SuperPrint 4.0
Advanced Printing Tools for Windows
White Paper

Much has been said about the power of Windows 95, its new 32-bit architecture and its long-awaited preemptive multi-tasking capabilities. It will therefore come as a surprise to many people that significant portions of Windows 95's printing subsystem (GDI) remain 16-bit. All applications – 32-bit or otherwise – have to interact with this 16-bit architecture, called GDI 16. GDI 16 limits performance under several conditions, e.g. 1) multi-tasking while printing, 2) multi-tasking while background printing, and 3) attempting multi-threaded processing. When a user prints or spools a print job, it is slow, inefficient, single-thread-processing GDI 16 that does virtually all the work.

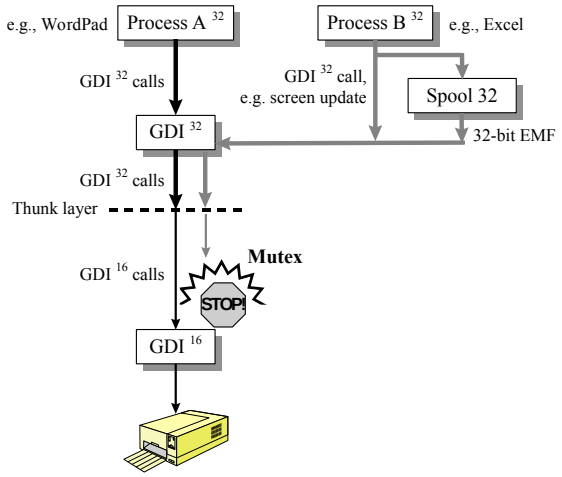
Printing with Windows 95: The Problem

Graphics Device Interface (or GDI) is Windows 95's printing and display subsystem. GDI is not reentrant (meaning that once entered, no other thread can enter until the first thread exits), so it has to be shared by all processes that are currently operating.

When a 32-bit application prints, it sends 32-bit API calls to GDI 32. GDI 32 passes the data to GDI 16 in a process known as "thunking."

Two problems arise. First, GDI 16's 16-bit rasterizing engine cannot process data as efficiently as an equivalent 32-bit engine. Second, since GDI 16 is not reentrant, it cannot work on more than one process at a time. Process A (see diagram at right) therefore sets up a barrier (a "Mutex") in GDI 16 to prevent any other process from entering GDI 16 until its work is completed. During this period, any other calls sent to GDI 16 are effectively "locked out."

Win 95's printing architecture



The "Mutex" set in GDI 16 keeps Process B waiting until the GDI 16 call from process A is completed.

The result is inefficient multi-tasking, especially bothersome if the process being worked on by GDI 16 is very intensive and time consuming, e.g. an image, or a document with lots of fonts, lots of pages, and/or other printing elements.

Any subsequent change on the screen or in printing status has to call GDI. Launching apps, opening menus, playing games, displaying progress while downloading files from the Internet, displaying images to the screen, background printing with Win 95's new metafile spooling, etc., all require processing by GDI. While the mutex is set for Process A, none of these activities can take place. Each of those processes is forced to "sleep."

Mutex side effects of the "sleeping" can be aggravating. Non-responsive buttons, screens not re-drawing, and partial message boxes displayed on top of applications are just a few example side effects from sleeping.

Sleeping can also significantly reduce CPU efficiency. If, for example, Process B receives a 20 millisecond time slice, makes a GDI call three milliseconds into this period, and is blocked by the mutex, it is forced to sleep, and it loses the remaining 17 milliseconds, which become *unrecoverable CPU time* – Process B can regain it.

Because of GDI 16, Windows 95 may not live up to users' expectations of smooth, high-performance multi-tasking while printing. Nor will CPU efficiency be optimized.

