Universal Printer Driver
Study Group Meeting Minutes
March 3, 1998
Austin, Texas

The meeting was started at 7:10 PM and was chaired by Don Wright. These minutes were recorded by Peter Michalek and edited by Don Wright

Attendees were:

Don Wright – Lexmark
Brian Batchelder – HP
Mark VanderWiele – IBM
Mark Hamzy – IBM
Fumio Nagasaka – Epson
Fumio Samitsu – Epson
Mabry Dozier – QMS
Lloyd Young – Lexmark
Yoshinori Murakami – Epson
Greg LeClair – Epson
Lee Farrell – Canon
Akihiro Shimura – Canon
Takashi Isoda – Canon
Peter Michalek – Shinesoft
Bob Broccolo – Kodak
Carl-Uno Manros – Xerox
Harry Lewis – IB<
Bob Pentecost – HP
Ron Bergman – DataProducts
Praveen Kanipakam – Sharp
Scott Isaacson – Novell
Paul Moore - Microsoft

This was the second meeting of the UPD study group. Before we met in San Diego in May 97.

This subject came up in Maui and we agreed to meet again to explore the issues and decide whether to pursue this approach.

Agenda:
* why are we here?
* what do we want to accomplish
* Paul Moore will give an overview of UPD. Then an open discussion
* We should also define the UPD.

Tentative definition:
Collection of software (exe or dll) which takes graphic context created
by the user, changes it to a printer stream and causes an equivalent representation to be presented on a printer.

It has nothing to do with redirection, tcp/ip or ipx.

* Paul Moore: good description for printer driver but not in general.

* Printer driver needs to figure something out when talking to printer (therefore the previous definition would be insufficient.

* Does universality include across all platforms?

Don summarized this on the slide:

* convert text & graphics to something a printer understands
* UPD needs to be:
  - universal:
  - single piece of code that adapts to printer
  - how? We need to consider various dynamic aspects of this UPD:
    -- compile time vs. run time
    -- single platform or all platforms?

* localization/internationalization
  - support for languages

Carl-Uno Manros: not only image out of the driver but also attributes and possible something like a JOB ticket.

Other aspects of UPD:
data stream
commonality: transport, datastream, other aspects of
a printer job (page description)

How do you represent content without notion of presentation?

* size is important, but other presentation parameters are not.
* UPD is not specific to any transport or print protocol: i.e. not specific to IPP, LPR, etc.

* Should the driver definition contain embellishments like overlays?

* What similar attempts have there been in the past and what can we learn from them?

  - PPD's
  - miniddrv/unidrv GPD
  - JPrint Bristol: java print api - upd concept

* What are the capabilities?
* How much flexibility do we provide and when:
  - compile time?
  - install time?
  - print time?

* It was re-iterated that the localization is very important

* Does universal mean of all printers or all times or universal for the future?

What's the range of printers the UPD should support?

* Is the driver going to present a standard API?

* Should the UPD separate PDL and job attributes?

* Picture - flow diagram:
app -> upd -> printer
* What's the relationship of screen driver and printer driver?
* How do you handle color: which color space should we use?

* Extent of customizability/extensibility of driver

****

Section 2: Paul Moore: GPD presentation

This is an incomplete summary of the presentation. More details can be obtained from Paul Moore.

MS UPD
Aims:
Single executable that prints to any printer
Reduce time and effort to support a new printer
Higher Quality, better performance

NT4 has RASDD (v1 or unidrive)
nt5 has unidriver
All NT5:
PS
Unidrive > 1500 printer plotters
Single ww binary

MS addresses the following universality requirements:
- localization
- single binary

There were suggestions from the audience that some printers are not supported.

PaulM: unidriver doesn't support real-time rendering (or Host Based RIP), only raster-type of printers.

PaulM: The job of the printer driver is to:
- render the description:
- merge the job ticket information with the page description

{Don Wright}: job ticket and page information is processed by the driver to render the page

proposition - chart:
GPD app

unidrive

callbacks UI PDL

Callbacks executed in real-time
Unidrive uses GPD to generate rendering code.

UI is done programmatically.
UI to do with features is dynamically created.

** What's a GPD?

GPD is an ascii text file specific for the printer model. It describes:
features (bins, color depth)
options (duplex, bins,)
constraints
PDL generation instructions

** Standardization requirements
MS propose GPD syntax be adopted to support cross platform

What needs to be done:
- improve UI
- close the loop on features
- GPD and protocol should work hand in hand
- vector device support
- unification with PPD model

Question: How do I support different languages with UniDrive?
Answer: Provide a separate GPD for each market

Question: Dynamic feature addition?
Answer: A two issues:
dynamically declaring features (duplex capability)
model with unforeseen extensibility
- GPD should be able to describe these features

****

Section 3 Next Steps

Is this an area we want to explore?

It was noted that maybe we can't handle all printers with one binary.

Don Wright: GPD and PPD offer features that overlap, maybe we should explore how to unify them?

Schemas are unifying, it would be good to unify GPD and PPD as well.

PPD files are usually target independent, sometimes dependent - e.g. Adobe did PPD's specifically for Apple

Requirements input will be inquired on the mailing list.

Paul Moore: Microsoft will not assert intellectual rights on the GPD structure but would not make source code available to other OS vendors.

Paul Moore will look for the requirements that Microsoft developed for their GPD format and post to the mailing list if it is available.

The meeting adjourned at 9:50 PM.