New Technical Notes

Macintosh

Developer Support

Help Manager Q&As

Toolbox M.TB.HelpMgr.Q&As

Revised by: Developer Support Center October 1992
Written by: Developer Support Center October 1990

This Technical Note contains a collection of Q&As relating to a specific topic—questions you've sent the Developer Support Center (DSC) along with answers from the DSC engineers. While DSC engineers have checked the Q&A content for accuracy, the Q&A Technical Notes don't have the editing and organization of other Technical Notes. The Q&A function is to get new technical information and updates to you quickly, saving the polish for when the information migrates into reference manuals.

Q&As are now included with Technical Notes to make access to technical updates easier for you. If you have comments or suggestions about Q&A content or distribution, please let us know by sending an AppleLink to DEVFEEDBACK. Apple Partners may send technical questions about Q&A content to DEVSUPPORT for resolution.

New Q&As and Q&As revised this month are marked with a bar in the side margin.

No pre-System 7 Help Manager

Written: 7/12/90
Last reviewed: 8/30/91

Contrary to rumors you might have heard, the Help Manager (including balloon help package) will not be available for pre-7.0 systems.

Help Manager menu command-key bug and workaround

Written: 6/4/91
Last reviewed: 8/13/91

I cannot get command-key equivalents to work on the System 7 Help menu. Under System 6 the first item's command-? equivalent in my Help menu works, but under System 7 the MenuKey seems to return 0.

—

What you are experiencing is a bug in System 7. The bug only effects menu keys and the Help menu, so the workaround to it is really quite simple. Before you call menu key, test the key to see if it is a question mark (in other words, perform the menu key function yourself in this instance). Some Pascal code that might do this (modified from the MPW sample.p application)

follows. Inside your case statement in your event loop, the keystroke event case would look something like this:

```
    keyDown, autoKey: BEGIN                {check for menukey equivalents}
        key := CHR(BAnd(event.message, charCodeMask));
        IF BAnd(event.modifiers, cmdKey) <> 0 THEN    {Command key down}
            IF event.what = keyDown THEN BEGIN
                AdjustMenus;    {enable/disable/check menu items properly}
If key<>'?' then           { test for our special help menu keystroke }
                  DoMenuCommand(MenuKey(key)) {if not,
                                               just handle normally}
Else begin
  HiLiteMenu(kHMHelpMenuID);
  DoMenuCommand(($10000*kHMHelpMenuID)+myHMItemNumber);
end;
            END;
```

The workaround will not cause any problems if the bug's fixed in the future, so it is safe to include in your software.

Menu help balloons with application extensions

Written: 6/19/91
Last reviewed: 10/15/91

How can my Macintosh application extension change the help balloons for menus it has modified in the application?

—

You can head patch (DON'T tail patch) HMShowMenuBalloon to call the Help Manager's HMSetMenuResID routine, which maps in an alternate menu help resource ('hmnu'). You create a copy of the application's 'hmnu' resource, modify the copy and map it in using HMSetMenuResID—something like this:

```
HMSetMenuResID(mEdit, 256);
```

When you remove your changes to the menu, unmap your 'hmnu' help menu resource with another call to HMSetMenuResID, as follows:

```
HMSetMenuResID(mEdit, -1);
```

BalloonWriter 1.0f3 rez input file bug & workaround

Written: 7/5/91
Last reviewed: 8/30/91

A rez input file generated by BalloonWriter 1.0f3 is not the same as the MPW 3.2 definitions in BalloonTypes.r. Until this is fixed, the only solution is to go in by hand and fix the resources yourself.

Balloon Help and modeless dialogs

Written: 8/9/91

Last reviewed: 9/16/91

I have a problem with Balloon Help for modeless dialogs. The balloons don't show up most of the time unless the user clicks the mouse over the dialog item of interest.

—

The problem you are having is that you are not calling isDialogEvent with nul events. The most likely cause for this is that you are not getting nul events, or you have your sleep time way too high in wait next event. If you are not getting nul events, it is probably due to unserviced update events in the event queue. If you can not ensure that you will get null events back from waitNextEvent, you must fudge them instead. This way the text edit insertion point will blink properly as well.

Balloon help for menus with submenus
Written: 8/19/91
Last reviewed: 10/8/91

We defined balloon help for our menus and submenus, but if we click on the main menu item, the balloon for this item only appears in a flash and then disappears. The help for the hierarchical items work, but if we drag on the main item again, no help appears.

