About Graphics Devices Information about graphics devices

When you turn on the machine, the system gathers all of the gDevice records into a DeviceList that shows all of the video cards attached to the Mac. The **graphics environment** is defined by GrafPorts, which contain information about windows, and graphics device records, which contain information about graphics devices such as screens that are attached to the system. Unless you design startup procedures that specify otherwise, the gDevice record that points to the video card in the lowest-numbered slot is the first one encountered on the DeviceList, is designated as the active device and, hence, the default startup screen. The startup screen is also known at the main screen in multi-display setups and is the one that carries the menu bar.

An entire new set of functions and procedures have been created to deal with multi-screen devices. With them, the user can set the desired depth, display mode, presence or absence of a menu bar, and relative position of each video display. All of these capabilities are available to the user through the **Control** Panel. A new resource type ('scrn') was created to store all of this information in the system file.

When you start a Quickdraw operation, InitGraf looks for 'scrn', which has the job of organizing the screens in a multiple display setup. If 'scrn' is missing, InitGraf assumes the presence of only a single screen.

Calling InitWindows starts a scan of the device list. From that list InitWindows puts together an area that combines all active screens and saves it as the GrayRgn global variable. GrayRgn, in turn, gives the parameters of the desktop.

Use the construction, FillRgn(GrayRgn,myPattern), for programs that paint the desktop; and write your programs so they pin moveable objects to the GrayRgn instead of the the screen boundary. The latter point will let the **Window Manager** stretch or drag windows from screen to screen while still letting **Quickdraw** draw to the window's port as if it were only looking at a single monitor. It accomplishes this by setting the window's base address field equal to the base address of the main screen. **Quickdraw** then compares the base address of the GrafPort or CGrafPort to the base address of the main screen. When they're equal, **Quickdraw** knows it may have to draw on more than one screen.

When there is more than one screen, **Quickdraw** calculates the entire drawing area, intersects the destination rectangle with the device's rectangle and issues the drawing command to that portion of the total area when there's an overlap. **Color Quickdraw** also provides the correct pixel value for the foreground and background colors and re-expands patterns and other structures appropriately for each device.

If you are developing a graphics-intensive application, you may need to create additional CGrafPort and graphics device records, in which case the **Graphics Devices Manager's** high-level routines probably provide all the features you need. With these routines you can create an **offscreen graphics world**: a graphics environment which you control specifications such as pixel depth and whether the pixel values are indexed or direct. Using these routines minimizes the possibility of compatibility problems, because these routines and data structures, collectively called the offscreen graphics world interface, will be maintained by Apple as the Macintosh graphics systems evolve. With system software version 7.0, the offscreen graphics routines are available on black-and-white machines (lacking Color QuickDraw), but they create an extended GrafPort record, not a true CGrafPort record.