DRAFT
ipp-sub-units-980407.pdf

S. Isaacson
Novell, Inc.
T. Hastings
Xerox Corporation
April 7, 1998

IPP Sub-Unit Objects
Version 0.02

Abstract

This document introduces new Sub-Unit objects into the IPP object model by defining a mapping from table entries in the Printer MIB to new object types and their associated attributes. A new Printer operation, Get-Sub-Units, is also introduced.

Table of Contents

1. Introduction

This document introduces new Sub-Unit objects into the IPP object model by defining a mapping from table entries in the Printer MIB to new object types and their associated attributes. A new Printer operation, Get-Sub-Units, is also introduced. This mapping allows for Printer MIB information to be queried (no setting or modification of information) using the IPP protocol rather than SNMP. This is not to say that the use of SNMP is invalid or improper, in fact just the opposite is true. The implementation of SNMP based Printer MIBs is almost universal for modern, network-attached printers. Therefore, since the information model as already been established and implemented, this document is simply a "IPP window" onto the Printer MIB. This mapping gives yet another way to access the same information that has already been validated and deployed using SNMP and Printer MIB technologies. It is expected that printer vendors that already support the Printer MIB and IPP/1.0 will find the implementation of the mapping defined in this document to be trivial with almost no additional overhead (i.e. additional memory or processing resources).

The basic approach for this mapping is to take all Printer MIB table entries, and define IPP object types for each. Each row in the Printer MIB's tables then become object instances of the associated type.

One exception is the Printer MIB's General Table. Each row in the Printer MIB's General Table already corresponds (roughly) to the IPP Printer object type. Since there is already an IPP Printer object type, instead of creating a new object type for entries in the General Table, almost all of the General Table columns (MIB objects) become new attributes for an IPP Printer object. Due to overlap with existing IPP Printer object attributes, some Printer MIB objects are not mapped to existing (not new) Printer object attributes.

The rest of the Printer MIB tables entries are implemented as Sub-Unit objects contained by an IPP Printer object (in the same way that the IPP Printer object contains IPP Job objects).

The following Printer MIB Tables are mapped to IPP object types:

Printer MIB Table Entry	**IPP Sub-Unit Object Type**
InputEntry	input
OutputEntry	output
MarkerEntry	marker
MarkerSuppliesEntry	marker-supplies
MarkerColorantEntry	marker-colorant
CoverEntry	cover
MediaPathEntry	media-path
ChannelEntry	channel
InterpreterEntry	interpreter

ConsoleDisplayBufferEntry	console-display-buffer
ConsoleLightEntry	console-light
AlertEntry	alert

The following Printer MIB table entries are not mapped as IPP object types:

LocalizationEntry: already covered by the natural language and charset IPP operation attributes)
StorageRefEntry: not applicable
DeviceRefEntry: (not applicable)

In addition, references to external tables (exteranal to the Printer MIB itself) are not mapped (such as the Host Resources MIB and the Interfaces MIB).

A new object type, not part of the existing Printer MIB or its tables is introduced. This Sub-Unit object type is called "medium". This object type is introduced to support the idea of medium characteristic attributes for both "ready" and "supported" media. The Printer MIB is only concerned with "ready" media (media loaded in one of the input trays).

2. Object Relationships

```
  +-------------------+
  |  Printer          |
  +--+----------------+
     |
     | +-------------------+
     +-| Input             |
     | +-------------------+  +-------------------+
     +------------------------| Interpreter       |
     | +-------------------+  +-------------------+
     +-| Output            |
     | +-------------------+  +-------------------+
     +------------------------| Console-Display-  |
     |                        |   Buffer          |
     | +-------------------+  +-------------------+
     +-| Marker            |
     | +-------------------+  +-------------------+
     +------------------------| Console-Light     |
     | +-------------------+  +-------------------+
     +-| Marker-Supplies   |
     | +-------------------+  +-------------------+
     +------------------------| Alert             |
     | +-------------------+  +-------------------+
     +-| Marker-Colorant   |
     | +-------------------+  +-------------------+
     +------------------------| Media-Path        |
     | +-------------------+  +-------------------+
     +-| Cover             |
     | +-------------------+  +-------------------+
     +------------------------| Channel           |
                              +-------------------+
```

Figure 1 illustrates the containment relationship. All IPP Sub-Unit objects are contained by an IPP Printer object.

