Why We Still Need OPI

by James E. Porter

Back in 1989, when the open prepress interface (OPI) specification was initially developed, it addressed a huge bottleneck that occurred in digital prepress workflows because image files, especially color, are very large. Then, the networks connecting page layout workstations to servers and output devices were limited, able to yield only a portion of the theoretical 10 Mbps throughput. OPI, an extension to the PostScript language permitted applications or users to insert a set of comments describing the location, size, and cropping of an image in a page layout, rather than the image itself. Using this low-resolution "proxy" or FPO file in place of the high-resolution master in page layouts significantly reduced the total file size, preserved precious bandwidth, and increased overall throughput.

But that was then and this is now. Today it is common to find well-designed networks using switches and 100Base-T/Fast Ethernet networks to connect G3 Power Macintosh workstations to UNIX and Windows NT servers. And Gigabit Ethernet is beginning to enter the realm of economic feasibility for the publishing and printing industries. So why do we still need OPI?

This article delves into some of the reasons that OPI as a methodology, including APR and DCS, is still an important tool for maintaining a productive prepress operation. It also explains how OPI can facilitate the conversion from PostScript to PDF workflows that is now beginning.

How OPI Works

Irrespective of the vendor's system or operating platforms, OPI is a relatively straightforward process. After an image is scanned or captured at high resolution, it is saved to a file server or image server. Using hot folders and polling, or events supplied by the operating system to initiate the process, the OPI application subsamples the image to create a low-resolution proxy. The low-res FPO sample is usually saved in TIFF or EPS format, depending on the high-resolution image, with the FPO version saved at 72 dpi to match the resolution of most desktop monitors.

Since the OPI sample generator can recognize multiple file formats, it can easily create low-resolution FPOs of images that may otherwise be unreadable by the page-makeup application. Some applications, for example, are unable to read the Scitex Handshake LW format or to process TIFF images compressed with certain flavors of CCITT compression.

The most commonly used file formats include TIFF, EPS, DCS v1 and v2, Scitex Handshake CT, and JPEG (these should be the minimum formats supported by any OPI server). Additional formats include Scitex Handshake LW, CopyDot TIFFs, and Photoshop Native.

To facilitate page makeup, designers accesses the FPO from the server, place it—including cropping, scaling, and rotation—then save the final piece. When the file is output to a PostScript device, two things happen. First, the OPI producer, as the page layout application is known, writes a set of comments according to the OPI specification (see "A Little History" sidebar, next page). The comments describe the FPO image's position and characteristics (cropping, scaling, rotation) on the page and also the name and search path that will locate the high-resolution image file on the server. Next, the OPI image server, known as an OPI consumer, performs the image swap and sends on the "fat" PostScript file to the RIP for processing and output. Depending on the system, the high-res image file can be (1) separated before being saved on the server, (2) separated by the OPI application on the server, or (3) saved in a composite format for separation by the RIP.

Workflow Enhancement

No matter how fast networks get, OPI still performs several important roles. The first is strictly a matter of optimizing capacity use: there is no reason to burden either the network or the page layout workstation with extraneous data. The performance of both declines as the file size increases—exponentially with Ethernet-based networks. (That is, if the file size increases by a factor of two, the performance declines by a factor of four.) As for the page layout workstation, using the high-resolution image data significantly slows down the import

Example

Aldus 1.3 Comments

```
%ALDImageFileName:
  OPIData:Ads:Bottles.tif
%ALDImageDimensions: 712 480
%ALDImageCropRect: 0 0 712 480
%ALDImagePosition: 0.000 0.000
  0.000 480.000 712.000
  480.000 712.000 0.000
%ALDImageResolution: 200.000
  200.000
%%BeginObject: image
%%EndObject
```

process. For example, a 30 MB image will require five seconds to "open" in QuarkXPress at full resolution using Fast Ethernet, but the proxy will open almost instantaneously. If the OPI system runs under Windows NT and there are multiple concurrent users, performance degrades further.

