Macintosh Technical Notes

#194: WMgrPortability

See also:	The Window Manager
	QuickDraw
	MultiFinder Development Package

| Written by: | Rick Blair | March 1, 1988 |

Where `WMgrPort` (the Window Manager's port), MultiFinder, and drawing outside of one's windows will be reconciled.

Beware

Drawing outside of windows from within an application is guaranteed to make that application less compatible with future systems. In order to be as MultiFinder compatible as possible, draw only in response to an update event or as part of the feedback for a user action, i.e. while tracking the mouse. MultiFinder compatibility is just as important as HFS compatibility!

MultiFinder documentation warns against drawing in `WMgrPort` since the system "owns" the desktop and windows of other applications besides your own are drawn within it. This note will tell you how and when to draw outside the confines of your own windows if you feel that you must.

In the future the system may provide calls for drawing outside your windows safely. When that occurs, the techniques described here may no longer be valid. Nevertheless…

WMgrPort and GrayRgn

`WMgrPort` has its `visRgn` set to include all active screens. Its `clipRgn` is initially set to "wide open" (the rectangle –32767, –32767, 32767, 32767), although Window Manager routines like `ClipAbove`, etc. will change it. Consider this `GrafPort` read-only. The global variable `GrayRgn` is a region which is equal to the `WMgrPort`'s `visRgn` minus the menu bar area.

Note that you should use `GrayRgn`, which is the best way to find out the shape, size, and coordinates of the screens. You will never have to use the `WMgrPort` directly, and should not call `GetWMgrPort` under any circumstances.

Rules

Only draw to the whole screen/desktop in a "modal" way. This can take the form of a brief animation across windows or the visual feedback for dragging from one window to another. It is important to know that no other application (including the Finder) will draw until you have finished. To guarantee this, you must follow some rules:

In the case of a drag, you should only draw while the mouse button is down. In the case of an animation effect, the drawing should be of brief duration. All operations should conclude with nothing left drawn outside your windows. Under MultiFinder (version 1.0 and 6.0 at least) you will be OK if you don't call `GetNextEvent`, `EventAvail`, or `WaitNextEvent` while drawing outside your windows. Use the `StillDown` function (or the `WaitMouseUp` function) for loops that wait for the mouse button to go up. Remember, however, it is only through possible future system-provided calls that you can be completely safe from others drawing underneath you.

Never draw something on the desktop and leave it there. There is no way to tell the system that you have drawn on that bit of desktop, so the Finder will draw right over you.

Examples

The most famous animation effect is the `ZoomRect` routine. It is used by the Finder to draw a series of nested rectangles around an icon that is being opened. The rectangles form a progression (zoom) out to where the window for the icon will be placed.

Another, potentially more interesting, case is where you want to drag something from one window to another, perhaps to copy it. This is often done with `DragGrayRgn`, which for this purpose will do the right thing (not call `GetNextEvent`, etc.).

How to do these effects

Use a `GrafPort` (not a window or the `WMgrPort`) that covers all the screens. `OpenPort` will set up most of the fields of the `GrafPort` properly. All you have to do is change the `visRgn` of your port to a copy of `GrayRgn` and put the `GrayRgn`'s `rgnBBox` into your `portRect`. Directly manipulating the `visRgn` of a **window** is a no-no under MultiFinder.

Draw using `srcXor` mode. This will allow you to erase as you go, by drawing each object a second time, also in `srcXor` mode. You must leave all areas outside your windows exactly as you found them.

WDEFs and MDEFs

Window and menu definition procedures draw in the current port, which is set to the `WMgrPort` by the Window Manager and the Menu Manager. Note that this means that you do not ever have to call `GetWMgrPort`, as mentioned above. We recommend that you never draw into it except from one of these procedures.