Sketch of an idea:
Define a Resource container object for Print Driver Extension and other sub-types

From: Tom Hastings
Date: 5/9/00
File: resource-object-idea.doc

We have discussed several times on the IPP telecons about adding a new Resource object. This mail note is a sketch of the idea to see if there is support for developing an IPP spec along these lines. The reason to bringing this idea up now is because of the discussion of the Print Driver Extension. The Resource object approach would allow a way to install print drivers on the Printer and to Get Print Drivers from the Printer to install on a client. It would allow a Printer to have more than one print driver, say one for each of a number of different OS platforms and/or PDLs.

I'd like to discuss the Resource container object at the New York City IPP WG Meeting, 5/17-18

Here is the sketch of the Resource object idea and its operations:

The Resource object is a generic container object for a number of sub-typed objects. The Resource object would have a number of operations defined. Each operation would include an operation attribute that identifies the object sub-type in question. Then implementers could define new sub-types easily without having to invent new operations.

There are some attributes common to all sub-types, such as the "resource-name", "resource-sub-type", "resource-owner", "resource-creation-date-time", etc., and a lot that are specific to a particular sub-type. Some sub-types will also have opaque data associated with each object instance (such as fonts, forms, images, and print drivers), and others will only have attributes (such as media).

The initial list of sub-types include (but many more are possible):

media - Just attributes which define the media characteristics. So the client can query the media library and the administrator can add new media and define their characteristics
fonts - attributes and the PDL data
forms - attributes and the PDL data
logos - attributes and the PDL data
images - attributes and the PDL data. high resolution images can be put into a shared image library and called out from a number of documents print drivers - attributes and the opaque data is the executable code

The operations would include:

1. **Create-Resource** - the client specifies the resource type and a user-friendly name for it that is used in all subsequent operations, including Job Submission operations (Print-Job, Create-Job, Send-Document). A lease time is also requested (just like our Subscription objects) and the Printer returns the lease time granted (which may be infinite meaning no need to renew or finite, meaning that the client must renew the lease, else the resource will be deleted).

2. **Delete-Resource** - delete the named resource of the indicated sub-type

3. **Renew-Resource** - renews the lease, in case it wasn't infinite

4. **Get-Resource-Attributes** - returns the requested (or all) attributes of a specified resource type and instance.

5. **Get-Resources** - returns the requested (or all) attributes of a specified resource type that matches the supplied filter criteria consisting of any of the resource attributes. Here the filtering is more complex than we have for Get-Jobs or Get-Subscriptions operations, since the client could filter on any of the resource attribute values. For example, a client could request all of the media that has a "media-size" equal to particular pair of x and y dimensions and get back the other resource attributes of any matched media instances.

6. **Get-Resource-Data** - returns the opaque data for those resources for which the data is not copyrighted, etc.

These operations are very similar to the Subscription object operations that the IPP WG has nearly approved for notification. The only differences are: Subscription Ids are assigned by the Printer as numbers, while resource object instances have user friendly names assigned by the creator. Both have leases, but Subscription leases are expected to be much shorter (minutes, hours) and handled by the software, not the user, while Resource objects leases are much longer (days, weeks, months), though both can be infinite depending on SA policy, and the user or administrator explicitly creates the Resources. Also Resources are known to users and administrators and are managed by them. Also Subscriptions don't have any complex filtering in the query requests.

[We don't want to merge Subscriptions into Resources, just illustrate their similarities]

It will also allow the administrator to set up policy on who can create which kinds of resources and for how long.

Comments?

Thanks,
Tom