3. Issues Resolved in the Mapping

Several issues must be reconciled when supporting the Printer MIB tables as IPP MIB objects contained by an IPP Printer object.

3.1 Terminology

IPP Object Type (or just Object Type): The IPP Model document uses this term to describe the attributes and operations associated with a modeled entity. In other OO contexts, the term "class" is used.

IPP Object: The instantiation of an IPP Object Type.

MIB Object: A column in a MIB table - maps to an attribute in the IPP model.

3.2 Name vs Index

The IPP model assumes that all contained objects will have names and some sort of identifier or index. The Printer MIB assumes that all rows in tables are indexed with an integer index. Therefore, every IPP Sub-Unit object type will have the following attributes:

prt-sub-unit-index
prt-sub-unit-name

3.3 Input objects and Media objects

IPP uses "media-ready" and "media-supported" attributes in order to identify what media is ready and supported, respectively. The Printer MIB does not contain the notion of a MediaTable. A separate descriptive object is useful for each media type when characteristics such as size, weight, color, grain, transparency, etc. need to be associated with the media. The Printer MIB chooses to associate these characteristics with the each row in the InputTable. The reasoning behind this decision is found in the Printer MIBs stated objective of only returning "ready" information (not "supported" type information).

To rationalize these two different models, the Medium object is introduced as a new Sub-Unit object and an Input object now contains a indexed reference to a the Medium object. The Medium object which has all the characteristics about that instances (size, color, weight, etc.). The set of "media- ready" is the set of Medium objects referenced by Input objects. In addition, the following Printer MIB objects are moved from the Input object to the Medium Object:

prtInputMediaWeight
prtInputMediaType
prtInputMediaColor
prtInputMediaFormParts

The following attributes are left as Input object attributes

prtInputMediaLoadTimout
prtInputNextIndex

3.4 Interpreter Object vs. Document Format

IPP uses an MIME media type to identify the "document-formats-supported" by a Printer and the format of any documents supplied to the Printer. The Printer MIB uses rows the Interpretor Table with the following MIB Objects:

prtInterpreterLangFamily
prtInterpreterLangLevel
prtInterpreterLangVersion

If an IPP Printer object supports both the "document-format" Job Template attribute and one or more contained Interpreter objects, then the implementation for that IPP Printer object MUST correctly map between the two forms of representing the PDLs that are suppored.

3.5 Scope Rules

Later on, a single IPP operation called "Get-Sub-Units " will be used to query ANY and/or ALL Sub-Unit objects contained by an IPP Printer object. This operation can be used to query:

1. All Sub-Unit objects.
2. All Sub-Unit objects of a certain type or types (all Input and Output objects)
3. A specific Sub-Unit object (Input object "tray 1")

3.6 MIB Object names vs IPP Attribute names

All MIB object names follow the ASN.1 naming convention of "prt" followed by the some string of multiple words or abbreviations with each word or abbreviation being all lower-case letters and except for an initial upper-case letter. For example: "prtInputMediaType". All IPP attribute names are special keywords (all lowercase with words being separated by a ‘='). For example: "document-format-supported".

Therefore, the following rules will apply in creating IPP attribute names from Printer MIB Object names:

1. The attribute name will begin with "prt-".
2. The rest of the words or abbreviations in the MIB object name (after the inital "prt") will be separated by a ‘-' and the initial upper case letter will be made a lower case letter.single

For example: "prtCoverStatus" becomes "prt-cover-status"

3.7 Mapping of SMI data types to IPP syntax types

The following rules apply:

1. ‘Interger32' and ‘Counter32' just map to ‘integer"
2. ‘OCTET STRING' generally maps to ‘text', howerver some may map to ‘name' and some may map to ‘octetString'
3. ‘Prt<whatever>TC' maps to ‘enum'
4. ‘presentOnOff' is just a special TC and it maps to ‘enum' as well

4. IPP Sub-Units

IPP Sub-Unit object types are defined in the same way as other IPP object types: for each type, all known attributes are listed and qualified.