—

When submenus are present, the help item for the parent menu item is not supposed to show for three reasons: First, the menu item is the title of the submenu and you do not see the help for a menu title when the menu is pulled down. (This is what actually causes the help balloon not to show. The system is treating the menu item as a menu title, with the side effect that you can never position the cursor over the item without popping down the submenu, so the item's associated help will never show.) Second, Apple's human interface group felt that there would be little need for menu help once the submenu has appeared (envisioning a balloon that says something like "The submenu attached to this menu contains items that..." which would be redundant). The final reason is that if the balloon came up it would obscure the submenu that just popped up, forcing users to move the cursor to the submenu before they could see what choices are available. This looks bad and works bad.

Help for nested dialogs
Written: 8/19/91
Last reviewed: 10/15/91

We define our helps through 'hdlg' and the call HMSetDialogResID Does this call support nested dialogs?

—

The HMSetDialogResID is a global function. It acts on all dialogs that are currently up in

addition to dialogs that are to come up (contrary to what the manual implies). So, to handle help for nested dialogs you should do the following:

Always associate an 'hdlg' with a dialog via the 'DITL' help item whenever possible, thus eliminating the need to use HMSetDialogResID.

Before bringing up any dialog that you must use HMSetDialogResID with, you should first call hmGetDialogResID to preserve the current state. When the dialog in question goes away or is de-activated (in the case of a modeless dialog), you should reset the ID to what it was before the dialog was put up. Also, if you are using this with modeless dialogs, be sure to again call hmSetDialogResID at activation.

Help menus with two menu bars
Written: 9/11/91
Last reviewed: 10/15/91

I'm appending items to the system help menu but my program uses two menu bars and when I call SetMenuBar the items that were added to the help menu are lost. What can I do to keep from having to rebuild the help menu each time I do a SetMenuBar? I don't fully understand the ownership of the help menu (each application gets its own?).

—

Well, if you must use two menu bars, then you will always have to re-add your help menu items after a call to setMenuBar. Of course, the reason for this is that the help menu, process menu, and keyboard menu are not considered part of your menu list. The system thinks it is doing you a favor by isolating these menus from you (which in the case of the process menu makes quite a good bit of sense). Unfortunatly, you have found the one big flaw in the whole plan. The solution for you is to write a routine that calls setMenuBar and re-adds the help menu items all at the same time.

Balloon help for application windows
Written: 9/11/92
Last reviewed: 10/15/91

If I program the balloon help myself for my windows, does this mean I must program for the menus also, or can I use the resource method?

—

Each different type of balloon help—dialog, menu, and application—is independent of the others. Before the Menu Manager puts up any balloons it will remove the current balloon. You should do the same (see information and warning on page 11-71 of Inside Macintosh Volume VI). Bottom line, you can use the resource method for menu and dialog help (dialogs that use the Dialog Manager), and still manually provide help for your application windows.

Displaying button help strings for Chooser device package
Written: 10/28/91
Last reviewed: 2/17/92

How do I put up balloons for buttons in the Chooser's window? The Chooser automatically inserts the buttons for me and gives me no method of appending anything to its dialog before it is displayed.

—

To get your own help strings displayed for any buttons or radio buttons displayed for your Chooser device package, simply add a ‘STR#’ resource with ID -5694 to your device resource file. The format of the resource is described below:

```
#define ChooserResID    -15904
#define PackResID        -5694            // this is in the _Pack14 range

/*    Any Chooser package can get balloons on all items that are normally
     added to the Chooser dialog by containing a STR# resource of
     resource ID = PackResID (-5694) and following the convention below: */

     resource 'STR#' (PackResID, purgeable) {
    {   "kEnabledLeftButton",   // message for button (or control) that is
                                // Enabled but not checked
       "kDisabledLeftButton",   // message for the control that is Disabled
       "kCheckedLeftButton",    // message for the control that is Checked
                                // (i.e., CtrlValue>0)
       "kOtherLeftButton",      // message for the control that is Other
                                // (i.e., CtrlValue>1)
       "kEnabledRightButton",
       "kDisabledRightButton",
       "kCheckedRightButton",
       "kOtherRightButton",
       "kEnabledOnButton",
       "kDisabledOnButton",
       "kCheckedOnButton",
       "kOtherOnButton",
       "kEnabledOffButton",
       "kDisabledOffButton",
       "kCheckedOffButton",
       "kOtherOffButton",
       "kEnabledOnOffTitle",
       "kDisabledOnOffTitle",
       "kCheckedOnOffTitle",
       "kOtherOnOffTitle",
    }
   };
```

You were also asking about overriding the default help when the cursor is over your device list in the Chooser. To do this would require a lot of effort on your part. It would involve getting the ‘hdlg’ help resource used by the Chooser (probably with HMGetDialogResID), stuffing your help string in place of the existing string for the device list (the device list is item 6 in the dialog list), and then putting back the original string once another device is selected. DTS doesn’t recommend doing this since it might break in future releases of system software. The current Chooser mechanism simply wasn’t set up for it.