Background Printing with Windows 95: The Other Problem

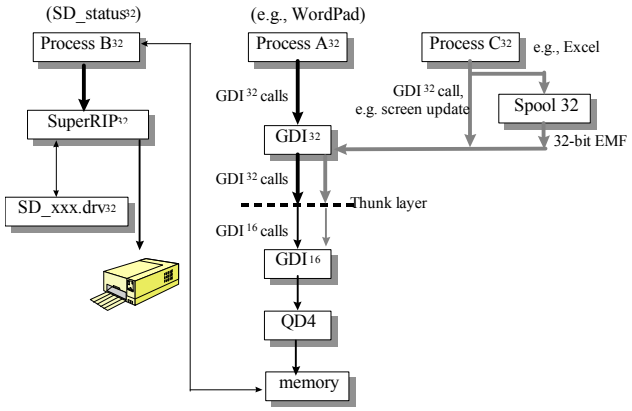
Windows 95 contains a 32-bit spooling subsystem, called Spool 32. When EMF spooling is selected (this is the default) GDI 32 quickly encapsulates GDI calls into an enhanced metafile (EMF), then returns control of the PC to the user. The EMF is then submitted to Spool 32. However, since the EMF has to pass through GDI 16 during playback, the same GDI bottleneck and process-blocking mutex will hamper multi-tasking by other processes.

Printing with *SuperPrint 4.0*: The Solution

SuperPrint minimizes the GDI 16 bottleneck by removing the rasterization process from GDI 16's control and putting it into *SuperPrint's* hands. Instead of setting the mutex and working on Process A's GDI calls, GDI 16 simply passes QD4 the calls to the appropriate *SuperPrint* modules, and clears the mutex immediately. This drastically reduces the time that the mutex is set during print job processing.

Once Process A's calls have been passed to QD4, *SuperPrint* generates a new 32-bit process (Process B). Now that GDI 16 is not working on Process A's calls, the mutex is no longer blocking any other processes. Process A and Process B look to Windows 95 like 32-bit processes, and Windows' scheduler schedules time slices for each one.

Multi-tasking While Printing with SuperPrint 95



SuperPrint allows preemptive multi-tasking even while Process A is being printed.

While Process A uses its time slices to send print data, Process B uses its time slices to deliver that data to the rasterizer within Process B. Whenever one process gets ahead of the other, it can simply "wait," which means it yields its CPU cycles to other processes. CPU cycles aren't lost, as in the case above where a process has to sleep. Instead, they're allocated to another process that needs them. The result is fast, 32-bit processing of print data; smooth multi-tasking; and efficient CPU utilization.

