IPP Driver Issue/Printing From Windows Application

I was thinking through how a user would print from an application. I talked it over with some people who are far more knowledgeable about the relationship between applications and printer drivers. Some of the concerns included:

- Current applications often load and unload the print driver multiple times before the print job is finally sent to the printer.
- Some applications have a great deal of knowledge about the printer drivers they use. Most adhere to the GDI standard to obtain this information. Any applications that go around this interface have dependencies on specific drivers.
- How is a printer located?
- How is a driver for the printer obtained?
- A lot of users do not know what a driver is or how to select the appropriate one.
- Lots of people (including me) don't know a printer driver is a collection of files.
- Some users are not aware of the environment they are using. They know how to turn on the machine. They need processes as automated as possible.

I have provided 2 scenarios. One to describe how IPP printing would function with current windows applications. I assume this can be easily extended to other client environments. The second to explore how IPP printing might work with IPP aware windows applications. I am unsure of how this would work in other client environments.

In the "current" scenario the application selects a preconfigured printer. Printer selection from the application is unidirectional. There is no negotiation for printer capabilities. Outside the application the printer has been bound to a printer driver. Once the application selects the printer the application will then query the associated driver to obtain printer metrics and capabilities. The printer/driver must first be installed outside the application. Then the application may select the IPP aware printer. The "current" scenario to cover this example follows.

In the future applications can be internet printing aware. Following the "current" example is the "Internet Aware" example. In this scenario the user will select a printer from within the application that is used to generate the document. The application will first ask the driver shell to select a printer. Once that is completed the printer will be installed and the driver bound. Then the printers metrics and capabilities will be assessed.

The "current" paradigm has the advantage of working with off the shelf applications. The printer installation utility can be automated as much as possible and be presented to the user as a platform specific wizard. The "Internet Aware" paradigm has the advantage of selecting the printer to meet the current documents requirements. My personal preference is the "current" paradigm. The management of printers is handled by the OS or a utility closely tied to it. Applications do not get involved in printer management. This solution can be rapidly prototyped.

Current (application not internet aware)

My understanding in this area is very limited. This is my understanding of printing from a windows application. The application makes a call down into the driver to determine the amount of memory required. The application will allocate the memory. The application will create a structure the driver can use to pass information about itself to the application. The application calls the driver to populate this structure. The application will use this information to provide WYSIWYG. When it is time to print the application gets a job handle from the OS. The application writes to this handle and the OS passes this on to the driver.

Internet printing in this paradigm requires the printer and its associated driver to be selected and installed prior to the application print. . An IPP printer selection and driver installer download utility would have to be used. This utility would have to be freely available and common for all print service providers. The functional equivalent of this utility could be incorporated into the operating system.

