
Providing Help Balloons for Menus

If your application uses the standard menu definition procedure, you will find that it is easier to provide help balloons for menus than for any of your other interface elements. This section is relatively lengthy compared to the sections describing dialog boxes, alert boxes, and windows only because it explains in greater detail much of the work you will also perform while providing help balloons for those elements.

This section assumes that your application uses the standard menu definition procedure. If your application uses its own menu definition procedure, use **Help Manager** routines to display and remove help balloons. See **Displaying and Removing Help Balloons** . Even if you use these routines, you should read this section so that your balloons emulate the behavior that the **Help Manager** provides for standard menus.

To create help balloons for a menu-pull-down, pop-up, or hierarchical-that uses the standard menu definition procedure, create a resource of type 'hmnv'. You can provide help balloons for the menu title and for each individual menu item.

The **Help Manager** can display different help balloons for the various states of a menu item. Each menu item can have up to four help balloons associated with it, one for each state:

- enabled
- disabled (that is, dimmed for the user)
- enabled and checked
- enabled and marked (that is, marked by a symbol other than a checkmark-for example, a bullet or a diamond)

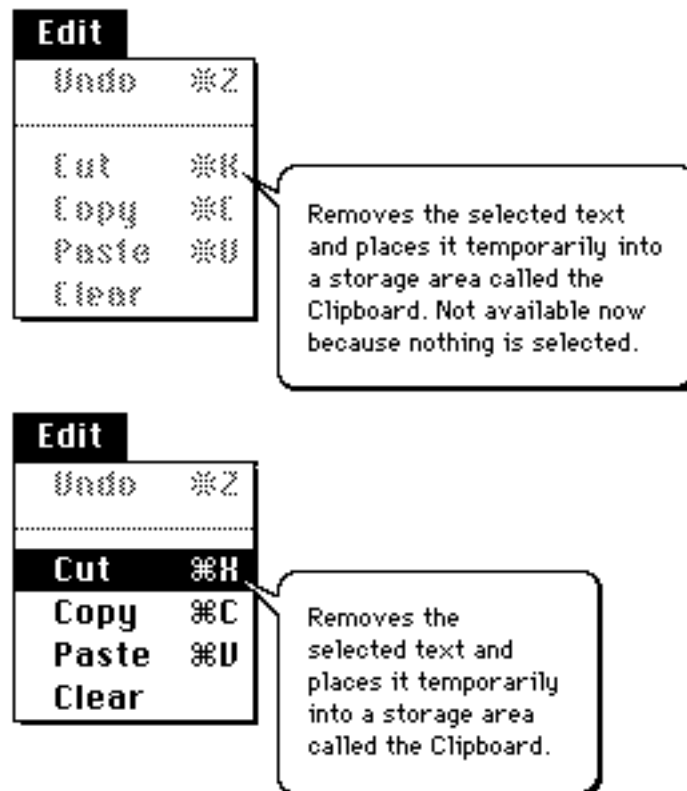
For example, you can define a help balloon that the **Help Manager** displays when the **Cut** command is enabled and another help balloon for display when the **Cut** command is dimmed. Remember that the help balloon that you provide for a dimmed menu item should explain why it is not currently available or, if more appropriate, how to make it available.

Note: Although *enabled* and *disabled* are the commands you use in a resource file to display or to dim menus and menu items, you should not use these terms in your help balloons or user guides. Rather, use the terms *menus*, *menu commands*, or *menu items* for those that are enabled, and use the terms *not available* or *dimmed* to distinguish those that have been disabled.

When your application calls the **MenuSelect** or **MenuKey** function, the **Menu Manager** tracks the cursor, highlights enabled menu items, and pulls down any additional hierarchical or pop-up menus as the user moves the mouse. As the user drags the cursor across or through a menu, the **Menu Manager** uses the **Help Manager** to display any

help balloons associated with the current state of the menu title or menu item.

The Figure below shows different help balloons for two instances of a menu, one with the **Cut** command dimmed, the other with the **Cut** command enabled.



Help balloons for different states of the **Cut** command

You do not specify hot rectangles or tip locations for menus. The rectangles defined by the **Menu Manager** for menu titles and menu items are used for the hot rectangles. The **Help Manager** initially tries to draw a help balloon for a menu item using variation code 0 (shown in the Figure below) with the tip placed eight pixels inside the right edge and halfway between the top and bottom edges of the menu item's rectangle. If the balloon's initial position lies wholly or partially offscreen, the **Help Manager** tries to redraw the balloon by moving its tip to the left edge of the item rectangle and using variation code 3. The **Help Manager** uses variation codes 1 and 2 in its attempts to draw help balloons for menu titles. The **Help Manager** never moves the tip for menu titles; instead, the tip is always located just below the bottom of the menu bar at the midpoint of the menu title's text.

