog Editor may not be faithfully displaying items the way the Dialog Manager will, particularly if Apple makes any changes to it in the future. The Editor lets you preview your dialog by actually running it through the Dialog Manager.

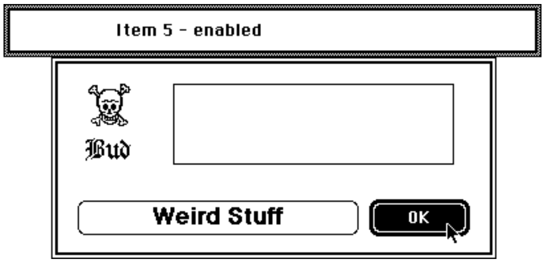


The **Try Out...** command in the Editor's menu asks the Editor to build a temporary resource file into which it places copies of all resources that represent your dialog, as well as all standard resources from your file to which items in the dialog refer (e.g. 'WDEF', 'PICT's, 'CNTL's, 'CDEF's, 'ICON's, 'MENU's, 'MDEF's, and Balloon Help resources). It then calls upon the Dialog Manager to build and display the dialog as a modal dialog. The Editor runs your dialog within a simple shell, complete with an event filter that implements as much of the generic behavior of the dialog as it can. This includes building working lists for your List User items; installing working pop-up menus; cutting from and pasting into Edit Text items; and grouping radio buttons.

Note: The shell that the Editor uses is much the same as what the Editor's starter code does for a modal dialog (for more on this, see the next section, "Starter Code for your Dialog".


To exit, hold ⌘ and type a period (.)

While running your dialog, the Editor displays a small window next to your dialog. This window initially contains a hint telling you how to get out of **Try Out** mode, but also gives you item number feedback as you click on various items in your dialog.



When you're through interacting with your dialog, hold the ⌘ key and type a period (.) to get back to editing. Or if the dialog has an **OK** or **Cance**l button already in it, just click on either of them. The Editor closes the temporary file and throws it away.

Before running your dialog, the Editor checks to make sure that all item numbers are in order and unique. If any are missing, duplicated, or invalid, the Editor will complain and then give you the choice of fixing the numbers yourself or automatically renumbering the list and continuing with the **Try Out...** command. For more on setting item numbers, see the "Dialog Item Numbers" section earlier in this chapter. If you do automatically renumber, you can undo the renumbering when you've finished trying the dialog out.

The Editor also checks for a variety of missing resource references, and lets you know about any. When running the dialog, items with missing references may be ignored and/or not displayed.

Note: Because **Try Out...** mode does not know about the values your application is going to set using the `ParamText` toolbox call, any text items with `ParamText` arguments are displayed using a '~' rather than a '^'.

If a Control item refers to a 'CNTL' template resource that in turn refers to a custom control definition function (a 'CDEF') that looks suspiciously small, the **Try Out...** command substitutes its own generic 'CDEF'. This is because the Dialog Manager may blindly assume that the 'CDEF' is complete, when in fact it is just a *stub* resource that your application changes at runtime to jump back into itself. The threshold size is the same as that specified in the **Report Definition Functions...** preference (for more on this, see "Editor preferences" later in this chapter).

Sorcery: If no text is selected in your dialog while you are trying it out, you can type ⌘C (for **Copy**) to take a screen snapshot of the dialog and place it in the clipboard. If you need selected text, use Shift-⌘P instead.

STARTER CODE FOR YOUR DIALOG

Once you have designed your dialog, choose **Starter Code...** from the **Dialog** menu to create a text file containing starter C source code that you can use to get your dialog up and running quickly. The code in the file is a completely self-contained module that operates the dialog for which it was created. It does not require any outside libraries to be linked in with it. This simple prototyping capability lets you get the syntax of operating the dialog out of the way fast, so that you can concentrate on the semantics of it by customizing the starter code. More importantly, because the Editor lets you specify which User items should represent Lists and which PopUp menus, the code generator can create all the proper code to deal with these items in the module it creates.

Note: You will probably want the text file created to have the same creator type as your development system uses. To set the default text file creator type, use the **File Preferences** section of the **Preferences...** command in the **Edit** menu.

Before creating the source code, the Editor asks you to specify whether you want source code that implements a modal or non-modal dialog. Code for modal dialogs relies on a filter function; whereas code for non-modal dialogs relies on a set of routines that can be called in response to events processed in your program's main event loop.

Most of Resorcerer's dialogs and the initial code to run them were prototyped (bootstrapped) using its own Dialog Editor and C code generator.

PRINTING YOUR DIALOG

When the dialog you're editing is the front-most window, choose **Print...** from the **File** menu to print a copy of the dialog window and its contents on your currently chosen printer.

If you have set **Show Item Numbers** or **Show Item Boxes**, then these will be printed also, exactly as they appear on your screen.

