

F2. Initializations

Facelt-based programs always contain a few lines of code devoted to initialization followed by a "main event loop". The following program, for example, would automatically support access to Desk Accessories, interact properly with MultiFinder and System 7, and support several standard menu items (Pascal source):

```
fRec.uName := 'Minimum.Rsrc';
Facelt(nil,DoInit,0,0,0,0);
repeat
  Facelt(nil,DoLoop,0,0,0,0);
until false;
```

where the DoInit and DoLoop commands are described in the "Program Commands" topic in the "Commands" menu, and the "Minimum.Rsrc" file contains the MENU resources loaded by Facelt plus the LoadIt module needed to call FaceWare modules.

The simplicity of the above example program is one of the distinguishing characteristics of programming with Facelt. Nearly all additional code that you add to a program will be program-specific, making a "code generator" unnecessary when programming with Facelt.

The remainder of this topic discusses the major initializations that are automatically done by Facelt on DoInit. Each of these can be disabled by simply removing the associated resources from the program's resource file. Other initialization-related actions taken by Facelt can be optionally disabled with bit flags passed in parameters a and c when calling DoInit (described under DoInit in "Program Commands" topic in the "Commands" menu).

Menu Initialization

Nearly all Mac programs make use of the main menu bar at the top of the main screen. The pull-down menus in this bar are always available unless they have been disabled by the program or a modal window has been opened. Menus in the main menu bar often contain hierarchical menu items attached to hierarchical menus, and windows often contain pop-up menus. Facelt and ViewIt provide complete support for initialization and management of all three of these common menu types.

• Auto-Loaded Menus

When DoInit is called, Facelt looks for MENU resources in the program file or the program's temporary resource file (w/ file name = uName) and automatically initializes and installs these if they have been properly numbered. Facelt searches for MENU resources with res IDs starting at 1001, 1002, ..., that have menu IDs of 101, 102, ... (menu ID = res ID - 900). MENUs continue to be loaded until 6 consecutive res IDs are encountered that do not have associated MENU resources (so you can leave gaps in your res ID numbering for later additions).

MENU resources can be created and edited using Apple's ResEdit resource editor (included on the FaceWare Utilities disk if you purchased ViewIt). The "menu ID" of each MENU resource can be set by opening the "Edit Menu & MDEF ID" dialog in ResEdit. The following table reviews how menu ID numbers are used in Facelt-based programs:

- 101-190 - recommended for main program menus
- 191-195 - reserved for temporary use by drivers
- 196-199 - reserved for ViewIt's FSSC menus
- 236-255 - reserved for use by ViewIt controls & DAs

where the range of menu IDs 101-190 would correspond to program MENU resources with res IDs 1001-1090.

What To Do: Most programmers begin programming with a copy of a demo resource file that already contains Apple, File, Edit, and Window menus (corresponding to resource IDs 1001-1004 and menu IDs 101-104). They then use ResEdit to modify these menus and/or to add new menus to support program-specific options. **REMEMBER:** When adding a new menu that you want Facelt to auto-load, don't forget to set its menu ID to res ID - 900 or Facelt won't load it.

• Hierarchical Menus

Hierarchical menus can also be auto-loaded by Facelt. The only difference between these menus and menu bar menus is that (1) the first character of their menu titles should be a "+" or "-" to inform Facelt that the menu is not a menu bar menu (don't worry, this character is never displayed), and (2) there should be at least one menu item in another menu that is linked to this hierarchical menu. ResEdit can be used to establish the link by checking the "has Submenu" option when editing the hierarchical menu

item and then entering the menu ID (not the res ID!) of the linked menu.

Hierarchical menus work best if they are numbered in the same, orderly manner as that described above for menu bar menus. For example, if MENU 1005 (menu ID 105) has 3 hierarchical menu items, then the associated hierarchical MENUs might be numbered 1006, 1007, 1008 (menu IDs 106, 107, 108), with the next main menu bar MENU being numbered 1009 (menu ID 109).

ResEdit Bug: The "Open Submenu" option in ResEdit only works if the submenu's resource ID is equal to its menu ID, which will not be the case for menus auto-loaded by Facelt.

- Pop-Up Menus

Pop-up menus are similar to hierarchical menus but are usually not linked to hierarchical menu items. Rather, they must be "popped up" by calling "PopUpMenuSelect" (or the utility command PopMen). In most cases, however, pop-up menus used by Facelt-based programs are those supported by menu controls in ViewIt windows, and these are both initialized and managed by control drivers without the need for the main program to call PopUpMenuSelect.

In cases where your program must initialize and manage the pop-up menu without ViewIt's help, Facelt can still be used to auto-load the menu by assigning it a res ID and menu ID in accordance with the scheme presented above, and by setting the first character in its title to "+" or "-" (as with hierarchical menus). If such a program-managed pop-up contains standard items, then Facelt's DoMenu command should be used to process items selected from the menu.

Window Initialization

- Main Help Window

If ViewIt's on-line help and editing resources are available, Facelt will attempt to open the main ViewIt Help window. This feature can be disabled by adding 64 to parameter c when calling Dolnit (as described under Dolnit in the "Program Commands" topic) but is not something that most programmers worry about since finished programs will usually not contain the on-line help resources required to open the help window.

- Splash Screen

Facelt opens a default "splash screen" when Dolnit is called. To replace this splash screen with your own, create a new FWND resource and name the resource "Splash" (any resource ID). Facelt will then use this "Splash" FWND in place of its own when opening the splash window. Note that your splash window should not have a title bar, and its title (set in ViewIt's Window dialog) should be equal to the minimum time in seconds that Facelt should keep the splash screen open. To prevent any splash screen from being shown, add 32 to parameter c when calling Dolnit.

Palette Initialization

The scheme used by Apple to designate colors on a Mac that supports Color QuickDraw allows you to choose from any one of 16 million colors. Unfortunately, you are not always able to display this many colors, and which of these can be displayed is determined by the "depth" of your screen and the current color "palette". The screen depths supported by Color QuickDraw are 1, 2, 4, 8, 16, and 32 bits per pixel, corresponding to 2, 4, 16, 256, 32768, and 16 million displayable colors (only 15/16 and 24/32 bits are used to define RGB colors at the higher screen depths).

Each program has a "program-wide color palette" which determines which colors will be displayed from the 16 million possible red, green, and blue combinations. If necessary, you can use the utility command SetPal to change the colors in this palette (described under SetPal in "Color Utilities"), but this will only be necessary if you need a large number of colors that are not available in the default palette (such as 256 shades of gray). Experienced programmers can also obtain a handle to the current program-wide palette by calling "GetPalette" with an argument of -1 (i.e., "GetPalette(WindowPtr(-1))"). This handle can then be used with other Color QuickDraw calls to directly manipulate the contents of the palette.

What To Do: The only programmers who need worry about resetting the program-wide palette are those whose programs need to define a large fraction of the displayable colors (such as 256 shades of gray on an 8-bit deep monitor).