The resource ID of each 'hmnu' resource should match the corresponding menu ID. For example, to provide help balloons for a menu with ID 130, create an 'hmnu' resource with resource ID 130.

The 'hmnu' resource contains four distinct components: the header, the help balloon content for any menu items missing from or

unspecified in the rest of the 'hmnu' resource, the help balloon content for the menu title, and the help balloon content for a variable number of menu items.

Here is the general format of an 'hmnu' resource.

Component	Menu help resource element
Header	<u>Help Manager</u> version Options Balloon definition function Variation code
Missing items	Identifier for help balloon content Balloon content for missing enabled items Balloon content for missing dimmed items Balloon content for missing enabled-and-checked items Balloon content for missing enabled-and-marked items
Menu title	Identifier for help balloon content Balloon content for enabled state of menu title Balloon content for dimmed state of menu title Balloon content when the menu title is dimmed by presence of a modal dialog box Balloon content for all menu items dimmed by presence of a modal dialog box
First menu item	Identifier for help balloon content Balloon content for enabled item Balloon content for dimmed item Balloon content for enabled-and-checked item Balloon content for enabled-and-marked item
Next menu item	(Same as for first menu item)
	. . .

The Listing below shows an example of part of a simplified menu help resource for an Edit menu.

A partial menu help resource

```
resource 'hmnu' (130, "Edit", purgeable) {
    // header information
    HelpMgrVersion,
    hmDefaultOptions,                // options
    0,                                // balloon definition function
    0,                                // variation code
    // missing items information
    HMSkipItem {
        // no missing items, so skip to menu title information
    }
```

```

    },
    { // Edit menu title's help balloon content
        HMStringItem { // use following pstrings
            // use string below when menu is enabled
            "Use this menu to manipulate text.",
            // use string below when menu is dimmed
            "Not available now because this file cannot be changed.",
            // use string below if modal dialog box is present
            "This menu's not available; respond to the dialog box.",
            // use string below if modal dialog box is present
            "This command's not available; respond to the dialog box.",
        },
        // Undo command's help balloon content
        HMStringItem { // use following pstrings
            // use string below when command is enabled
            "Undoes your last edit.",
            // use string below when menu is dimmed
            "Not available now; only editing changes can be undone.",
            // can't be checked, so empty string goes below
            "",
            // can't be marked, so empty string goes below
            "",
        },
        // dashed line between Undo and Cut commands
        HMSkipItem { // no help balloons for dashed lines
        },
        // Cut command's help balloon content
        HMStringItem { // use following pstrings
            // use string below when command is enabled
            "Cuts the selected text to the Clipboard.",
            // use string below when command is dimmed
            "Not available now because no text is selected.",
            // can't be checked, so empty string goes below
            "",
            // can't be marked, so empty string goes below
            "",
        }
        // Copy, Paste, and Clear commands' help balloons go here
    };

```

The header component comprises these fields:

- **Help Manager** version
- options
- balloon definition function
- variation code

In the header component, always specify the HelpMgrVersion constant (hmBalloonHelpVersion) in the first field.

In the options field, specify a constant (normally, hmDefaultOptions) or the sum of several constants' values from this list.

(**Specifying Options in Help Resources** describes these options.)

Note that the **Help Manager** never creates a window for a help balloon specified in an 'hmnu' resource. The **Help Manager** saves the bits behind the balloon when it creates the balloon. When it removes the balloon, the **Help Manager** restores the bits without generating an update event. You cannot specify options for drawing a window for a balloon in an 'hmnu' resource.

The balloon definition function field in the header specifies the resource ID of the window definition function that is used to draw the frame of the help balloon. To use the standard balloon definition function, specify a 0 for this field; this is the suggested default. If you use your own balloon definition (see **Writing Your Own Balloon Definition Function**), specify its resource ID in this field.

The variation code field in the header specifies the preferred position of the help balloon. For example, the standard balloon definition function displays help balloons according to eight different positions. If you specified the standard balloon definition in the preceding field, supply a variation code from 0 to 7 to display the balloon according to one of the eight positions shown in the third Figure in the section entitled **Help Balloon Display**. The preferred variation code is 0. If you are unsure of which variation code you should use, specify 0; the **Help Manager** will use a different variant if another is more appropriate. If you use your own balloon definition function, you specify its variation code in this field in the header.