Note: You can also copy a picture of the dialog window to the Mac clipboard by first ensuring that no items are selected, and then choosing **Copy** from the **Edit** menu.

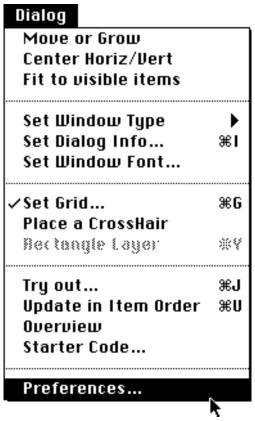
DECOMPILING A DIALOG

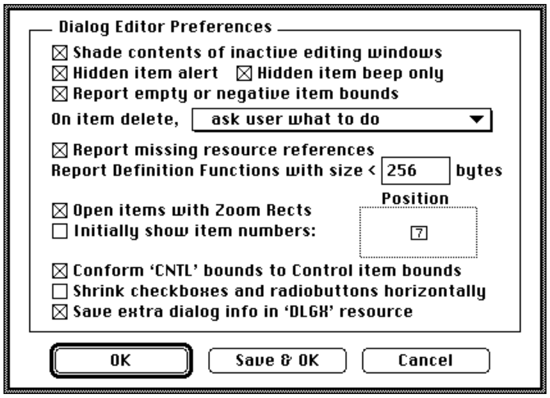
The Dialog Editor will decompile the resources that comprise a dialog directly to the clipboard, using the Rez language. To do so, make sure that all items in the item list are deselected (including hidden ones) by clicking somewhere in the dialog window background, or by choosing the **Select...None** command from the **Edit** menu. Then choose **Copy** from the **Edit** menu.

You can then **Paste** the text form of the resource set into the Rez text file with which you are compiling your resources. If you don't want the default 'dctb' or 'ictb' resource declarations, just delete them from your text file.

DIALOG EDITOR PREFERENCES

You can keep commonly used Dialog Editor settings as preferences to be set automatically every time you use the Editor. Choose **Preferences...** from the Editor's first menu. The settings are installed temporarily (that is, until you quit Resorcerer) when you click on the **OK** button, or permanently (so that they will be in effect the next time you run Resorcerer) when you click on the **Save & OK** button.





Dialog Editor Preferences

☒ Shade contents of inactive editing windows
☒ Hidden item alert ☒ Hidden item beep only
☒ Report empty or negative item bounds
On item delete, ask user what to do
☒ Report missing resource references
Report Definition Functions with size < 256 bytes
Position 7
☒ Open items with Zoom Rects
☐ Initially show item numbers:
☒ Conform 'CNTL' bounds to Control item bounds
☐ Shrink checkboxes and radiobuttons horizontally
☒ Save extra dialog info in 'DLGX' resource

Shade contents of inactive editing windows

It is usually a good idea to shade the contents of inactive dialog windows, due to the visual confusion that might otherwise result ("is it real or am I editing it?"). However, there are times when it is inappropriate to shade inactive windows: for instance, when you want to take a screen snapshot of more than one dialog at once, or when you want to compare two or more dialogs that are next to each other. Shading also causes lots more update activity, which on older Mac systems may take too much time for comfort.

We recommend that you leave shading on most of the time. Turn shading off when you have to, but then turn it back on again.

Hidden item alert

When **Hidden item alert** is checked, the Editor notifies you when you open a dialog that contains hidden items. The alert explains what to do to select and show these items, and is raised the first time the dialog window is brought to the front and activated.

If you check **Hidden item beep only**, the Editor will notify you by beeping without raising the alert.

We recommend that you keep Hidden item alert always set. Inexperienced users will want to keep **Hidden item beep only** unchecked; more experienced users will want to check it.

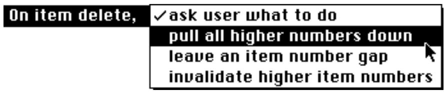
Report empty or negative item bounds

If you attempt to set the bounds of an item to a rectangle that is empty or has reversed (negative) coordinates, Resorcerer will let you know. However, under certain rare circumstances, you may really want items with empty bounding boxes. In these cases, you will probably want to suppress this warning.

We recommend that you keep this preference always set.

On item delete...

This popup menu lets you choose which of four possible item renumbering actions you want the Editor to take when you delete items. These actions are more fully explained in the “Editing Items” section of this chapter.



The most common setting for this preference is the **Pull higher numbers down** option.

Report missing resource references

With this preference set, the Editor will look up any icons, controls, pictures, or menus referred to by the data in any Icon, Control, Picture, or PopUp Menu User items. If it can’t find the referenced resource in the same file as your dialog, the Editor will alert you.

We recommend that you keep this preference set.

