
Help Balloons in Static Windows

To provide help balloons for the static windows of your application without modifying its code, create a resource of type 'hwin' (window help) and additional resources of type 'hrct' (rectangle help) or 'hdlg' (dialog item help). If your static windows provide help balloons with these resources, the **Help Manager** automatically tracks the cursor and displays and removes help balloons as the cursor moves into and out of the hot rectangles associated with these resources.

The 'hwin' resource allows you to associate 'hrct' and 'hdlg' resources with your static windows. You use the 'hrct' and 'hdlg' resources to define help balloons for the individual elements within your windows. While the **Help Manager** uses the item rectangles defined in the 'DITL' resource as the hot rectangles for 'hdlg' resources, you can specify your own hot rectangles for dialog boxes and other static windows when using 'hrct' resources.

An 'hrct' resource specifies tip locations, hot rectangles, balloon definition functions, variation codes, and help content for areas within a static window.

Providing Help Balloons for Items in Dialog and Alert Boxes

explains how the 'hdlg' resource describes the tip, alternate rectangle, and help information for dialog items in a dialog box. That section also describes how to associate the 'hdlg' resource with a dialog box by adding an item of type helpItem to the dialog box's 'DITL' resource. This section describes how you can instead treat your dialog boxes or alert boxes as static windows and use an 'hwin' resource instead of helpItem items to associate them with 'hdlg' resources.

The 'hwin' resource identifies windows by their titles or by their windowKind values. You can list all of your windows within one 'hwin' resource, or you can create separate 'hwin' resources for your separate windows. (You will probably find that it is easier to maintain your window help if you create only one 'hwin' resource, but, as described later in this section, you must create separate 'hwin' resources for windows that require different options—for example, one window matched to its 'hwin' resource by a string anywhere in the window's title and another window matched to its 'hwin' resource only by the exact string of the window's title.) An 'hwin' resource contains the resource ID (or IDs) of one or more 'hrct' and 'hdlg' resources. An 'hwin' resource can use 'hrct' and 'hdlg' resources for various parts of the same window.

To use an 'hwin' resource, the window must have either a title or a windowKind value in its window record that adequately distinguishes it from other windows. Within an 'hwin' resource, you could identify the Verb Tenses window shown in the Figure above by its title, and you could identify the scrolling palette window in the Figure above by its windowKind value.

The **Window Manager** describes the windowKind field of the window record. Note that windowKind values of 0, 1, and 3 through 7 are

reserved by the Operating System and that dialog boxes or alert boxes must have a value of 2. Because your dialog boxes and alert boxes must have a windowKind value of 2, you can use this value to define only one 'hwin' resource for all untitled dialog boxes and alert boxes. You may find it difficult-using help resources alone-to provide help balloons for untitled dialog and alert boxes. However, you can use an 'hwin' resource to associate generic help for the common elements of all your untitled dialog boxes and alert boxes, and you can use the **HMSetDialogResID** function to provide help for the unique elements among them.

Getting and Setting Information for Help Resources explains the **HMSetDialogResID** function.

You describe the tip, a rectangle, and help information for each object in static windows using either 'hrct' or 'hdlg' resources. Shown here is the general format of an 'hrct' resource. 'Hdlg' resources are described in,

Providing Help Balloons for Items in Dialog and Alert Boxes.

Component	Rectangle help resource element
Header	Help Manager version Options Balloon definition function Variation code
First hot rectangle	Identifier for help balloon content Tip location Hot rectangle coordinates Balloon content for hot rectangle
Next hot rectangle	(Same as for first hot rectangle) . . .

The 'hrct' resource contains a header and information that describes the hot rectangles within the window and the help balloons associated with each hot rectangle.

As with the other help resources, specify the HelpMgrVersion constant in the first field of the header component of the 'hrct' resource. In the options field, specify a constant (normally, hmDefaultOptions) or the sum of several constants' values. (**Specifying Options in Help Resources** describes these options.)

Specify the balloon definition function and variation code (both typically 0) in the third and fourth fields of the header. (These are described in detail in **Providing Help Balloons for Menus.**)

Following the header, you specify tip locations, hot rectangles, and associated help content for all the areas in the window that would benefit by having help balloons.