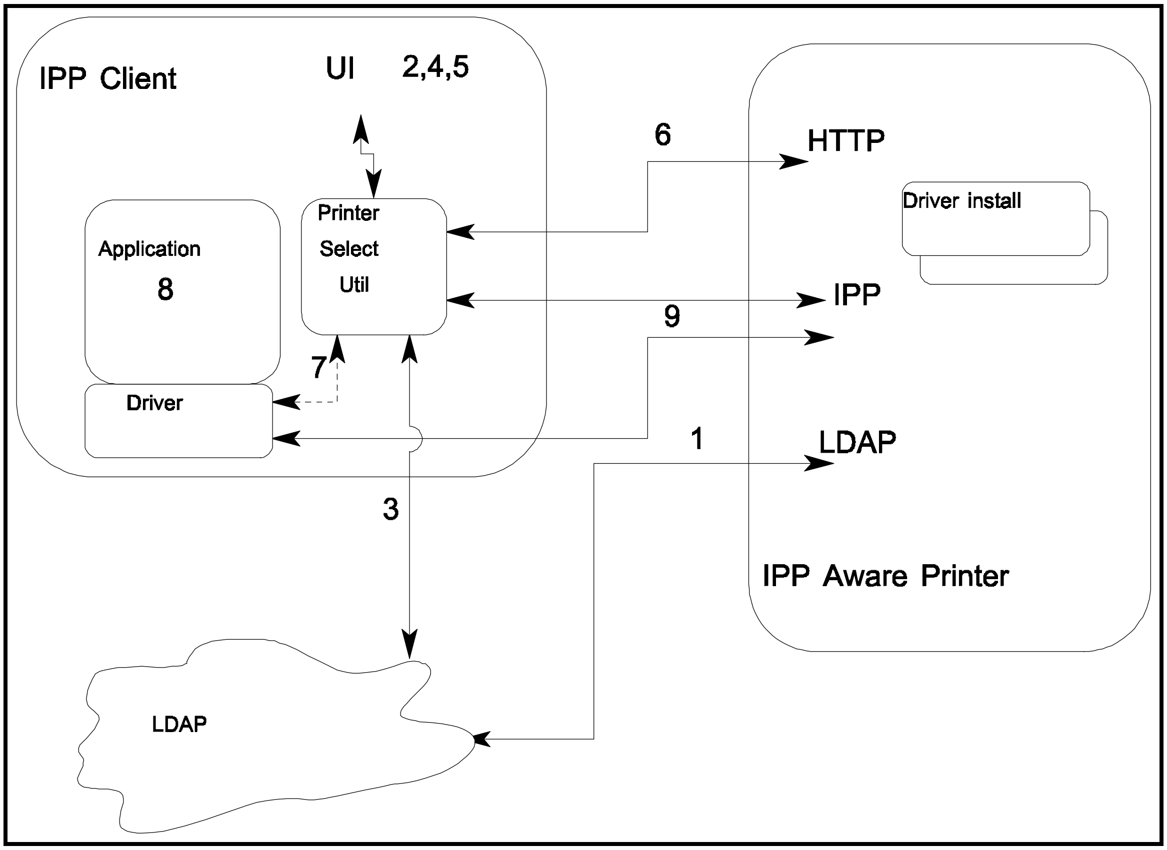

Fig 1.

Current (application not internet aware) IPP print from application

1) Printer would advertise itself using LDAP. An attribute of the LDAP printer entry would be the URL of the driver installer.
2) Utility would present UI to allow user to enter printer selection criteria. The selection criteria would be based on the printer LDAP schema.
3) Utility performs LDAP query to obtain appropriate printers.
4) User would be presented a list of printers.
5) User selects desired printer.
6) Utility obtains URL of driver installer resource from printer LDAP entry. Utility then uses Transparent Content Negotiation in HTTP to obtain the driver installer for the appropriate PDL and client environment. IPP printers do not have to store the driver installer locally. The LDAP driver installer attribute can contain the URL of some other network host.
7) Utility executes the driver installer to install the printer and the driver in the client environment.
8) User may now start application and select the printer.
9) Application interacts with driver that uses IPP to communicate with printer.

Internet aware (internet aware application)

This paradigm gives installation and selection of printers to the application user. This does not preclude the "current" paradigm. This does allow a user to search for a printer that meets the needs of his current job. This is how I believe printing would occur. An IPP driver shell would be installed on client. The IPP driver shell would be common for all IPP drivers. IPP drivers would snap into shell. The shell would pass application requests through to the driver. The application would now have to be aware that printer selection is a 2 part process. The first GDI calls to the shell would be to locate an appropriate printer. Once the printer is located, and the associated driver installer has been successfully run, then the normal interrogation of the driver for printer metrics could occur. Printing would continue as in the "current" paradigm. The exception to this is that the application would have to be aware that if a different target printer is selected the printer metrics would have to be reevaluated.