How to tell if a window is a help balloon window

Written: 11/7/91
Last reviewed: 12/11/91

How can I tell if a window is a help balloon window?

—

First, call the Help Manager procedure HMIsBalloon, to determine whether a balloon is being displayed at all. Then call HMGetBalloonWindow to get the help window’s window pointer, and compare that window pointer to the window you’ve got.

Note that if HMIsBalloon returns true and HMGetBalloonWindow returns a window pointer of NIL, it means that the balloon "window" that is displayed really isn't a window at all; this will happen if the balloon is being displayed on top of a pulled-down menu, for instance (we call this "to boldly go where no window has gone before"…).

HMShowBalloon styled TEHandle limit and workaround
Written: 12/9/91
Last reviewed: 2/17/92

Sometimes balloons won't show up when I call HMShowBalloon; I get a paramErr (-50) instead. The hmHelpType is khmmTEHandle. The HMShowBalloon function calls TextWidth on the hText of my TEHandle (the result of which is 1511 (338 chars)), then multiplies that by the lineHeight (12), yielding 18132. It then compares this to 17000, doesn't like the result, puts -50 into a register and backs out of everything it has done previously. What is the Help Manager doing?

—

The Help Manager checks against 17000 to ensure that the help balloon window is always smaller than the Macintosh system screen. However, the value used (17000) should be a much larger value. As it is right now in System 7, you're limited to about the same number of characters with a styled TEHandle as you are with a Pascal string: 255 characters.

To avoid this limitation, use clear concise phrases to make your help message as short as possible. If you've already done that and your help message still isn't short enough, you can use khmmPict or khmmPictHandle and use a picture for your help message. Using a picture has the disadvantages that pictures use a little more memory for the same help message and you'll have to add carriage returns for line breaks yourself where you need them. You can use MacDraw or something similar to create the pictures.

Balloon help equivalent for System 6 users
Written: 12/9/91
Last reviewed: 2/6/92

We are currently adding balloon help to our application, but have run into a problem with adding help to a custom Standard File routine. Our application needs to run under both System 6 and System 7, so we're using the older SFPGetFile() call instead of System 7's CustomGetFile. Since we're not using the new Standard File calls, we can't take advantage of the sfItemBalloonHelp item provided with the new interface. How can we add help for our custom file dialog?

—

To add help balloons for additional dialog items in a SFPGetFile or SFPPutFile dialog, add

an 'hdlg' resource with resource -4000 for SFPGet calls or -3999 for SFPPut calls. This resource contains help balloons for the standard items but with additional help items for your dialog additions. Even though your DITL will not be the standard -4000 (or -3999), Standard File will pick the -4000 (or -3999) hdlg out of your resource fork and use it with your custom Standard File dialog.

Help balloons and short messages with long words
Written: 12/11/91
Last reviewed: 2/6/92

Sometimes long words get broken across lines in certain help balloons. Is there a way to tune the line-breaking algorithm balloon help uses?

—

If your application is simply adding help resources to your dialogs, then you're pretty much stuck with the algorithm the Help Manager uses. It decides how big the balloon should be based on the length of your text and the height of your text. Then it uses TextEdit to display your text in a rectangle that size. The problem you're seeing is short messages with long words break the algorithm. The easy workaround is to make the height of your text larger. For example, try putting in one carriage return before the first word in your text and two carriage returns after the last word. You'll probably end up with an extra line of white space above and below the text in the balloon, but it will wrap correctly. The other easy solution is to insert "- " (a dash and a space) where you want your really long word to break. Both of these methods involve trying the balloon text to see if it breaks where you want it to break.

More than 239 characters in a help balloon
Written: 1/20/92
Last reviewed: 4/23/92

My help message doesn't appear when I use a 'TEXT' resource in a static window as the "message" in the balloon, following string resource examples in Inside Macintosh Volume VI and modifying the code to indicate 'TEXT'. Why doesn't this work?

—

While using 'TEXT' resources in Help Manager balloons is a way to provide stylized text, it does not mean that strings longer than 256 are possible. In fact, strings only up to 239 characters in length are valid. String lengths greater than 239 result in

Workaround for help menu MenuKey bug
Written: 2/11/92
Last reviewed: 6/11/92

Menu accelerators don't work in System 7.0's help menu. In addition, mouse clicks don't work in hierarchical menus added to the help menu. For example, we added to our help menu—a "quit" item with "Q" as the accelerator and a submenu containing an "exit" item with "X" as the accelerator to our menu. Selecting "quit" or "exit" should quit the program, but we are not getting that result.