For each hot rectangle, specify the format that the help content takes. As with the other help resources, specify the format using one of these

identifiers:

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

After specifying the format of the help content, give the points of the tip locations and the hot rectangles in local coordinates. Then provide your help content for that rectangle, as either a text string or a resource ID.

Specify the format, tip, hot rectangle, and help content for every applicable area in the window. You will create an 'hwin' resource that contains the resource ID of this 'hrct' resource and that associates this resource with a titled window. The **Help Manager** automatically tracks the cursor and displays and removes help balloons as the user moves the cursor into and out of the hot rectangles defined in this resource.

If you need to supply a help balloon for an area within another area that needs a different help balloon, create 'hrct' resources for both the inner and outer areas and specify their areas as hot rectangles. In the resource fork of your file, list the 'hrct' resource for the inner area ahead of the 'hrct' resource for the outer area. Then, with the cursor within the inner hot rectangle, the **Help Manager** scans its 'hrct' resource first and displays its help balloon instead of the help balloon for the outer hot rectangle. When the cursor moves from the inner hot rectangle to the outer, the **Help Manager** removes the inner area's help balloon and instead displays the balloon for the outer hot rectangle.

As previously explained, you can create an 'hdlg' resource to specify the tips, alternate rectangles, balloon definitions, variation codes, and help content for dialog items and use an 'hwin' resource to associate that 'hdlg' with a dialog box or alert box. When help is enabled and your application calls **ModalDialog** or **IsDialogEvent**, the **Help Manager** automatically tracks the cursor and displays and removes help balloons for items specified in the 'hdlg' resource.

To associate 'hrct' and 'hdlg' resources with windows, create an 'hwin' resource. Shown here is the general format of an 'hwin' resource.

Component	Window help resource element
Header	Help Manager version Options

First window resource	Resource ID of associated 'hrc't' or 'hdlg'
flagged by a	Resource type ('hdlg' or 'hrc't')
untitled	Length used to compare title strings-or, if minus sign (-), the <u>windowKind</u> value of an window
untitled	Window title string-or empty string if
Next window	(Same as for first window)
	.
	.
	.

The 'hwin' resource maps windows to 'hrc't' resources and 'hdlg' resources. Within the 'hwin', you identify 'hrc't' resources and 'hdlg' resources by their resource IDs and by their types. As shown in the preceding list, the first two fields of a window component contain the resource ID and type. You identify windows in one of these two ways:

- by specifying the number of characters used for matching a window title in the third field, and by specifying a string containing this number of sequential characters from the window's title in the next field
- by flagging the third field of the component with a minus sign (-) and specifying the windowKind value from the window's window record in this field; leave an empty string in the next field

When an active window has a title or windowKind value that matches an 'hwin' resource, the **Help Manager** provides help balloons for the hot rectangles associated with the specified 'hrc't' or 'hdlg' resources.

The 'hwin' resource specifies the **Help Manager** version and options in the header. Specify the HelpMgrVersion constant in the first field of the header. In the options field, specify a constant (normally, hmDefaultOptions) or the sum of several constants' values. Notice that compared to the other resources related to the **Help Manager**, options regarding local coordinates and bits behind the balloon are not applicable to the 'hwin' resource, but the 'hwin' resource has a unique option-hmMatchInTitle.

If you're providing help balloons for a desk accessory or a driver that owns other resources, use the hmUseSubID constant in the options field. (See the **Resource Manager** for a discussion of owned resources and their resource IDs.)

You can specify the hmMatchInTitle constant to match windows containing a specified number of sequential characters starting with any character position in the window title. If you do not specify the hmMatchInTitle constant in the options field, the **Help Manager** matches characters starting with the first character of the window

title.

For example, if the 'hwin' resource specifies the hmMatchInTitle constant, 4 as the number of characters to match, and "Test" as the window title, the **Help Manager** uses this 'hwin' resource when the cursor is located in any active window that is titled Test, Window Test, or Test Case or is given a title with any other string that contains the characters "Test."