After the header, you specify the format and content for help balloons for missing items, the menu title, and menu items.

Use the missing items component of this resource to specify how the **Help Manager** should handle menu items that are not described in the 'hmnu' resource. The missing field is also used for menu items that are described in the 'hmnu' resource but which lack help content for any states.

The missing items component of this resource is useful when you have menu items with similar characteristics or when the number of menu items is variable. For example, if the help information for a dimmed item applies to all dimmed menu items, you can specify the help information in the dimmed field of the missing items component instead of in the dimmed fields of the components for individual menu items.

For missing items (as in the rest of the items listed in an 'hmnu' resource), you store the help balloon content in text strings within this resource, or in separate 'STR ', 'STR#', 'PICT', or 'TEXT' and 'styl' resources. As described in "Specifying the Format for Help

Balloon Content," you will use these identifiers in the resource.

Identifier	Help content format
<u>HMStringItem</u>	Pascal string within the help resource
<u>HMPictItem</u>	'PICT' resource
<u>HMStringResItem</u>	'STR#' resource
<u>HMTEResItem</u>	'TEXT' and 'styl' resources
<u>HMSTRResItem</u>	'STR ' resource
<u>HMSkipItem</u>	No content-skip this item

Specify an identifier from the preceding list to describe the format of the help balloon content. Then, depending on the identifier you specified, in the next four fields supply either text strings for the balloon content or resource IDs of resources that contain the help balloon content.

The next four fields correspond to the following states of missing menu items:

- the balloon's content when a menu item is enabled, and either its content is not specified in the help resource (though the menu item is specified in the help resource) or the menu item itself is not specified in the help resource
- the balloon's content when a menu item is dimmed, and either it is missing from this resource or its help content is specified with either an empty string ("") or a resource ID of 0
- the balloon's content when a menu item is enabled and checked, and either it is missing from this resource or its help content is unspecified
- the balloon's content when a menu item is enabled and marked (with a character other than a checkmark), and either it is missing from this resource or its help content is unspecified

The content you supply is displayed in a help balloon for any menu item that appears at the end of the menu but is not described in this resource, and for any menu item in this resource whose content is specified as either "" (empty) for strings or 0 for resource IDs.

There are two additional identifiers that you can specify for menu items in 'hmnu' resources. These are explained later in this section.

Identifier	Purpose
<u>HMCompareItem</u>	Provide help only when string matches current menu item
<u>HMNamedResourceItem</u>	Get help content from the resource that has the same name as the current menu

item

The Listing below illustrates the help resource for a menu titled Colors. Notice in the missing items component that the field describing dimmed states for menu items has the message "Not available; either you have not selected text to color, or your monitor does not support color." Because this resource does not specify a message for any command's dimmed state, this message appears in help balloons for the Blue, Red, and Green commands whenever they're dimmed.

After the missing items component, you specify the help balloon content for the menu title and the menu items, in the same order in which they appear in the 'MENU' resource.

You can use the HMSkipItem identifier for items that appear in your menu but for which you do not provide a help balloon. For example, you can specify HMSkipItem for dashed lines that appear in menus. (Dashed lines cannot have help balloons.) If you specify HMSkipItem, you'll note that the **Help Manager** does not display help balloons for that menu item, even if the missing items component provides help information.