—

Yes, this is a bug in the Help Manager and Menu Manager. The system is treating any MenuKey return from the help menu as a “system” hit instead of an application menu hit. It should be addressed in the next major system release . In the meantime, you’ll have to track menu keys in the help menu yourself, which is an added step in your keyDown handler, but is the only way to do it currently…

```
case keyDown:
if(ERecord.modifiers & cmdKey){
if((ERecord.message & charCodeMask) == myMenuHelpKey){
DoHelpMenu();
} else {
/* do regular MenuKey call */
 }}
```

Be sure to use a variable for myMenuHelpKey, instead of a constant, and read it from the menu when you start. This will prevent you from having to recompile if you change the help menu key for design or internationalizing reasons.

HMGetIndHelpMsg and 'hrct' resources

Written: 2/24/92
Last reviewed: 6/11/92

This is regarding a problem with the Help Manager. My application opens a number of resource files. If my 'hrct' resources aren't in the most recently opened resource file, I get an error -192 (resource not found) from HMGetIndHelpMsg. Is this a Help Manager bug?

—

Your 'hrct' resources get loaded only from the most recently opened resource file because that's where the Help Manager is looking for them. The Resource Manager uses Get1Resource to get 'hrct'-type resources to avoid Resource Manager conflicts with other resources. This is simply a behavior of the Help Manager. To properly use HMGetIndHelpMsg, call UseResFile with the refNum of the file containing the 'hrct' before you call the Help Manager. If the 'hrct' is stored in your application resource fork, you can use HomeResFile or CurResFile (at the start of your program before you've opened any other resource files) to get the refNum of the application resource file.

Not all Help Manager resources are treated this way due to the overall design of the Macintosh toolbox; 'hdlg' resources, for example, are loaded using GetResource so that things like Standard File can have common help throughout all applications.

HMSetBalloonContents controls runaway balloons

Written: 5/24/91
Last reviewed: 10/9/91

Help! I've just added Balloon Help to my application, but I'm having some problems. Whenever a balloon appears, it immediately begins floating away off the top of the screen. What can I do to stop this madness?

—

It appears you failed to heed our warning when it comes to routines that can move balloons. Consult Appendix D of Inside Macintosh X-Ref, "Routines That May Pop Balloons or Cause Barometric Disturbances," for a complete listing of these help balloon meteorological nightmares. In addition, be sure to call the new trap HMSetBalloonContents:

```
OSErr HMSetBalloonContents (balloonContents: INTEGER);
```

```
CONST { types of balloon contents }
   helium = 0;
   air = 1;
   water = 2;
   whippedCream = 3;
```

with balloonContents set to something greater than helium.

HMSetMenuResID & kHMCompareItem with 'hmnu' resource

Written: 3/14/91
Last reviewed: 7/29/91

My DA uses a lot of hierarchical menus for which I have help strings in a 'STR#' resource. How do I get around potential renumbering problems? Is it safe to assume that I never get renumbered under System 7? Also, if a menu item has a hierarchical menu, how do I get help for the item?

—

According to the Help Manager chapter of Inside Macintosh Volume VI,

“You can use the HMSetMenuResID function to set the 'hmnu' resource for a menu that did not previously have one or to supplement the existing 'hmnu' resource for a menu.

```
   FUNCTION HMSetMenuResID (menuID,resID:  Integer) :  OSErr;
```

The menuID parameter specifies the menu to associate with the 'hmnu' resource.The resID parameter specifies the resource ID of the 'hmnu' resource to use for the menu specified by the menuID parameter. The menu identified by the menuID parameter should correspond to an existing menu in your menu list. The Help Manager maintains a list of the menus whose 'hmnu' resources you map (set or override) using the HMSetMenuResID function.

Specify –1 in the resID parameter to unmap the pairing of a particular menu and 'hmnu' resource that you previously mapped using the HMSetMenuResID function. You should unmap any resource IDs before your application quits.

Result codes

```
      noErr                0        No error
      memFullErr        -108        Not enough room in the heap zone”
```

Regarding help items for hierarchical menus, you can use kHMCompareItem in your 'hmnu' resource to match against the menu item in question, and display the appropriate help message at that time. See the “Providing Help Balloons for Menus” section for details on the HMCompareItem identifier.