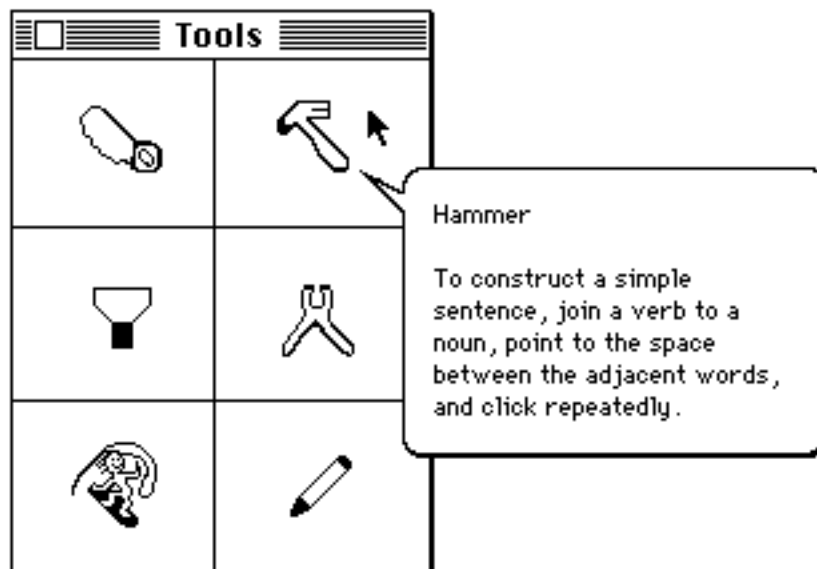
If you supply the hmDefaultOptions constant, the **Help Manager** treats the resource IDs in this resource as regular resource IDs and not as subrange IDs, and it begins matching characters at the first character of the window string. As long as they all use the same options, you can list help for all your windows in a single 'hwin' resource. You must create separate 'hwin' resources for windows that require different options.

Following the header, multiple 'hdlg' or 'hrct' resources can be specified. You specify the window title and the resource ID for each 'hdlg' or 'hrct' resource. You also specify how many characters of the window title the **Help Manager** should use when matching the window with the 'hdlg' resource.

For dialog boxes and alert boxes, you can use 'hrct' resources to define hot rectangles in addition to or other than those associated with the dialog items. For example, you might want to use an 'hwin' and an 'hrct' resource in a dialog box to associate a single help balloon with a group of related dialog items rather than provide separate help balloons for all the individual dialog items. (To create help balloons for single dialog items by using 'hdlg' resources alone, see the information in the separate section,

Providing Help Balloons for Items in Dialog and Alert Boxes

The Figure below shows a sample palette an application might use and the help balloon displayed for the hammer tool.



A tool palette with a help balloon

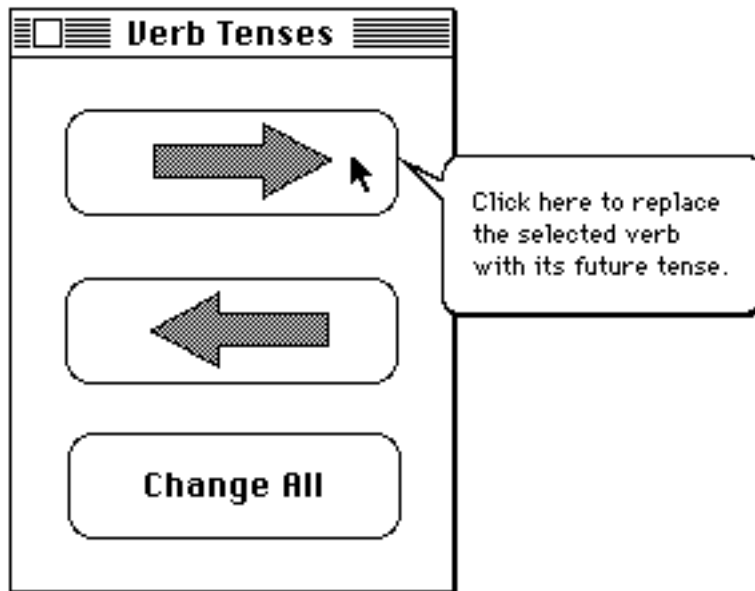
The Listing below shows the window help resource and the hot rectangle help resource for the palette in the Figure above.