The missing items component in a menu help resource

```
resource 'hmnu' (132, "Colors", purgeable) {
    // header information
    HelpMgrVersion, hmDefaultOptions, 0, 0,
    // missing items information
    HMStringItem {
        "", // no missing enabled items
        // balloon content for all dimmed items below
        "Not available; either you have not selected text to color, "
        "or your monitor does not support color.",
        "", // no missing enabled-and-checked items
        "", // no missing enabled-and-marked items
    },
    { // Help for menu title and items
        // Colors menu title's help balloon content
        HMStringItem { // use following Pascal strings content
            // use string below when menu is enabled
            "Use this menu to display text in color.",
            // use string below when menu is dimmed
            "Not available because this monitor does not support color.",
            // use string below when modal dialog box is present
            "Use this menu to display text in color. Not available until "
            "you respond to the alert or dialog box.",
            // use string below when modal dialog box is present
            "Colors your selected text. Not available until you "
            "respond to the alert or dialog box.",
        },
        // Blue command's help balloon content
        HMStringItem { // use following Pascal strings for content
            // use string below when command is enabled
            "Displays the selected text in blue.",
            "", // use missing items content when menu is dimmed
```

```

        "", // command can't be checked, so use empty string here
        "", // command can't be marked, so use empty string here
    },
    // Green command's help balloon content
    HMStringItem {          // use following Pascal strings for content
        // use string below when command is enabled
        "Displays the selected text in green.",
        "", // use missing items content when menu is dimmed
        "", // command can't be checked, so use empty string here
        "", // command can't be marked, so use empty string here
    },
    // Red command's help balloon content
    HMStringItem {          // use following Pascal strings for content
        // use string below when command is enabled
        "Displays the selected text in red.",
        "", // use missing items content when menu is dimmed
        "", // command can't be checked, so use empty string here
        "", // command can't be marked, so use empty string here
    }
}
};

```

For the menu title component, as for the missing items component, you begin by specifying the identifier that describes the format for the help content. Depending on the identifier you specify, in the next four fields you supply either text strings for the balloon content or the resource IDs of resources that contain the help balloon content. These four fields correspond to these states of the menu title:

- the balloon content when the menu title is enabled
- the balloon content when the menu title is dimmed
- the balloon content for the dimmed title when a modal dialog box appears
- the balloon content for all dimmed menu items when a modal dialog box appears

The third and fourth fields of the menu title component specify help information for menus that are dimmed when an alert box or modal dialog box is the frontmost window. Users of system 7.0 can access selected menus in the menu bar while displaying an alert box or a modal dialog box. For example, the **Show Balloons (or Hide Balloons)** command is always available from the Help menu so that users can see your help balloons for the modal dialog box or alert box. While some menus are accessible, others may not be. The **Compatibility Guidelines** describe the circumstances under which menus are enabled or not when a modal dialog box is displayed.

With users having access to your menus whenever a modal dialog box or alert box is present, it is important to provide help that explains to users why your menus and items are dimmed by the dialog or alert box. Use the third field in the menu title component of an 'hmnu' resource to

specify the help balloon content that the **Help Manager** displays for that menu title when it is dimmed by the presence of a modal dialog box. For example, the Colors menu help resource shown in the Listing above displays this message to the user when a modal dialog box is present: "Use this menu to display text in color. Not available until you respond to the alert or dialog box."

Use the fourth field to specify the help balloon content for all menu items that become dimmed when a modal dialog box appears. For example, in the Colors menu example, this message is displayed to the user who selects the Blue, Green, or Red command when a modal dialog box is present: "Colors your selected text. Not available until you respond to the alert or dialog box."

After you provide the header, missing items, and menu title information, you specify the help content for each menu item. Each item in this resource must appear in the order that it appears in the corresponding menu. For each menu item, begin by specifying one of the previously listed identifiers to describe the format for the item's help balloons. Then, depending on the identifier you specify, supply either a text string for the balloon content or the resource ID of a resource that contains the help balloon content.

Each field for a menu item corresponds to one of the following states:

- the balloon's content when this menu item is enabled
- the balloon's content when this menu item is dimmed
- the balloon's content when this menu item is enabled and checked
- the balloon's content when this menu item is enabled and marked with a character other than a checkmark

Note that, for any item in the resource, you can specify only one format for the content of the states specified in the component. For example, if you specify the HMSTRResItem identifier for the **Undo** command, you must store the help content for all the states of the command in 'STR' resources. (However, if you specify a resource ID of 0 or an empty string as the content of any items in order to use the content from the missing items component of the resource, the content will follow the format specified in the missing items component.)

You do not have to provide help balloon content for every state of the menu item. If you do not provide help content for a particular state, the **Help Manager** uses the help information defined in the missing items component. If the missing items component does not provide help information for that menu state, then the **Help Manager** does not display a help balloon for that state of the item.

The Listing below shows a sample 'hmnu' resource for another **Edit** menu. (Although Listings in other sections illustrate menu help resources that contain their help balloon content in the form of Pascal strings within their own resources, you should keep your help balloon content in separate, more easily localized resources.) The 'hmnu' resource in the Listing below stores its help balloon content in a

separate 'STR#' resource (which is given a corresponding resource ID of 130 for easier maintenance).