Changing the default help balloon for your Macintosh application

Written: 4/8/91
Last reviewed: 5/1/91

In the Macintosh Finder, when you point to the TeachText icon with Balloon Help turned on, a little description about the application appears. How can I create a similar description for my application?

—

TeachText is a special utility application included with Macintosh system software. Consequently, some TeachText-specific information is stored in system files. Specifically, the Finder Help file contains special strings which are used for the TeachText help balloons. Normal applications do not have this privilege.

You can create an 'hfdr' resource in your file, which lets you specify your own help balloon text. You can use the Rez tool in MPW SARez, or BalloonWriter to make an 'hfdr'. Refer to “Overriding Help Balloons for Application Icons” in the Help Manager chapter of Inside Macintosh Volume VI for more details.

Help balloons can’t be attached to HyperCard palettes
Written: 4/12/91
Last reviewed: 8/30/91

Help balloons cannot be attached to HyperCard palettes. The problem is that palettes are structures that are built on the fly and handled by an XCMD called “Palette.”. Short of writing your own Palette XCMD, there’s no known way to attach help balloons to HyperCard palettes or to their contents.

Determining within HyperTalk whether balloon help is enabled
Written: 4/12/91
Last reviewed: 8/30/91

There’s no easy way to tell whether Balloon Help is enabled from within HyperTalk, but an XFCN could be written to call HMGetBalloons, which would return a boolean indicating whether Balloon Help is enabled.

Macintosh Help Manager and resource files
Written: 4/29/91
Last reviewed: 6/7/91

Is there some kind of bug with the Macintosh Balloon Help feature in System 7.0? It goes into the application and uses the text from the 'STR#' resource ID 4001 instead of the 'STR#' resource ID 4001 of Finder Help.

—

The System 7.0 Help Manager stores its balloon strings in the Finder Help resource file.

Under certain circumstances, the Help Manager will access the string resources in your application before it accesses the resources of the Finder Help file. Consequently, problems will occur if your application contains certain 'STR#' and 'STR ' resources.

There are only two circumstances in which your application may be affected. First, if your application uses a 'STR#' resource with ID 4001, the Help Manager will use the first string resource of the list instead of the corresponding resource in Finder Help. When the pointer is placed over your application icon's text on the desktop, the default text, "Change the icon's name by clicking on the name and typing," will be changed to the text stored in 'STR#' 4001 of your application.

Second, if your application has a 'STR ' resource with ID 17251, the Help Manager will use that string resource instead of the corresponding string resource in Finder Help. The default text, "This is an application—a program with which you…" will be changed to the text stored in 'STR ' 17251 of your application.

To avoid these problems, you have a few options. You can create your own 'hfdr' resource to override 'STR ' 17251, or you can avoid using 'STR ' 17251 and 'STR#' 4001. If you must use 'STR#' 4001, paste the text "Change the icon's name by clicking on the name and typing" into the first string and use the rest of the strings for your application's use.

Control Panel cdev Balloon Help
Written: 8/26/91
Last reviewed: 9/16/91

How do I completely mimic the help in Apple's Control Panel cdevs? Apple's cdevs have two forms of help: (1) point to the icon and you get the descriptive text of what the cdev does, and (2) point to the name and you get both the descriptive text and help on how to change the name. I can only get (1) from the (hfdr, -5696) resource. Also, when I open my control panel, THEN point to the icon, I get the default help message, as if my balloon help were not present. If the control panel isn't open, then the help is correct. Apple's cdevs don't do this; the correct help is displayed regardless of whether or not the cdev is open. What's up?

—

The truth is that you cannot mimic the help given for Apple's cdevs. The Finder is special-cased for the items that Apple includes on its disk, and in the Finder help file you'll find all the help messages that go along with the cdevs and DAs and such. You'll also find some 'fmap' resources. These basically point the Finder to the proper help message for each filetype/creator combination that Apple has. This is how the Finder knows to treat the icon specially. You aren't supposed to modify these resources, so there is no way to mimic the Apple's cdev help.

cdevs and foreign language balloons
Written: 8/26/91
Last reviewed: 9/16/91

My cdev contains a 'STR#' resource for each foreign language we support. Prior to Balloon Help, text was taken from the resource file by doing a GetResource('STR#', BASE_ID + countryCode). Now, with Balloon Help, I have to have fixed resource IDs in the 'hdfr' and 'hdlg' resources. It would be nice to be able to tell the Help Manager to use the BASE_ID + countryCode selector, though as a workaround, the INIT associated with the cdev renames resources so that the correct text is selected.

___

You can set the dialog resource number explicitly from the code. Use the HMSetDialogResID (dialog number + country code), and have a different 'hdlg' for each country.