Issues on the Printer Driver Extension Proposal
Hugo Parra; 5-9-00

I've received some good feedback on the "Internet Printing Protocol (IPP): Printer Driver Extension" document. I've summarized the issues raised along with some of the discussion that's taken place around those issues. Hopefully we can discuss and close a few of these at the NY meeting.

ISSUE 1: Does IPP really need this functionality?

The proposal addresses two separate, though closely related, capabilities:

1) The ability for a printer to furnish the information workstations need to select and locate the necessary files (printer drivers, PPDs, etc.) to "install" the printer on the workstation. In my opinion, this is an absolute need in IPP. Our customers rank the ability to remotely manage workstation printer installation and printer driver updates (which the proposed IPP capability would enable) as number one, above fancy notification, advanced job submission options, extended printer and job control capabilities, etc.
2) The ability for a printer to furnish the files needed for installation via IPP. Being able to download the installation files from the printer would provide one standard way to obtaining the files and would eliminate many of the common error conditions encountered today. It would truly enable Internet printing by not forcing Internet users to submit documents in PDF format and not forcing the administrators of the printer to provide a separate mechanism (protocol, firewall hole, etc.) for installation files downloading. The document proposes this as an OPTIONAL capability in IPP.

ISSUE 2: Security

In some scenarios, the installation files include executable binaries. Downloading executables from the network is a delicate matter and should be protected by reliable client-side authentication. TLS should be required of implementations of this optional functionality, especially when executable files are involved.

ISSUE 3: Isn't this functionality redundant with the current IPP model?

The IPP spec defines "printer-driver-installer" (uri). The single uri is supposed to point to a *driver installer* for the printer object. This is really a punt on the whole problem because nowhere is *driver installer* even vaguely described. The assumption is that the workstation would negotiate with the driver installer using some unspecified protocol the correct printer driver to use for the printer's make and model to match the workstation's OS, natural language, and preferred PDL. Implementations that interoperate in this area give the current model are highly unlikely.

ISSUE 4: Collections

The proposal uses the Collection syntax in two instances. As collections haven't been accepted by the IPP WG at large, its use could inhibit the adoption of the proposed extension. I would like to propose the following two alternatives:

a) Define a new 'urx-supported'-like syntax
b) Define a new 'resolution'-like syntax

Also …
c) It's been proposed that we use the 'resource container' object. Resource containers are a new idea a few members have discussed as a general-purpose extension to the IPP model. See separate message from Tom Hastings for more information on this option.

ISSUE 5: Attribute size limits (1024, 64k)

The original proposal calls for the installation files to be returned as a field of a collection attribute. It requires that an exception to the 1024-byte attribute value and 64k-byte attribute length limit be made. This change is significant and would probably require a version number upgrade.

To avoid this, I proposed the following modification:

Instead of returning a printer driver as an attribute value, "Get-Printer-Attributes" returns it as post "end-of-attributes-tag" data in similar fashion to the way "Print-Job" sends the document content.

To accommodate this change, the last field of "printer-driver-supported-coll" is modified as follows:

From: "binary-file" (octetString) /* contains the printer driver */
To: "binary-file-returned" (boolean) /* indicates whether a printer driver is being returned after the "end-of-attributes-tag" */

Just as previously proposed, this data is returned by the printer only when the workstation explicitly requests it and only for the first entry that matches the qualifications specified in "printer-driver-info-request".

ISSUE 6: Storage requirement

Requiring printers to embed the installation files for all the client OSes/ document formats/ natural languages they support is unreasonable. Printers are not required to embed any installation files. They may include one or more or they may refer workstations to other sources where those files may be found.

ISSUE 7: Provide cross-platform support

The proposal should accommodate installation paradigms others than the Windows printer driver installation model. The intent of the proposal is to enable printers to furnish the information and, optionally, the installation files needed to install a printer on any platform. The 'binary-file' field can refer to PPD or UPDF files or any other type of file needed to complete the installation on a given platform. To highlight this goal the following changes might be appropriate:

a) Change title of document from "Printer Driver Extension" to "Printer Installation Extension"
b) Rename the 'printer-driver-uri' field to 'installation-files-uri'
c) Rename the 'binary-file-returned' field to 'installation-files-returned'
d) Add the 'installation-file-type' field to "printer-driver-supported-coll" with supported values: 'print-driver', 'ppd', 'updf', etc.

ISSUE 8: Standardize on PostScript and PPDs so no executable binaries need to be downloaded.

Even though currently most network printers support PostScript, in Windows, specific PostScript drivers are still needed to access the full capabilities of their corresponding printers. In many cases, using a generic PostScript driver with the appropriate PPD file results in unacceptable degradation of functionality.