OPI also facilitates efficiency in workflows that re-use the same image repeatedly, a common tactic used in the catalog industry to increase response rates. Importing a high-res image multiple times needlessly taxes the server, the network, and the layout workstation. OPI clearly minimizes the load, especially on the page layout workstation because there is less data to manipulate and print.

Given that publishing is an iterative process that usually requires multiple rounds of proofing before final output, OPI plays a major role in cycle time reduction. When printing to a PostScript proofing device, the layout application must first convert data from native format to PostScript and then spool it to the RIP for processing. When color raster data is included in the data stream there is a considerable delay, during which the workstation and its operator are completely idle. However, by using the various "Omit TIFF" and "Omit TIFF+EPS" options provided by the page-makeup application, the page can be printed in a matter of seconds. When using these options, the page-makeup application replaces the image data it would have printed with a set of PostScript comments (see example Aldus comments).

OPI also enables better data management practices. Storing all images on the server makes it possible to implement more efficient and more secure workflows. In addition to minimizing data traffic, storing the high-res images on a centralized server provides an easy mechanism for logging images into an image database according to predefined methodologies. This enables everyone—including operators and clients—to access images more easily. The images will always be cataloged according to predefined rules, avoiding the one file/multiple names and many files/same name problems that are possible when images are not centrally stored and cataloged.

Obviously, storing images on the server offers better security than does a distributed approach. First, data can be regularly backed up using popular applications such as Retrospect, BackUp Exec, or Legato. Further, storing high-resolution images on the server minimizes the chance that an image will be altered or deleted—either accidentally or intentionally. For the same reasons, this practice facilitates revision control when images undergo heavy retouching to create multiple derivative images.

Finally, OPI is the most efficient way to provide clients access to their images over the Internet and other remote-access systems. For the same reasons, it streamlines workflow at the client's site and when clients submit files for processing and final output. While prepress shops have built heavy-duty networks, many designers and publishers are still working on 10Base-T and cannot efficiently handling large image files. OPI also streamlines the submission process when clients return a file for final output. For those using ISDN or private networks, the cost to transfer a file is directly proportional to its size, so it is faster and cheaper to use OPI.

Transitional Value

With the printing industry now undergoing a transition to "direct-to" technologies such as computer-to-

A Little History...

OPI is an extension to the PostScript language that minimizes file size by permitting an application or user to insert a set of comments describing the location, size, and cropping of an image in a page layout, rather than the image itself. In addition to spatial issues, OPI comments can also describe the brightness and contrast of an image.

The specification was initially developed by the Aldus Corporation as a way to cope with large image files, and to enable PostScript-based desktop publishing applications such as PageMaker to work with CEPS such as Crosfield, Hell, Scitex, and Screen.

After Aldus was acquired by Adobe Systems, both the TIFF and OPI specs became the intellectual property of Adobe, which continued to publish the specification. Following its debut in 1989, the specification was upgraded to v1.3 in September of 1993. A draft of the 2.0 spec was released in August of 1995, though it remains in "draft" with the caveat that "it may change significantly before its final release." Given its lineage, the 1.2 and 1.3 specifications supported only TIFF (tag image file format) images. However, since different applications generated different OPI comments that resulted in incorrect or no image substitution at all, Adobe revised the 2.0 specification to support EPS files.

Of course, OPI isn't the only image replacement system/methodology. Both Scitex and Quark developed their own tools to expedite production. Scitex developed APR (automatic picture replacement) to enable designers to perform page makeup while enabling its high-end workstations to focus on image retouching. For its part, Quark developed the DCS (desktop color separation) specification, which is a derivative of EPS, to complement XPress workflows. The first release, DCS v1.0, required that images be separated into the process colors and saved along with a low-res composite file, which gave rise to its nickname as "five-file EPS." With DCS v2.0, separated files can be saved in either a single file or multiple files; in addition, v2.0 supports spot colors and additional separations for applications such as HiFi color.

plate and, soon, direct-to-press printing, OPI offers considerable value, especially in for publication printing and newspapers. The reason is the prevalence of TIFF/IT files that are the current standard for digital advertising materials. TIFF/IT files are huge because they are both separated and screened to satisfy final output requirements, just like supplied films. A single-page, four-color magazine ad saved as a TIFF/IT final page format can run up to 6 GB when it is decompressed by the printer.