Corresponding 'hmenu' and 'STR#' resources

```
resource 'hmenu' (130, "Edit menu help", purgeable) {
    HelpMgrVersion, 0, 0, 0, // standard header information
    HMSkipItem { // missing items information
        // no missing items, so skip to menu title information
    },
    { // menu title and items below
        // Edit menu title's help balloon content
        HMStringResItem { // use an 'STR#' resource for content
            130,1, // 'STR#' res ID, index when menu is enabled
            130,2, // 'STR#' res ID, index when menu is dimmed
            130,3, // 'STR#', index for title with modal dialog box
            130,4 // 'STR#', index for items with modal dialog box
        },
        // Undo command's help balloon content
        HMStringResItem { // use an 'STR#' resource for content
            130,5, // 'STR#' res ID, index when command is enabled
            130,6, // 'STR#' res ID, index when command is dimmed
            0,0, // command can't be checked
            0,0 // command can't be marked
        },
        // dashed item
        HMSkipItem { // no balloon help for dashed items
        },
        // Cut command's help balloon content
        HMStringResItem { // use an 'STR#' resource for content
            130,7, // 'STR#' res ID, index when command is enabled
            130,8, // 'STR#' res ID, index when command is dimmed
            0,0, // command can't be checked
            0,0 // command can't be marked
        },
        // content for Copy command's help balloons goes here
    }
};

resource 'STR#' (130, "Edit menu help strings") { // Edit help text
    { // array StringArray: 17 elements
        // [1] help text for enabled Edit menu title
        "Use this menu to undo your last action, to manipulate text, to "
        "select the entire content of a document, and to show what's "
        "on the Clipboard.";
        // [2] help text for dimmed Edit menu title
        "Use this menu to undo your last action, to manipulate text, to "
        "select the entire content of a document, and to show what's "
        "on the Clipboard. This menu is unavailable now.";
        // [3] help for dimmed Edit menu title with modal dialog present
        "Use this menu to undo your last action, to manipulate text, to "
        "select the entire content of a document, and to show what's "
        "on the Clipboard. This menu is unavailable until you respond to "
        "the alert box or dialog box.";
        // [4] help for dimmed Edit menu items with modal dialog present
        "This command is unavailable until you respond to the alert or "
```

```

        "dialog box.";
    // [5] help text for enabled Undo command
    "Undoes your last action; use this command to replace material "
    "you have cut or pasted, or to remove material you have pasted "
    "or typed.";
    // [6] help text for dimmed Undo command
    "Undoes your last action; use this command to replace material "
    "you have cut or cleared, or to remove material you have pasted "
    "or typed. Not available now because your last action did not "
    "involve cutting, clearing, pasting, or typing.";
    // help text for all other commands goes here
}
};

```

The 'hmnu' resource in the Listing above specifies the standard balloon definition function and variation code in the third and fourth fields of the header. The missing items component is specified using the HMSkipItem identifier, meaning that this 'hmnu' resource does not provide any help balloons for menu items that are missing from this resource or that do not have help content specified for any states.

Following the menu title, each menu item is listed in the order in which it appears in the menu. For items that do not specify information for a particular state, the **Help Manager** normally uses the information from the missing item. However, this 'hmnu' resource does not provide help content in the missing item component. Instead, all help content is specified with each menu item in this resource. Because there are no enabled-and-checked or enabled-and-marked states for the **Undo** and **Copy** commands, these states are specified with resource IDs of 0.

If you have a menu item that changes names, you can use the HMCompareItem identifier to compare a string against the current menu item in that position. If the string specified after the HMCompareItem identifier matches the name of the current menu item, the **Help Manager** displays the help balloon content specified in the next four fields of the **Help** menu resource. Because of performance considerations, the HMCompareItem identifier should not be used unless necessary.

Here is the general format for specifying help within an 'hmnu' resource for a changing menu item.

Component	Menu help resource element
Changing menu item	<u>HMCompareItem</u> identifier String to compare against current menu item Identifier for help balloon content Balloon content for enabled item Balloon content for dimmed item Balloon content for enabled-and-checked item Balloon content for enabled-and-marked item

In the 'hmnu' resource, create components that use the HMCompareItem identifier for every name that can appear in that menu position. For example, the Listing below shows an 'hmnu' resource for a menu command that toggles between **Show Colors** and **Hide Colors**.