Report Definition Functions with size less than … bytes

This preference lets you set the threshold size, in bytes, at which to consider any ‘WDEF’ or other definition functions as non-stub functions. A stub definition function is a resource that does not have the complete compiled code of a definition function in it as expected. Instead, it has only a small amount of allocated storage in which the application, at runtime, places machine instructions to transfer control back into the application’s main code. Fortunately, stub functions are usually only a few bytes in length, whereas real functions are usually at least 1K in length, so a threshold value in the middle of these two extremes is usually pretty safe.

You can use this feature to force the Editor to flag all definition functions, simply by setting the threshold value to something very high, such as 32000. For instance, if you are finding that a custom ‘WDEF’ is misbehaving, setting the threshold to such a large value will cause the Editor to give you the opportunity to substitute the standard system ‘WDEF’ in its place while you’re editing the other parts of the dialog. You’ll have to open some other dialog first, though, to get to the Preferences command.

We recommend a value of 256 bytes as an arbitrary but safe threshold.

Use ZoomRects when opening items

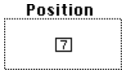
Zoom rectangles help identify which items belong to which open item information windows. This preference lets you specify whether or not to use them. On older Mac systems ZoomRects may be too slow for comfort; or you may just not like them.

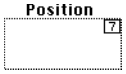
Initially show item numbers

When this preference is set, the Editor automatically turns **Show Item Numbers** mode on every time you open a dialog.

Position

The Editor draws item numbers according to the **Position** preference. With respect to each item’s bounding box, numbers can appear centered (the default), or in any corner, or centered along any side. To change the current alignment, click on the sample item number and drag it to any other position in the sample item bounding box.





Conform 'CNTL' bounds to Control item bounds

With this preference set, the Dialog Editor directly changes the 'CNTL' resource's bounding box to be the same as the bounding box of the Control item that refers to it. Whenever you make a change to the location or size of a Control item, only the 'CNTL' in the same file as your dialog is affected.

'CNTL' resources are *not* part of a Dialog's resource set as opened by the Dialog Editor. Consequently, if you discard any changes to the resource set after changing a Control item with this preference turned on, the 'CNTL' resource won't be reverted. However, if the 'CNTL' resource to be changed is open for editing, the Dialog Editor asks the Control Editor to make the change to the opened resource, and the Control Editor does allow you to revert any changes made.

When this preference is off, the **Use Best Size** command in the **Item** menu conforms a selected Control item's size to that of the 'CNTL' resource whose resource ID is given in the Control item's information.

We recommend that you always keep this preference checked.

Shrink checkboxes and radiobuttons horizontally

This preference affects the behavior of the **Item** menu's **Use Best Size** command only when two or more Checkbox or Radio button items are selected. Balloon help usually looks better when these items have bounding boxes that conform to the lengths of their text titles. However, on systems with a right-to-left script installed, it is usually better to use the same width for a group of related checkboxes or radiobuttons, so that the group aligns correctly on the right.

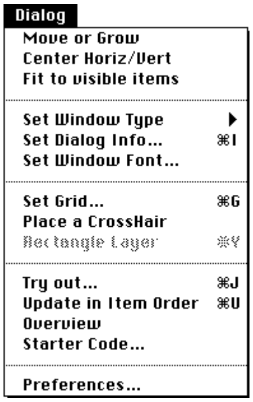
We recommend that you always leave this preference checked.

Keep extra dialog info in 'DLGX' resource

The Editor keeps information about the state of your dialog and its items in a separate companion resource of type 'DLGX' with an ID the same as your major resource.

This extra resource is completely transparent to the Dialog Manager, and is in no way required to run the dialog. If you do not want the Editor to create or maintain this extra resource, uncheck the **Keep extra dialog info in 'DLGX' resource**. We recommend that you always keep this checked.

DIALOG EDITOR MENU COMMANDS



The Dialog Editor installs two menus of commands to the right of Resorcerer’s **Resource** menu. The first of these menus is named according to the type of major resource you’ve asked to edit: **Dialog** (‘DLOG’), **Alert** (‘ALRT’), **Window** (‘WIND’), or **ControlPanel** (‘nrct’). Commands in the first menu apply to the dialog, alert, window, or control panel dialog as a whole.

The second menu, the **Item** menu, appears to the right of the first **Dialog**, **Alert**, or **ControlPanel** menu, but is absent when you are editing just a ‘WIND’ resource, since windows do not have associated dialog item lists. Commands in the **Item** menu are concerned only with the items in the dialog.

In addition, the Editor supports the printing of the front-most dialog when you choose **Print** from Resorcerer’s **File** menu. When no items in a dialog are selected, the **Edit** menu’s **Copy** command will copy a picture of the dialog window to the Mac clipboard.

