

F9. Facelt Commands

Facelt supports commands that deal with "program-wide" features. See ViewIt Guide for window-related commands and the Utilt Guide for low-level utility commands.

If working with Facelt (vs. FaceSt), and a modal window is not the front window, then standard label ID numbers can also be used as direct Facelt commands. The following line, for example, would execute the standard "Copy" menu item:

```
Facelt(nil,14,0,0,0,0); #14 = Copy
```

Dolnit -61 a,b,c,d,uName

This command was first described in the "Initializations" topic in the "Startup" menu. The additional information presented here only applies if using Facelt.

If using Facelt or FaceSt ($c \geq 0$), then parameter a can be used to set various Facelt or FaceSt options that affect program operation, and parameter c can be used to disable certain default operations performed by Facelt or FaceSt when Dolnit is called.

a = sum of following constants (bit flags):

1* = change cursor to deck of cards when returning control from DoLoop (can be helpful during development)

2 = return control to program whenever active window is changed (returns with uMenuID = 1100, uMenuitem = 2, uResult = 0)

4 = return control to program whenever clipboard is changed (returns with uMenuID = 1100, uMenuitem = 4, uResult = 0)

16* = disable all auto-saving on Quit or Transfer (no Save Document notices are posted to FCMD modules)

32 = return control to program after successful "Save Settings" (returns with uMenuID = 1100, uMenuitem = 32, uResult = 0)

64* = flush application events 12 and 13 from event queue before returning control (bug fix that prevents crash in MacFortran 2.4 Debugger)

512* = return control to program if user attempts to Open or Print a file "from the Finder" (returns with uMenuID = 1100, uMenuitem = 512, uResult = 1 (Open) or 2 (Print), uString = file type, uName = file name, and the current directory is reset to that containing the file)

1024* = return control to program before ExitToShell or transfer is executed by Facelt (see discussion below)†

c = sum of following constants (bit flags):

1 = skip toolbox initializations (done in FaceProcXY)

2* = skip auto-initialization & insertion of main menus

4* = skip "About ProgramName" standard item setup

8 = skip resetting of program-wide palette to clut 1000

16* = skip processing of old-style Finder info (this prevents Facelt from calling "ClrAppFiles" so that you can process this information yourself)

32* = skip putting Font names into STR# 1106††

where the constants that are followed by "*" correspond to functionality not supported by FaceSt, and "return control" in the context of FaceSt simply means that a pseudo-menu event is posted.

† Adding 1024 to a causes Facelt to return control to the program just before executing ExitToShell (if Quit chosen) or transferring to another program (if Transfer chosen). The message returned has uMenuID = 1100, uMenuitem = 1024, uResult = 1 (Quit) or 2 (Transfer), and uName = name of program to transfer to. Any subsequent call to Facelt will then cause the program to immediately quit or transfer according to the contents of uName: if uName is empty, then the program quits; if uName is not empty, then the program transfers to the named application. If you reset uName to transfer to a different application, be certain that the current directory contains the application.

Also note that bit flag 1024 does not replace the use of negative label numbers ("#-106") in menu items, since these can be used to get control before Facelt goes through its Quit or Transfer-related operations

(opening of the Transfer dialog, prompts to save changes, etc.), whereas bit flag 1024 returns control after these operations are performed and the Quit or Transfer cannot be cancelled.

†† An alphabetical list of fonts available at launch time is placed in STR#1106, where each font name in the list is preceded by a "" character (ASCII 96). This list made it easier to create scrollable font lists in old DialIt dialogs, but has no use within ViewIt windows, so you may want to disable this feature by adding 32 to parameter c.

DoAuto2 -54 d,uString

Opens a new window by executing the auto-initialization string found in uString. This string should include the baseID, versID, resID, and any other parameters required by the window driver: "[baseID],[versID],[resID],..." If using ViewIt, this command is obsolete since the NewWnd command accomplishes the same task. Parameter d can be used to specify the degree of updating done after the window is opened, and has the same meaning as in DoUpdt2.

NOTE: DoAuto2 is not supported by FaceSt, and should not be called when a modal window is open.