Transferring many of these files will overwhelm even Gigabit Ethernet networks. However, by maintaining the TIFF/IT-P1 files on the image server and using an OPI workflow with FPO proxies, the imposition file can be kept to a much smaller size to expedite final output. In such a workflow, the server could be connected to the RIP via a wide-bandwidth connection such as fibre-channel arbitrated loop to deliver sufficient throughput. Alternatively, the OPI could be integrated with the RIP on a single server/workstation to completely avoid moving uncompressed files across the network.

In addition, OPI enables printers to integrate PDF, Adobe's emerging file format, with existing PostScript workflows. Given that the majority of installed RIPs are PostScript Level 2 or earlier and cannot process native PDFs, OPI provides the means to handle them. In such a workflow, the OPI system automatically converts the PDF to an EPS file, then generates an EPS low-res proxy file. Following that, a normal OPI workflow is used to place it in the page or signature.

Using Adobe's new page-makeup application, InDesign, graphic designers can place PDF graphics as well as EPS and TIFF files into their page layouts. Compatibility problems with legacy PostScript devices still remain, but can be overcome by OPI. Using the workflow mentioned before, the PDFs can be converted to EPS and, using OPI, omitted from the workflow in the same manner applications such as PageMaker and QuarkXPress use to facilitate output to a PostScript printer.

The Ideal OPI System

To ensure efficient and secure workflows, the ideal OPI server should satisfy several qualities and capabilities. First, it should be smart—able to recognize different files types. Many images are saved in legacy formats such as Scitex Handshake LW that desktop applications cannot recognize, but the OPI server can automatically recognize them and generate low-res files in a compatible format such as TIFF or EPS.

Further, while most prepress work is performed on the Macintosh platform, PCs are gaining a foothold. PCs are very common in corporate marketing departments, which are increasingly performing work in-house. As a result, the OPI should be capable of generating a low-res FPO that is compatible with both Macintosh and PC platforms. In a cross-platform environment, the OPI system can generate a single FPO containing two distinct screen previews, one for each platform.

The OPI system should be efficient. When using EPS images in a non-OPI system, one of the issues is that the EPS file contains information about all the separations. This approach requires sending the entire image file with each separation during output. For example, as a four-color image is transferred to the RIP, a 10 MB EPS will accompany each separation. This means transferring a total 40 MB of image data. An intelligent OPI system, on the other hand, can recognize an EPS image that can be separated—such as one from Adobe Photoshop—and minimize transfer time and maximize productivity by sending only the appropriate separation data with the page data. In the

APR
automatic picture replacement

CCITT
Comite Consultatif International de Telephonique et Telegraphique
one of four organizations comprising the International Telecommunications Union (ICU) that develops standards for data communications

CEPS
color electronic prepress system
usually refers to high-end proprietary systems

CT: continuous tone

DCS: desktop color separation
pre-separated four-color files include five image components (C,M,Y,K and a low-res composite for placement)

EPS: encapsulated PostScript
vector-based representation of a graphic or page which also includes a low-res bitmap representation for on-screen viewing

JPEG: Joint Photographic Experts Group
an ISO committee that has defined standards for the digital compression of graphical images; a data compression scheme and a file format

LW: line work

OPI: open prepress interface

RIP: raster image processor

TIFF: tag image file format
probably the most common format for exchanging graphic, bitmapped files

TIFF/IT: tag image file format for image technology
ISO 12639 data exchange standard for saving and transporting CEPS bitmap images; the TIFF/IT P1 variant is used for transferring high-quality composite color images to printers without the need to RIP the file at the target site

Definitions adapted from The GATF Encyclopedia of Graphic Communications, GATF*Press, 1998, and from GRACoL Version 3.0, Graphic Communications Association, 1999.*

cited example, only 2.5 MB of Photoshop EPS data would accompany the cyan plate as it goes to the RIP.