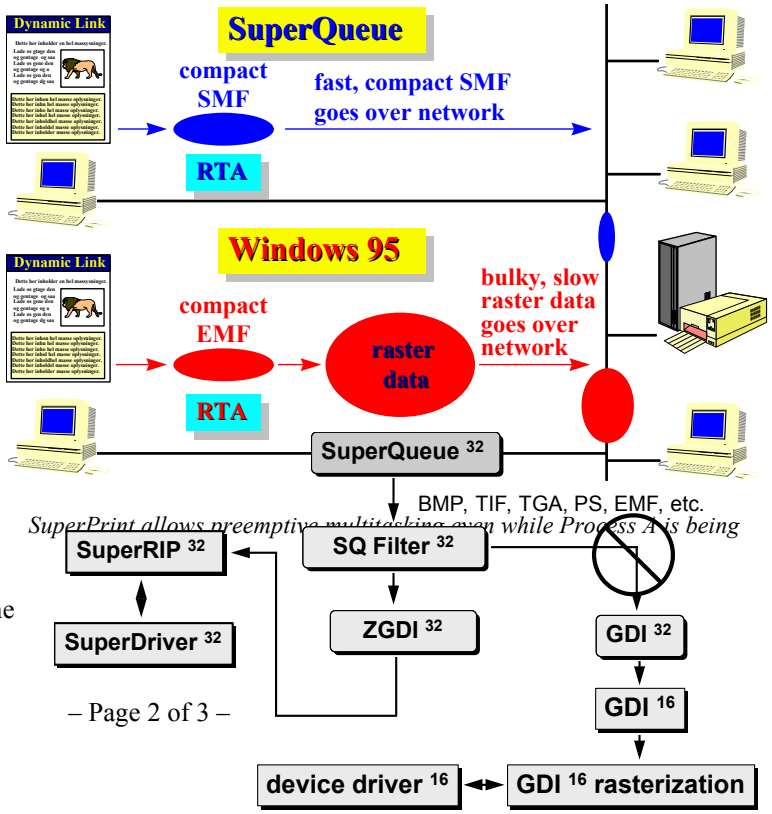
Background Printing with *SuperPrint*

Like Windows Spool 32, *SuperPrint*'s new 32-bit *SuperQueue* background printing utility encapsulates GDI calls in a SuperMetafile (SMF), then returns control of the PC to the user. Unlike Spool 32's EMF's, *SuperPrint* SMF's go directly to *SuperPrint*'s rasterizer with no mutex setting. Once again, the process is very efficient, and multi-tasking is very smooth.

Note: EMF's are compatible with *SuperPrint*'s architecture, so a user can handle Microsoft EMF's through *SuperPrint* as efficiently as it handles *SuperPrint* SMF's!

Unlike Windows 95's EMF's, *SuperPrint*'s SMF's can be sent over a network. Since Windows 95 isn't able to transport an EMF over the network, it must be "un-encapsulated" and rasterized prior to being sent to the network. That means 1) client CPU time is taken away from multi-tasking other important jobs, and 2) a lot more data (raw printer data) is sent over the network.

When using Windows 95 with *SuperPrint*, the compact SMF can be sent right to the printer server for processing. That means 1) no rasterization overhead on the client processor, and 2) efficient network traffic.



SuperPrint allows preemptive multitasking even while Process A is being

ZGDI: Perfecting the Solution

Zenographics has developed an "alternate GDI 32" called ZGDI 32 that understands the calls that GDI 32 normally processes for certain file types, such as BMP, TIFF, TGA

and PostScript. (EMF support will be provided shortly in a free maintenance release, which will allow any Windows document to be processed this way). This allows *SuperPrint* 4.0 to bypass Win 95 GDI altogether.

Bypassing GDI 16 allows *SuperPrint* to break through the 32,000 scanline limitation imposed by Windows GDI, which limits large format printing to 80" at 400 dpi, and limits imagesetter printing at 2400 dpi to 13-14". With *SuperPrint*'s ZGDI, print length and resolution are virtually unlimited!

SuperPrint 4 enables true 32-bit processing through use of its own GDI.

Advantages of *SuperPrint*'s Multi-Threading

Some printers (e.g. wide format printers, high-resolution film recorders, continuous tone color laser copiers) require large amounts of rasterization or "Ripping (from Raster Image Processor). If there isn't sufficient PC memory, this process has to be broken down into pieces or "bands." With Windows 95 and GDI 16, rasterizing and sending data must occur in a serial fashion: This is because there's no ability to launch a new thread in a 16-bit environment until a given process has been completed.

Printing with a conventional driver architecture	**Printing with *SuperPrint* 4**
Step 1: Rip band 1 ↓	Step 1: Rip band 1 ↓
Step 2: Send band 1 ↓	Step 2: Send band 1, Rip band 2 ↓
Step 3: Rip band 2 ↓	Step 3: Send band 2, Rip band 3 ↓
Step 4: Send band 2 ↓	Step 4: Send band 3, Rip band 4
Step 5: Rip band 3 ↓	
Step 6: Send band 3…	

SuperPrint 4 improves CPU utilization, for faster throughput of documents.

In contrast, *SuperPrint* 95 can rasterize a band, and then, while it is being sent, another thread can be started to send the band. Ripping and sending data can take place concurrently, significantly reducing overall print time.

Summary

Windows 95 will continue to have a "hybrid" 16-bit/32-bit architecture for the foreseeable future. End users are upgrading to Windows 95 for two primary reasons: to utilize the new interface, and to get increased multitasking performance in the new "32-bit environment." Users will experience a lack of smooth, efficient multi-tasking while printing from Windows 95. *SuperPrint* 4.0 provides the missing link, replacing the 16-bit print process with a true 32-bit multitasking print system.

For more information, please contact:

Scott Mackay, Product Manager
Zenographics
34 Executive Park
Irvine, CA 92614
(800) 566-7468 ext. 289
scottm@zeno.com

WWW: http://www.zeno.com
CompuServe & MSN: GO ZENO