It is OPTIONAL for an IPP Printer to support any Sub-Unit object. However, if the Printer object supports the functionality or physical characteristics corresponding the appropriate Sub-Unit object, then the Printer object SHOULD implement support for that (those) Sub-Unit Objects.

4.1 New Printer Attributes

The following attributes are added to the IPP Printer object:

```
   prtGeneralConfigChanges          Counter32
   prtGeneralCurrentOperator        OCTET STRING,
   prtGeneralServicePerson          OCTET STRING,
   prtInputDefaultIndex             Integer32,
   prtOutputDefaultIndex            Integer32,
   prtMarkerDefaultIndex            Integer32,
   prtMediaPathDefaultIndex         Integer32,
   prtConsoleLocalization           Integer32
   prtConsoleNumberOfDisplayLines   Integer32,
   prtConsoleNumberOfDisplayChars   Integer32,
   prtConsoleDisable                INTEGER,
   prtAuxiliarySheetStartupPage     PresentOnOff,
   prtAuxiliarySheetBannerPage      PresentOnOff,
   prtGeneralSerialNumber           OCTET STRING,
   prtAlertCriticalEvents           Counter32,
   prtAlertAllEvents                Counter32
```

These map to:

```
   prt-general-config-changes(integer(0:MAX)
   prt-general-current-operator(text)
   prt-general-service-person(text)
   ...
```

Due to overlap with existing IPP Printer object attributes, some Printer MIB objects are not mapped to existing (not new) Printer object attributes. the following objects from the General Table are not carried into the IPP mapping:

```
     prtGeneralCurrentLocalization:
         covered by the Printer object's natural language and
         charset attributes
     prtGeneralReset:
         this could be an operation, but is definitely not an IPP
         Printer attribute
     prtGeneralPrinterName:
         covered by the IPP Printer object's "printer-name" attribute
```

4.2 Console Display Buffer

```
    prtConsoleDisplayBufferIndex    Integer32,
    prtConsoleDisplayBufferText     OCTET STRING
```

4.3 Channel

```
    prtChannelIndex                     Integer32,
    prtChannelType                      PrtChannelTypeTC,
    prtChannelProtocolVersion           OCTET STRING,
    prtChannelCurrentJobCntlLangIndex   Integer32,
    prtChannelDefaultPageDescLangIndex  Integer32,
    prtChannelState                     PrtChannelStateTC,
    prtChannelIfIndex                   Integer32,
    prtChannelStatus                    PrtSubUnitStatusTC,
    prtChannelInformation               OCTET STRING
```

4.4 Cover

```
     prtCoverIndex           Integer32,
     prtCoverDescription     OCTET STRING,
     prtCoverStatus          PrtCoverStatusTC
```

4.5 Input

```
      prtInputIndex                      Integer32,
      prtInputType                       PrtInputTypeTC,
      prtInputDimUnit                    PrtMediaUnitTC,
      prtInputMediaDimFeedDirDeclared    Integer32,
      prtInputMediaDimXFeedDirDeclared   Integer32,
      prtInputMediaDimFeedDirChosen      Integer32,
      prtInputMediaDimXFeedDirChosen     Integer32,
      prtInputCapacityUnit               PrtCapacityUnitTC,
      prtInputMaxCapacity                Integer32,
      prtInputCurrentLevel               Integer32,
      prtInputStatus                     PrtSubUnitStatusTC,
      prtInputMediaName                  OCTET STRING,
      prtInputName                       OCTET STRING,
      prtInputVendorName                 OCTET STRING,
      prtInputModel                      OCTET STRING,
      prtInputVersion                    OCTET STRING,
      prtInputSerialNumber               OCTET STRING,
      prtInputDescription                OCTET STRING,
      prtInputSecurity                   PresentOnOff,
      prtInputMediaLoadTimeout           Integer32,
      prtInputNextIndex                  Integer32
```