Graphic designers almost always crop images to suit the page layout, leaving extraneous image data that will not print. An efficient OPI system will expedite output by automatically discarding the nonvisible data before outputting to the RIP. Say that a designer uses only a 1x1-inch fraction of 4x4-inch TIFF that is about 16 MB. With an OPI system, the RIP will have to process only the 1 MB of image data that is visible in the picture box and not the entire 16 MB.

Proofing is another area where a well-designed OPI system can improve the entire workflow. The most popular inkjet proofing systems, such as Iris, output at 600 dpi resolution, and many more operate at 300 dpi; any additional data is purely overhead. The OPI system, however, can be configured to subsample the data so that the proofing device receives only the appropriate resolution to expedite both transfer and RIPing times. For monochrome output devices, the OPI server can also be configured to convert color data into black and white before sending it to the proofer, which reduces the amount of data transferred by up to 75%.

For similar reasons, designers who want to take advantage of PDF for remote proofing can also benefit from OPI. If a page is output using only the 72-dpi FPO, the result is a poor-quality image with numerous "white holes" that can be disconcerting to the viewer. Using OPI, however, makes it possible to include composite printable data so that when the PostScript is distilled into PDF, the image can be viewed and printed with low-resolution images to generate a better quality proof.

An OPI system should be secure. The most often heard complaint about OPI is that the link between the high-resolution master and the FPO gets broken, usually because the file was renamed or moved to a different volume on the server. OPI system vendors have addressed this in different ways. At IPTech, for example, CanOPI uses an event-driven approach to generate the low-res FPO to ensure that the link remains intact. Making every action related to an image—copying, saving, deleting, renaming, moving, etc.—an event that is tracked by the server, automatically updates the FPO to ensure that the image outputs correctly. This approach is usually associated with UNIX-based OPI solutions such as Helios Ethershare OPI or Xinet's Full Press. NT solutions typically employ hot folder polling.

Finding images with an OPI system that uses a polling-based hot folder approach, or performing image replacement using another OPI vendor's samples, such as Scitex APR, that do not always have full path qualifiers available can be more challenging. Here, users must specify search paths—either a number of individual folders to search or a folder hierarchy to search recursively. The danger of using a recursive search is apparent to anyone who has ever tried it. If we encounter a different image with the same name, we may get the wrong output. Equally problematic is a potentially infinite inclusion loop, which can occur if the first image we find is the low-resolution itself. To prevent this problem, some OPI systems enable users to specify a "maximum rec ursion level."

Finally, OPI should be flexible. Some OPI systems offer several options for organizing the storage of high-resolution and low-resolution image files, for example, creating a subfolder relative to the high-resolution image, creating the same folder hierarchy as the high-res but under a different base folder, or placing all samples in the same folder. All elements for a particular job need to be collected in one folder (all low-res samples, fonts and of course the page itself). To meet this need, one system allows files not recognized as a valid image format to be moved into the same folder as the samples. Another vendor solves this problem by generating the FPO in the original folder and moving the high-res to a subfolder.

Conclusion

In the final analysis, we will continue to need OPI as long as we continue to rely on networks for publishing. Although technology continues to improve, delivering increased bandwidth, faster processors and less expensive memory and storage, it seems that our thirst grows faster. While color was once used sparingly, with one or two relatively small images for any given page, today it is common to find more and bigger images used to gain the reader's attention in a media-intensive world. And, while HiFi color has not been adopted as quickly as some had predicted, direct-to-plate and direct-to-press are likely to stimulate increased use of this six-color technique—which means two or more additional separations of image data to process. Our insatiable appetite for more color and more complex designs will continue to offset Moore's Law, and we will still need OPI.

James E. Porter is a senior software development engineer for Information Presentation Technologies, San Luis Obispo, California. Prior to joining IPTech in 1995, he spent five years with Hyphen Ltd. and Hyphen Inc. developing the Spectraserver and other products. Porter holds a BSC Honors Degree in computer science from the University of East Anglia, Norwich, England.

This article will be reprinted in the near future as SecondSight No. 77 (GATF Cat. No. SS77).
To order now, call 800-662-3916 or 412-741-5733. Fax: 412-741-0609; email: gatforders@abdintl.com.