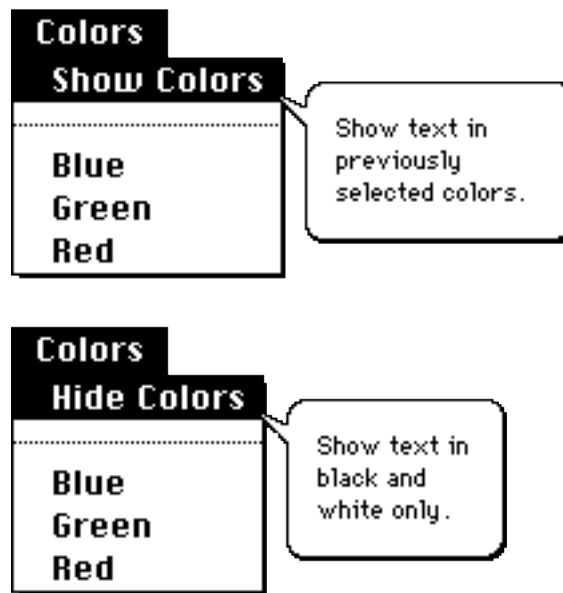
Using HMCompareItem for a changing menu item

```
resource 'hmnu' (132, "Colors menu help", purgeable) {
{
    HMCompareItem { // help content if first command is Show Colors
        "Show Colors",
        HMStringResItem {
            132, 1,                // enabled
            0, 0,                  // use missing items
            0, 0,                  // item can't be checked
            0, 0                    // no marked state
        },
    },
    HMCompareItem { // help if the first command is Hide Colors
        "Hide Colors",
        HMStringResItem {
            132, 2,                // enabled
            0, 0,                  // use missing items
            0, 0,                  // item can't be checked
            0, 0                    // no marked state
        },
    },
    // Blue command's help balloon content
    HMStringItem {                // use following Pascal strings for content
        // use string below when command is enabled
        "Displays the selected text in blue.",
        "", // use missing items content when menu is dimmed
        "", // command can't be checked, so use empty string here
        "", // command can't be marked, so use empty string here
    },
}
};

resource 'STR#' (132, "Hide & Show Colors commands help text") {
{
    // [1] help text for enabled Show command
    "Show text in previously selected colors.";
    // [2] help text for enabled Hide command
    "Show text in black and white only.";
}
};
```

As illustrated in the Figure below, when the menu command is **Show Colors**, the Help Manager displays the help balloon content described by the first HMCompareItem component. When the menu command is **Hide Colors**, the Help Manager displays the help

balloon content described by the second `HMCompareItem` identifier.



Help balloons for a changing menu item

You can also specify the content of a help balloon with the `HMNamedResourceItem` identifier, which causes the **Help Manager** to use a resource whose name matches the current name and state of the menu item. After the `HMNamedResourceItem` identifier, you specify the resource type ('STR ', 'STR#', 'PICT', or, for text, 'TEXT'), and the **Help Manager** uses the `GetNamedResource` function to find the resource with same name as the current menu item. (If you specify 'TEXT', you also get style information for the 'TEXT' resource by creating a 'styl' resource with the same name.)

If the menu item is dimmed, the **Help Manager** appends an exclamation mark to the menu item string and searches for a resource by that name. If the menu item is enabled and marked with a checkmark or other mark, the **Help Manager** appends the mark to the menu item string and looks for a resource with that name.

For example, this 'hmnu' resource specifies that the **Help Manager** extracts help content from a resource named Red of type 'STR ' when displaying a help balloon for an enabled menu command named Red. If the menu item is dimmed, the **Help Manager** gets the 'STR ' resource with the name Red! and uses its text string for the balloon content. If the Red command could be marked with an asterisk (*), the **Help Manager** would search for the resource with the name Red* of type 'STR '.

```
resource 'hmnu' (132, "Colors menu help", purgeable) {
    // see the Listing above for header info, missing items help, menu
    // title help, other menu items help
    {
        HMNamedResourceItem { // Red command's help balloon content
            'STR ' // use the 'STR ' resource named "Red"
        }
    }
}
```

```
    }  
};  
resource 'STR ' (333, "Red") {      // help text for enabled Red command  
    "Displays the selected text in red."  
};  
resource 'STR ' (334, "Red!") {      // help text for dimmed Red command  
    "Not available; either you have not selected text to color, "  
    "or your monitor does not support color.",  
};
```

If there is sufficient memory, the standard menu definition procedure saves the bits behind the help balloon and restores these bits for quick updating of the screen. If there is not sufficient memory to save the bits behind the help balloon, then-as with menus-the procedure generates appropriate update events.