Need to add:

```
      prtInputMediaIndex                Integer32,
```

4.6 Interpreter

```
  prtInterpreterIndex                  Integer32,
  prtInterpreterLangFamily          PrtInterpreterLangFamilyTC,
  prtInterpreterLangLevel              OCTET STRING,
  prtInterpreterLangVersion            OCTET STRING,
  prtInterpreterDescription            OCTET STRING,
  prtInterpreterVersion                OCTET STRING,
  prtInterpreterDefaultOrientation     PrtPrintOrientationTC,
  prtInterpreterFeedAddressability     Integer32,
  prtInterpreterXFeedAddressability    Integer32,
  prtInterpreterDefaultCharSetIn       CodedCharSet,
  prtInterpreterDefaultCharSetOut      CodedCharSet,
  prtInterpreterTwoWay                 PrtInterpreterTwoWayTC
```

4.7 Console Light

```
    prtConsoleLightIndex            Integer32,
    prtConsoleOnTime                Integer32,
    prtConsoleOffTime               Integer32,
    prtConsoleColor                 PrtConsoleColorTC,
    prtConsoleDescription           OCTET STRING
```

4.8 Marker

```
      prtMarkerIndex                  Integer32,
      prtMarkerMarkTech               PrtMarkerMarkTechTC,
      prtMarkerCounterUnit            PrtMarkerCounterUnitTC,
      prtMarkerLifeCount              Counter32,
      prtMarkerPowerOnCount           Counter32,
      prtMarkerProcessColorants       Integer32,
      prtMarkerSpotColorants          Integer32,
      prtMarkerAddressabilityUnit     INTEGER,
      prtMarkerAddressabilityFeedDir  Integer32,
      prtMarkerAddressabilityXFeedDir Integer32,
      prtMarkerNorthMargin            Integer32,
      prtMarkerSouthMargin            Integer32,
      prtMarkerWestMargin             Integer32,
      prtMarkerEastMargin             Integer32,
      prtMarkerStatus                 PrtSubUnitStatusTC
```

4.9 Marker Colorant

```
      prtMarkerColorantIndex          Integer32,
      prtMarkerColorantMarkerIndex    Integer32,
      prtMarkerColorantRole           PrtMarkerColorantRoleTC,
      prtMarkerColorantValue          OCTET STRING,
      prtMarkerColorantTonality       Integer32
```

4.10 Marker Supplies

```
   prtMarkerSuppliesIndex          Integer32,
   prtMarkerSuppliesMarkerIndex    Integer32,
   prtMarkerSuppliesColorantIndex  Integer32,
   prtMarkerSuppliesClass          PrtMarkerSuppliesClassTC,
   prtMarkerSuppliesType           PrtMarkerSuppliesTypeTC,
   prtMarkerSuppliesDescription    OCTET STRING,
   prtMarkerSuppliesSupplyUnit    PrtMarkerSuppliesSupplyUnitTC,
   prtMarkerSuppliesMaxCapacity    Integer32,
   prtMarkerSuppliesLevel          Integer32
```

4.11 Media Path

```
  prtMediaPathIndex                Integer32,
  prtMediaPathMaxSpeedPrintUnit  PrtMediaPathMaxSpeedPrintUnitTC,
  prtMediaPathMediaSizeUnit        PrtMediaUnitTC,
  prtMediaPathMaxSpeed             Integer32,
  prtMediaPathMaxMediaFeedDir      Integer32,
  prtMediaPathMaxMediaXFeedDir     Integer32,
  prtMediaPathMinMediaFeedDir      Integer32,
  prtMediaPathMinMediaXFeedDir     Integer32,
  prtMediaPathType                 PrtMediaPathTypeTC,
  prtMediaPathDescription          OCTET STRING,
  prtMediaPathStatus               PrtSubUnitStatusTC
```

4.12 Medium

From Mapping:

```
     prt-sub-unit-name
     prt-sub-unit-index
```

From DPA:

```
                 medium-associated-media
                 medium-assured-reproduction-area
                 medium-dimensions
                 medium-grain
                 medium-holes-axis-offset
                 medium-holes-count
                 medium-holes-diameter
                 medium-holes-locations
                 medium-holes-reference-edge
                 medium-realization
                 medium-sides
                 medium-size
                 medium-tooth
```

From InputEntry:

```
                 medium-form-parts
                 medium-color
                 medium-weight
                 medium-type
```

