New Technical Notes

Macintosh

Developer Support

Movable Modal Dialogs

Toolbox

Written by: “im” BeninJames ghaus August 1991

This Technical Note describes the process by which an application can remap the Help Manager 'hmnu' resource while a movable modal dialog box is on the screen. The Help Manager handles the case for modal dialog boxes but punts in the case of movable modal dialog boxes. The following information will help you get the correct interface performance.

What’s involved

The System 7 support for movable modal dialog boxes is limited to providing the new ‘WDEF’ variant. The rest of the implementation of movable modal dialog boxes is left to the application. Applications must provide handling for all events intended for a movable modal dialog box. This could be accomplished by calling the IsDialogEvent and DialogSelect Toolbox routines, or using other Toolbox routines such as FindWindow, BeginUpdate, DrawDialog, EndUpdate, TrackControl, TEClick.

How you process the events is up to you, but when it comes to appropriate balloon help the application must call the EnableItem, DisableItem and HMSetMenuResID Toolbox routines. The HMSetMenuResID is used before and after enabling or disabling the menus. HMSetMenuResID routine maps an alternative 'hmnu' resource to your menus.

The Systems 'hmnu' string resource

Listed here are two alternative 'hmnu' resources. The first one uses the same strings that the Help Manager shows when ModalDialog is called. The constant kHMHelpID is defined in the interface files BalloonsTypes.r, Ballons.h, and Balloons.p. In general it refers to the ID of various Help Mgr resources. In this case it selects a STR# resource in the System and the constants 31 and 32 refer to the string index within that resource. These strings are the ones the Help Manager uses when a Modal Dialog Box is on the screen.

```
resource 'hmnu' (256,"System Movable Modal Dialog hmnu") {
       HelpMgrVersion,
       hmDefaultOptions,
       0,
       0,

       HMStringResItem {          /* Missing items */
```

```
0, 0,
kHMHelpID, 31,
0, 0,
```

```
                0, 0,
        },
        {
                HMStringResItem {   /* Menu Title */
                        0, 0,
                        kHMHelpID, 32,
                        0, 0,
                },
        }
};
                        0, 0,
```

An alternate 'hmnu' from your application

If you don’t want to display the same strings that the Help Manager displays for Modal Dialog Boxes, you can map in your own alternate 'hmnu' resource such as the following.

```
resource 'hmnu' (256,"Application Movable Modal Dialog hmnu") {
        HelpMgrVersion,
        hmDefaultOptions,
        0,
        0,

        HMStringItem {              /* Missing items */
                "",
                "This item is not available because it cannot be used with"
                "the About box on your screen.",
                "",
                "",
        },
        {
                HMStringItem {      /* Menu Title */
                        "",
                        "This menu is not available because it cannot be used with"
                        "the About box on your screen.",
                        "",
                        "",
                },
        }
};
```

Using the alternate 'hmnu' resource

After displaying the movable modal dialog box on the screen, the application should disable inappropriate menus and items and map in the alternate 'hmnu' resources.

```
        menu = GetMHandle(mApple);
        DisableItem(menu, 0);
        HMSetMenuResID(mApple, 256);
        menu = GetMHandle(mFile);
        DisableItem(menu, 0);
        HMSetMenuResID(mFile, 256);
        menu = GetMHandle(mEdit);
        DisableItem(menu, 0);
        HMSetMenuResID(mEdit, 256);
        DrawMenuBar();
```

Removing the alternate 'hmnu' resource

After removing the movable modal dialog box from the screen, the application must enable appropriate menus and items and unmap the alternate 'hmnu' resources.

```
menu = GetMHandle(mApple);
EnableItem(menu, 0);
HMSetMenuResID(mApple, -1);
menu = GetMHandle(mFile);
EnableItem(menu, 0);
HMSetMenuResID(mFile, -1);
menu = GetMHandle(mEdit);
EnableItem(menu, 0);
HMSetMenuResID(mEdit, -1);
DrawMenuBar();
```

Note: The previous fragments of code do not perform error checking. Well-behaved applications perform error checking whenever required. In these example the menu handle should be checked for a nil value before calling DisableItem and EnableItem.

Further Reference:

- *Inside Macintosh*, Volume VI, Help Manager