Corresponding 'hwin' and 'hrct' resources for a tool palette

```
resource 'hwin' (128, "Window help resource", purgeable) {
    HelpMgrVersion,
    hmDefaultOptions,
    {
        128,          // resource ID of type specified on next line
        'hrct', // resource type for defining help
        5,           // length to use when comparing strings
        "Tools" // window's title string
    }
};
resource 'hrct' (128, "Tools palette help") {
    HelpMgrVersion,
    hmDefaultOptions,
    0,                      // balloon definition function
    0,                      // variation code
    {
        // definition for saw tool goes here
        HMStringResItem {          // definition for hammer tool
            {50, 127},              // tip
            {22,99,54,131}, // hot rectangle
            147,2                  // 'STR#' resource ID and index
        }
        // definitions for other tools go here
    }
};

resource 'STR#' (147, "Tools palette help text") {
    {
        // [1] saw tool
        // help text for saw tool goes here
        // [2] hammer tool
        "Hammer \n\nTo construct a simple sentence, join "
            "a verb to a noun, point to the space between the adjacent words, "
            "and click repeatedly.";
        // help for other tools goes here
    }
};
```

You can also use the 'hwin' resource to associate help for dialog items in a window. The Figure below shows the **Help Manager** displaying a help balloon for an item in the dialog box titled **Verb Tenses**.



A help balloon for a dialog box with a title

The Listing below shows how the 'hwin' resource associates an 'hdlg' resource with the dialog box illustrated in the Figure above. This 'hwin' resource associates help for three different windows: the first is the window titled Tools, the second is an untitled window with a windowKind value of 10, and the third is the dialog box titled **Verb Tenses**.

Specifying help for titled and untitled windows with an 'hwin' resource

```
resource 'hwin' (128, "Window help resource", purgeable) {
    HelpMgrVersion,
    hmDefaultOptions,
    {
        128,          // help resource ID for Tools window
        'hrc't',      // resource type for defining help
        5,            // length to use when comparing strings
        "Tools",      // window's title string

        129,          // help res ID for untitled window
        'hdlg',       // resource type for defining help
        -10,          // match on windowKind values of 10
        "",           // matching on windowKind, so empty string goes here

        130,          // help res ID for Verb Tenses window
        'hdlg',       // resource type for defining help
        11,           // length to use when comparing strings
        "Verb Tenses", // dialog box's title string
    }
};

resource 'hdlg' (130, "Help for Verb Tense control", purgeable) {
    HelpMgrVersion, // version of Help Manager
```

```

0, // start with first dialog item in 'DITL'
hmDefaultOptions, // options
0, // balloon definition ID
0, // variation code
HMSkipItem { // no missing dialog item help information
},
{ // help information for dialog items
    HMStringResItem {
        {20, 130}, // tip (local to item rect)
        {0,0,0,0}, // default alternate rectangle: use item rect
        131, 1, // highlighted control for future tense
        131, 2, // dimmed control for future tense
        0, 0, // no checked state for control
        0, 0 // no other states for control
    },
    HMStringResItem {
        {20, 130}, // tip (local to item rect)
        {0,0,0,0}, // default alternate rectangle: use item rect
        131, 3, // highlighted control for past tense
        131, 4, // dimmed control for past tense
        0, 0, // no enabled and checked control
        0, 0 // no other marks for control
    },
    HMStringResItem {
        {20, 130}, // tip (local to item rect)
        {0,0,0,0}, // default alternate rectangle: use item rect
        131, 5, // Change All button
        0, 0, // no dimmed Change All button
        0, 0, // no checked state Change All button
        0, 0 // no other states for Change All button
    }
}
};

resource 'STR#' (131, "Verb tense help strings") {
{
// [1] highlighted control for future tense: help text
"Click here to replace the selected verb with its future tense.";
// [2] dimmed control for future tense: help text
"Click here to replace a verb with its future tense. "
    "Not available now because you have not selected a verb.";
// [3] // highlighted control for past tense: help text
"Click here to replace the selected verb with its past tense.";
// [4] dimmed control for past tense: help text
"Click here to replace a verb with its past tense. "
    "Not available now because you have not selected a verb.";
// [5] Change All button's help text
"Click here to repeat your last change for all verbs within "
    "your document.";
}
};

```