DoLoop 0 a

Used to enter Facelt's main event loop which automatically handles most events. Control remains with Facelt's event loop until an event occurs that "belongs" to your program. Information about such program events is usually returned as menu or pseudo-menu events in the variables uString, uMenuID, uMenuItem, and uResult.

Parameter a can be used to have control returned to the program immediately after calling DoLoop. If a = -1, then control is returned after updating all windows, the active window status, etc. (same as calling DoUpdt2 w/ d = -1). If a = -2, then control is only returned after processing all pending events. These options are similar to the options supported by parameter b when using MdlWnd to handle events in modal ViewIt windows.

NOTE: DoLoop is not supported by FaceSt, and should not be called when a modal window is open.

DoEvtnt -51 fEvent

Forces Facelt or FaceSt to handle the event in fEvent. If using Facelt, then DoEvtnt is typically used to pass Facelt update or switching events encountered while programs are running in the background (see "Background Processing" topic). If using FaceSt, then DoEvtnt is used to pass FaceSt all FaceWare window-related events that are not handled by the main program (see "Hybrid Programs" topic).

NOTE: DoEvtnt should not be called when a modal window is open.

DoMenu -52 a,b,uMenuID,uMenuItem

Forces Facelt or FaceSt to handle the menu event passed in parameters a (menuID) and b (menu item number). If a and b are zero, then Facelt and FaceSt look in uMenuID and uMenuItem for the identity of the menu item to execute. DoMenu is typically used with FaceSt to pass it selections from main program menus. If the menu selection does not correspond to a standard menu item, then FaceSt reposts the menu event as a message returned by GetMsg.

NOTE: DoMenu should not be called when a modal window is open.

DoUpdt2 -53 d

Updates stuff according to the value of parameter d:

0 = no updating

1 = redraw all window contents that need updating

8 = update identity of active window, fix standard menu items, and reset current port to active window

-1 = update all of above (all bits set)

When using FaceSt, DoUpdt2 is called with d = 8 to update active window-related variables when the active window is changed. Another common use is to call DoUpdt2 with d = 1 to force the content of an erased window to be redrawn so that new drawing by the program is displayed properly.

GetMsg -55 [none]

FaceWare modules return special messages and events to the main program by placing them in a

private queue with the Utilt command PstEvt (see "Other" topic in the Utilt Guide for more info). When using Facelt, messages from the private queue are automatically processed and returned from DoLoop as menu or pseudo-menu events. When using FaceSt, the main program must call GetMsg to get the next message in the queue if the queue is not empty (see "Hybrid Programs" topic for details).

NOTE: GetMsg is not supported by Facelt.

GetFWs 157 a,b,c,d,uResult

Returns information about the current FaceWare windows present in the program, where a determines the type of information to return, b, c and d determine which windows to include in the search, and uResult returns the number of windows found. If uResult > 0, then the information is returned in STR# 1101 in the order given below (title, file name, etc.) for each of the windows found (1 contiguous set of strings per window, front to back).

a = sum of bit flags designating information to return:

1 = window title

2 = file name (whatever is between "" and "" in title)

4 = window pointer

8 = address of shared record

16 = baseID (from shared record header)

32 = resID (from shared record header)

64 = refCon from WindowRecord

b = shared record address filter

0 = accept all shared record addresses

other = window's record address must be equal to b

c = baseID filter

0 = accept all baseID values

other = baseID must be equal to c

d = resID filter

0 = accept all resID values

other = resID must be equal to d

The window pointer, shared record address, and refCon are returned as 4-character strings, and baseID and resID as 2-character strings, where the characters in each string are equal to the integers of interest (no string-to-number conversion is necessary). The Utilt commands GetI4 and GetI2 can then be used to move these values into the fRec variables ul4 and ul2.

When working with older, non-ViewIt windows, a common use of GetFWs was to use it to retrieve a pointer to the shared record that gets dynamically allocated by an FCMD module when auto-initializing a window. To retrieve this pointer, GetFWs was first used to find the window and move its record pointer to STR# 1101. GetI4 was then used to move the pointer value from STR# 1101 to ul4.

NOTE: GetFWs is not supported by FaceSt.