

August 3, 1988  
Vol. 1, No. 13

## In the World of the Presentation Manager Graphics Engine

by Tom Bates

This article briefly describes the graphics engine component of the Presentation Manager. The graphics engine is the artist in the family of Presentation Manager components in that it is the one which knows how to draw. More specifically it's the brain of the artist, with the device drivers (for a display or plotter for instance) being the hands of the artist.

Before anything gets drawn on screen or paper the call goes through the graphics engine on its way to a device driver. The picture below shows where the graphics engine fits in the multilayered world of the Presentation Manager. I've left out some pieces like the Hursley shell and the OS/2 kernel. As you can see from the picture, applications don't actually make calls to the graphics engine. Instead they call GPI (which stands for Graphics Programming Interface) or PMWIN (which is the window manager component). These are the components which actually call the engine.

The GPI layer is one of the PM components being

developed by the IBM people at Hursley. It provides a layer where graphics orders can be accumulated into complete pictures or pieces of pictures called segments. This would allow an architectural design application for example to draw a fancy window into a segment just once and then draw that segment everywhere on the building elevation where the window appears even if it has a different size or orientation in some locations.

The drawing doesn't start though until the calls get to the graphics engine. There are very few entry points to the graphics engine. For the purposes of this introduction all of these can be thought of as a single entry point which is called GreEntry(). GreEntry() is also referred to as the dispatcher for reasons which will become clear in a moment. The number of arguments to GreEntry() depends on what the desired graphics operation is.

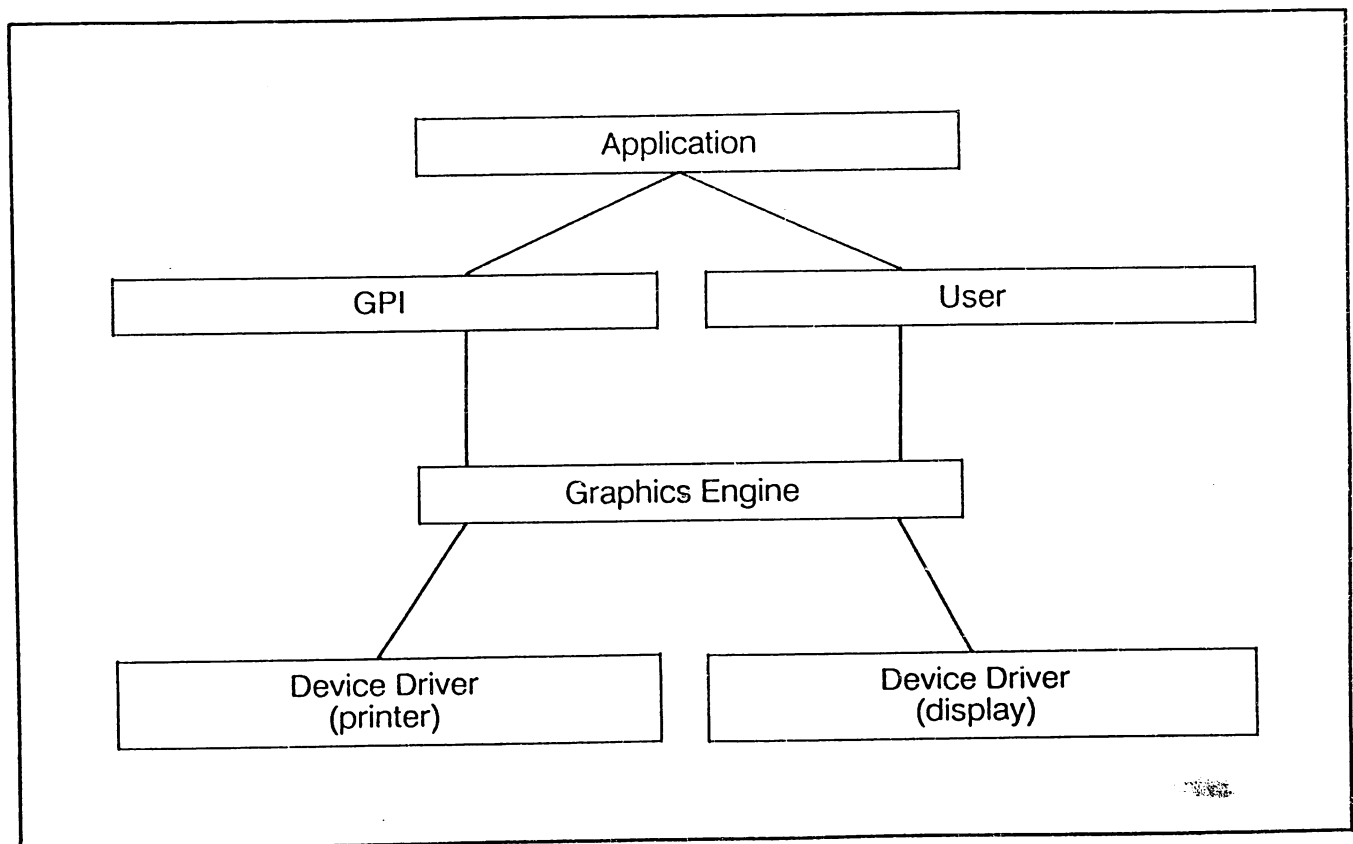
From the point of view of the dispatcher the two most important arguments are the hDC and the function number. The function number identifies