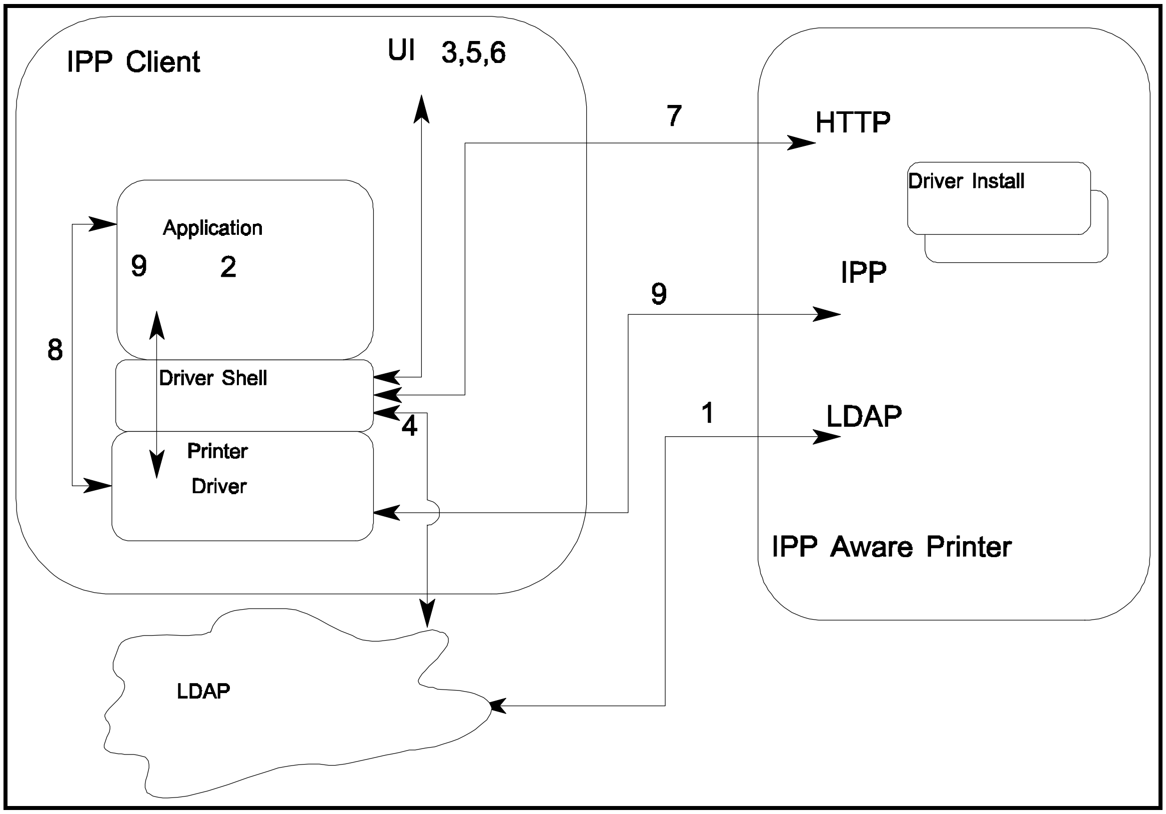

Fig 2.

Internet aware (internet aware application) print from application.

1) Printer would advertise itself using LDAP. An attribute of the LDAP printer entry would be the URL of the driver installer.
2) User starts application and selects the IPP printer(IPP driver shell).
3) IPP driver shell presents UI to allow user to enter printer selection criteria. The selection criteria would be based on the printer LDAP schema.
4) IPP driver shell performs LDAP query to obtain appropriate printers.
5) User would be presented a list of printers.
6) User selects desired printer.
7) IPP driver shell obtains URL of driver install resource from printer LDAP entry. IPP driver shell uses Transparent Content Negotiation in HTTP to obtain the driver install for the appropriate PDL and client environment. IPP printers do not have to store the driver locally. The LDAP driver install attribute can contain the URL of some other network host.
8) Driver installer is executed to install driver in a platform specific manner.
9) Application interacts with IPP driver shell that passes requests through to the driver that uses IPP to communicate with printer.