Universal Printer Driver (UPD) Meeting
July 7, 1998
Monterey, CA
Attendees

Bob Taylor	HP
Mike Whitmarsh	HP
Brian Nagy	Kodak
Rick Yardumian	Xerox
Don Wright	Lexmark
Ken Oakeson	HP
Frank Zhao	Panasonic
Lee Farrell	Canon
Yazi Sazchi	Japan Computer Industry
Stuart Rowley	Kyocera
Laurie Lasslo	HP
Peter Zehler	Xerox
Atsushi Uchino	Epson
Brian Batchelder	HP
Robert Herriot	Sun
Harry Lewis	IBM
Alan Berkman	HP
Sandra Matts	HP

Meeting agenda

- Process
- Requirements
- Look at GPD / PPD architecture

Process

UPD is in stage one of the PWG process: Creating a charter, requirements document.
Slides from meeting are at ftp://ftp.pwg.org/pub/pwg/upd/minutes/slid9807.ppt
Sandra stated that the major goal of the group is “to easily add support for multiple printers and different device representations without giving up complete control.” She highlighted three characteristics:

1. Language Independent (“PDL agnostic”)
2. Platform Independent
3. Color space

Sandra suggested a process for creating the UPD:

1. Define requirements concisely
2. Identify high level concepts and possible general solutions
3. Try not to bind to a specific technology (yet)

A few items were mentioned regarding the requirement for color support:

* Win 98 ICM supports sRGB and ICC based color spaces
* Win 95 and Win NT4 ???
* Unix ???
* VML: sRGB
* PGML: sRGB and ICC based color spaces
* VML and PGML will merge (hopefully) - W3C has a Scaleable Vector Graphics working group planning to meet in August to attempt to generate a combined proposal

Sandra proposes to use sRGB and ICC based color spaces. She does not know if the ICC profile is optional or required.

According to Sandra, the "monolithic model" of drivers that is currently being used by printer vendors is not easily extensible. Adding a printer can require a new driver to be developed. Another weakness is that this model is both platform and printer language specific.

Paul Moore (Microsoft) has proposed the "PPD/GPD" model in the past. It contains a text file that describes many of the characteristics of the printer. A base driver, user interface and rendering engine is provided by the operating system. Callback functions can replace (or augment) the core rendering capabilities. The user interface can also be replaced or modified.

PPD/GPD Pros:
*	Can add support for multiple printers easily
*	User interface is customizable
*	Callback functions can be used for rendering

PPD/GPD Cons:
*	PS and PCL specific; can't force low-level raster or new language definition
*	Callbacks are Windows-specific
*	Vector support does not exist (nor is it possible?)

Proposal 1: take the PPD/GPD model and "stretch" it to provide dynamic and extensible capability
*	Include dynamic attributes query device
*	Include platform independent callbacks for rendering (perhaps use IDL, similar to CORBA and COM?)
*	Include custom user interface dynamic attributes

Proposal 2: "stretch it even more" than Proposal 1:
*	Create an intermediate "printer language independent marking definition" (PLIMD) that can be translated for each language, or defaulted to a "null" translation
(The null translation would amount to having created a new printer language.)

Pros of Proposal 2:
*	handles any printer language
*	extensible - add a new language translator component for a new language
*	platform independent
*	can easily add support for new printer languages

Cons of Proposal 2:
*	extra layer of translation - good for abstracting, but could create a performance hit
*	might not be possible to create a PLIMD in a reasonable timeframe (or at all)
	*	Adobe will provide a PS- and PDF-to-PGML translator
	*	Printing a PGML document will produce a device language output

Sandra will contact Paul Moore, and request him to provide unrestricted access to the GPD material that is shipped in the Microsoft Beta DDK.
Ideally, it would be made available in an editable format (e.g. Word.)

Don Wright: "We would like there to be a UPD of some type, but we still
want the ability to write a 'monolithic driver'."

Several people commented that the effort to develop a PLIMD would be very large -- possibly prohibitively large.

As a compromise, it was suggested that we first concentrate on Proposal 1, addressing the details of extending the core rendering capabilities last.

Requirements (Need to prioritize in categories)

1. Localizable
2. Platform independent
   2.1. Modules written for one Operating system will work on other operating systems
3. Printer Device Language (PDL) independent
   3.1. It was discussed that PDL independent is maybe too strong a term. Suggestion that language agnostic is more applicable.
4. Color management system
   4.1. Must be able to integrate with the color management system of the OS.
5. Device independent
6. Extensible
   6.1. Be able to support multiple printers without creating a separate monolithic driver for each printer.
   6.2. Where are the customization dlls?
   6.3. How are they installed?
   6.4. What is the process for the discovery of these libraries?
   6.5. How to deliver / install platform independent callbacks?
7. Protocol independent
   7.1. Not rely on a specific protocol for transmission of GPD and extensions.
8. Overlays, Job Tickets, etc.
9. Fonts
   9.1. Font Formats
   9.2. Device font support
   9.3. Unicode and character set support. Character encodings.
   9.4. Font Substitution
10. Bidirectional and Unidirectional
11. When has GPD′ changed? Checksum or magic number or tag. (I will use GPD′ until we can invent our own acronym for the UPD printer description file.)
    11.1. Support dynamic changes to device configuration.

Items of discussion

Versioning and extending the GPD′. Have to develop a mechanism or process for versioning of the GPD′ specification.

UI Feature constraints: Developers noted that the current Adobe driver feature constraints are not complex enough to allow all of the real world UI constraints.

UPD – Universal Printer Driver. A UPD consists of several components. The static section has three components.

1.) A GPD file
2.) A core rendering component
3.) User Interface

The dynamic or extensible part contains:

1.) callbacks for rendering
2.) callbacks for UI

Use Cases for UPD

Traditional

- GPD and driver are on disk.
- Get GPD from web site.
- Install at PC or server
- GPD/driver is in the OS and installed through install program.

Pull GPD from somewhere else

- Get GPD from the printer
- Get GPD from a URL
- Get GPD from a server

Print Pool

- Lock GPD down so all printers have same features (disable dynamic update)

Action items:
Get the GPD specification. Not everyone has access to this document since it is MS confidential. The GPD spec is available on the Jan 98 MSDN Professional Subscription. The version is from the Beta 1 DDK and so it is not the most current. Also the Spec is in an html help format which is not the best format for printing and reading. We are hoping for a Word editable version of the GPD Specification. Once Beta 2 ships in August the Beta 2 DDK will be put onto www.microsoft.com/hwdev in the Developer Kits area. There's a placeholder today but no DDK yet.

ICC Specification is at www.color.org. There is a link to the spec in PS and PDF formats. Actual pdf file is at ftp://sgigate.sgi.com/pub/icc/icc34.pdf

PPD Specification is at
http://www.adobe.com/supportservice/devrelations/PDFS/TN/5003.PPD_Spec_v4.3.pdf

Charter must also include a glossary for terminology and definitions.