The following is a short overview of each menu’s commands. You can find complete explanations in earlier sections of this chapter.

Commands from the **Dialog / Alert / Window / ControlPanel** menu are:

Move or Grow
Change the position or size of your dialog’s window, and the data representing it in its template resource. If the ⌘ key is down, don’t change the resource data. Holding the **Space Bar** down is a synonym for entering this mode.

Center Horiz/Vert
Center your dialog window on its screen first horizontally. If already centered horizontally, then center it using the 1/3 - 2/3 rule. Otherwise, center the window both horizontally and vertically.

Fit to visible items
Compute the bounding box of every visible or partially visible item, and change the dialog window’s size so that it encompasses these visible items, with distances taken from the difference between the

upper left corner of the window and the upper left corner of the bounding box.

Set Window Type / Set Alert Type
For dialogs and windows, choose the window type from among the standard types; for alerts, choose the alert icon to display.

Set Dialog Info / Set Alert Info / Set Window Info
Open the dialog, alert, or window template data, as well as the window colors for viewing and editing.

Set Window Font...
Choose a text style to associate with the editing window so that text items in the item list will appear in that style. This is in case you want to duplicate any similar text style change your application will be performing on this dialog window.

Set Grid...
Set the characteristics of any underlying grid to snap items to. If the command is checked the grid is active.

Place a Crosshair
Lets you click anywhere to place an alignment crosshair in the dialog's background. Double clicking on either hair aligns the selection. See also **Align with Crosshair** in the **Item** menu.

Rectangle Layer / Dialog Item Layer
Change the foreground editing layer to Control Panel rectangles or Control Panel dialog items. This does not apply to other types of dialog besides Control Panels dialogs.

Try out...
Collect copies of all resources in the current dialog's resource set, as well as referenced resources, and run them directly through the Mac's Dialog Manager. Any **OK** or **Cancel** buttons will dismiss the dialog, and ⌘ . (period) will always get you back to editing mode.

Update in Item Order
Draw all items from back to front in the order their item numbers specify. This is not always the same as the Dialog Manager would do, but is often close. See the **Try out...** command above.

Overview
Create a half-size overview of the dialog within standard screens.

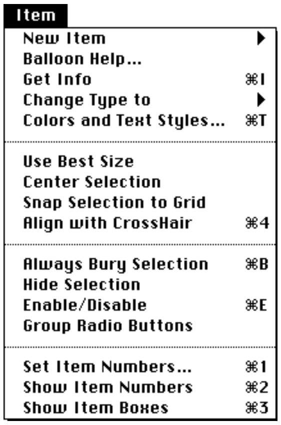
Starter Code

Generate a code module corresponding to the front-most dialog and its item list. Only for dialogs: not windows, alerts, or Control Panel dialogs.

Preferences...

Set various preferred settings either just for now, or for all times you use the Editor.

The **Item** menu commands are:



New Item

Create a new item, by specifying its characteristics first, or by choosing from among all item types in the hierarchical sub-menu.

Balloon Help

Set or delete the resource link information for any Help Items that attach System 7 on-line help to this dialog item list. If the item list has any appended Help Item, the command will be checked.

Get Info

Open the item data for editing in its own window.

Change Type To

Change the types of all selected items from what they are to the type chosen from the hierarchical sub-menu of types.

Colors and Text Styles

Conform all selected text and/or control items to a given text style and/or color.

Use Best Size

Conform the bounding box of all selected items.

Center Selection

Center either horizontally or vertically the bounding box of all selected items, and move those items to their respective positions.

Snap Selection to Grid
Force all selected items to be aligned with the current grid.

Align with Crosshair
Force the closest vertical edge of all selected items to conform with the position of the vertical crosshair. If Shift is down, move the items; otherwise, stretch them. Double-clicking on either vertical or horizontal part of the crosshair conforms vertically or horizontally.

Always Bury Selection / Unbury Selection
If any selected item is not marked for burial, mark all selected items as auto-buriable, and bury them. If all selected items are already marked, unmark them.

Hide Selection / Show Selection
Formally hide all unhidden selected items, unless all are already hidden in which case show them. Formally hiding an item entails adding 16384 to the X coordinates of its defining box. An informally hidden item is one whose position is just outside the dialog window bounds.

Enable / Disable
Enable all selected items, unless all are already enabled, in which case disable them.

Group / Ungroup Radio Buttons
Assign the next free group number to all selected Radio Button items if they are all ungrouped. If any belong to a group, ungroup them.

Set Item Numbers...
Set the item numbers of various classes of items, including all selected items, by invalidating them so that you can ⌘–click on them in the order you want.

Show Item Numbers
Turn Show Item Number mode on or off.

Show Item Boxes
Turn Show Item Boxes mode on or off.