the operation to be performed (draw a line, draw a box, etc.) and the hDC specifies the device context on which the operation should be performed (i.e. the display, the printer, etc.). GreEntry() uses the function number to look up the address of the function that will do the real work. The function might reside in the engine itself or in the device driver. These addresses are stored in a table associated with the specified device context. Thus two calls to GreEntry() with the same function number but different hDCs can lead to two entirely different pieces of code actually performing the operation. This is because when a particular device is initialized and associated with an hDC it can write its own addresses into the dispatch table associated with that DC.

The DC is much more than just a table of addresses though. It is at the heart of drawing under the Presentation Manager and has a long and distinguished pedigree in graphics systems. It contains a complete drawing environment for a given instance of a given device. It includes attribute "bundles" which specify things like background and foreground colors for character, line, pattern

and other drawing primitives. It contains information about clipping or transforms that should be performed when drawing takes place. It controls the collection of various kinds of information as drawing proceeds (like a tightest bounding box for everything drawn). An exhaustive description of all the wonders of a DC is just not possible here. Perhaps it is best to just quote an anonymous source who once said, "the device context is like a drawing FORCE which can be directed against various device surfaces." Sometimes these engine folks get a little carried away.

One of the important goals in designing the graphics engine was to provide a device independent graphics interface for applications. The dispatching mechanism is one of the ways in which the graphics engine does this. The application need not know anything about the specific device that will actually handle its request for a line to be drawn. It just makes a call with a generic set of draw-a-line arguments and specifies the proper hDC for the given device. The call will be dispatched by GreEntry() to the draw-a-line handler of that device which is then responsible for making sense of the arguments in a device dependent way.



Some of the functions (there are over 200 of them) are handled directly by the graphics engine and some by the device itself. In some cases the device can even return the call to the graphics engine after it has started processing it if it decides it really can't do the right thing. In such a case the engine "simulates" the operation for the device. An important feature of this mechanism is to provide a flexible device driver interface which allows applications to take advantage of "smart" devices with fancy hardware support for graphics operations while also providing high quality simulations for less capable devices.

Presentation Manager moves us forward to a future of high quality graphics on personal computers. The graphics engine is at the center of this future. It incorporates many of the best features of several past and present graphics systems such as GDI (Windows), Cedar Graphics, Postscript, Quickdraw and GDDM (IBM mainframe level graphics package). Among its array of features are a full set of linear transformations, a rich set of curve primitives (arcs, splines, fillets, etc.), the ability to clip through arbitrary shapes and a rich font technology that includes raster and vector fonts which can be scaled and drawn in just about any way imaginable.

These days the developers on the engine team (about 15 people, managed by Chuck Whitmer) are ironing out the last few bugs, with an eye on that ever closer day in October. There still are a few, but they're dropping fast and the rate (of incoming bugs that is) seems to be slowing down (finally!). We are all looking forward to the day when we can start development work on new versions.

Speaking of the future...

Upcoming versions of the graphics engine will of course go over to 32 bit mode. Of more interest from a programmer's point of view will be new

capabilities. Things under consideration here include imaging (i.e. support for image scanners), better font technology and maybe even 3D graphics capabilities. scanners), better font technology and maybe even 3D graphics capabilities.

---

## *More on Hawaii:*

---

### **Iao Valley State Park**

This valley has been a sacred spot and a place of pilgrimage since ancient times. Before Westerners arrived, the people of Maui, who came here to pay homage to the "Eternal Creator," named this valley *Iao*, "Supreme Light." In the center of this velvety green valley is a pillar of stone rising over 1,200 feet (actual height above sea level is 2,250 feet), that was at one time a natural altar. Now commonly called "The Needle," it's a tough basaltic core that remained after water swirled away the weaker stone surrounding it. Iao Valley is actually the remnant of the volcanic caldera of the West Maui Mountains, whose grooved walls have been smoothed and enlarged by the restlessness of mountain streams. Robert Louis Stevenson had to stretch poetic license to create a word for Iao when he called it "viridescent."

The road here ends in a parking lot, where signs point you to a myriad of paths that criss-cross the valley. The paths are tame and well maintained, some even paved, with plenty of vantage points for photographers. If you take the lower path to the river below, you'll find a good-sized and popular swimming hole; but remember, these are the West Maui Mountains, and it can rain at any time! You can escape the crowds even in this heavily touristed area by following the path toward "The Needle" until you come to the pavillion at the top. As you head back, take the paved path that bears to the right. It soon becomes dirt, skirting the river and the tourists magically disappear. Here are a number of pint-sized pools where you can take a refreshing dip. Iao is for day use only. On your way back to Wailuku you might take a five-minute side excursion up to Wailuku Heights. Look for the road on your right. There's little here besides a housing development, but the view of the bay below is tops!