4.13 Output

```
    prtOutputIndex                    Integer32,
    prtOutputType                     PrtOutputTypeTC,
    prtOutputCapacityUnit             PrtCapacityUnitTC,
    prtOutputMaxCapacity              Integer32,
    prtOutputRemainingCapacity        Integer32,
    prtOutputStatus                   PrtSubUnitStatusTC,
    prtOutputName                     OCTET STRING,
    prtOutputVendorName               OCTET STRING,
    prtOutputModel                    OCTET STRING,
    prtOutputVersion                  OCTET STRING,
    prtOutputSerialNumber             OCTET STRING,
    prtOutputDescription              OCTET STRING,
    prtOutputSecurity                 PresentOnOff,
    prtOutputDimUnit                  PrtMediaUnitTC,
    prtOutputMaxDimFeedDir            Integer32,
    prtOutputMaxDimXFeedDir           Integer32,
    prtOutputMinDimFeedDir            Integer32,
    prtOutputMinDimXFeedDir           Integer32,
    prtOutputStackingOrder        PrtOutputStackingOrderTC,
    prtOutputPageDeliveryOrientation
        PrtOutputPageDeliveryOrientationTC,
    prtOutputBursting                 PresentOnOff,
    prtOutputDecollating              PresentOnOff,
    prtOutputPageCollated             PresentOnOff,
    prtOutputOffsetStacking           PresentOnOff
```

4.14 Alert

```
   prtAlertIndex                Integer32,
   prtAlertSeverityLevel        PrtAlertSeverityLevelTC,
   prtAlertTrainingLevel        PrtAlertTrainingLevelTC,
   prtAlertGroup                PrtAlertGroupTC,
   prtAlertGroupIndex           Integer32,
   prtAlertLocation             Integer32,
   prtAlertCode                 PrtAlertCodeTC,
   prtAlertDescription          OCTET STRING,
   prtAlertTime                 TimeTicks
```

5. New "Get-Sub-Units" IPP Printer Operation

The following new (post IPP/1.0) operation is introduced:

5.1.1 Get-MIB-Objects Operation

This OPTIONAL operation allows a client to retrieve the Sub-Unit objects (and their attributes) contained by the target Printer object. The client may also supply a list of contained object attribute names and/or attribute group names. A group of attributes will be returned for each MIB object that is returned.

This operation is similar to the Get-Jobs operation, except that this Get-Sub-Units operation returns attributes from possibly more than one object type.

5.1.1.1 Get-Sub-Units Request

The client submits the Get-Sub-Units request to a Printer object.

The following groups of attributes are part of the Get-Sub-Units Request:

Group 1: Operation Attributes

Target:
The "printer-uri" operation attribute which is the target for this operation as described in section 3.1.3.

Natural Language and Character Set:
The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.1.

Requesting User Name:
The "requesting-user-name" attribute SHOULD be supplied by the client as described in section 8.3.

"limit" (integer(1:MAX)):
The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It is an integer value that indicates a limit to the number of objects returned. The limit is a "stateless limit" in that if the value supplied by the client is 'N', then only the first 'N' MIB objects are returned in the Get-Sub-Units Response. There is no mechanism to allow for the next

'M' objects after the first 'N' objects. If the client does not supply this attribute, the Printer object responds with all applicable objects.

"requested-attributes" (1setOf keyword):
The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It is a set of object attribute names and/or attribute groups names in whose values the requester is interested. This set of attributes (where there is a match between what is requested and what attributes belong to which object types) is returned for each Sub-Unit object that is returned. The allowed attribute group names are: ‘all'. If the client does not supply this attribute, the Printer SHALL respond as if the client had supplied this attribute with two values: ‘prt-sub-unit-name' and ‘prt-sub-unit-index'

"which-sub-unit-types" (1setOf keyword):
The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It indicates which object types SHALL be returned by the Printer object. The values for this attribute are the name of the contained object types: ‘input', ‘output', ‘media-path', etc. If a client supplies some unsupported object type, the Printer object SHALL copy the attribute and the unsupported value(s) to the Unsupported Attributes response group, and return the ‘successful-ok-ignored-or-substituted-attributes' status code. If the client supplies either a "which-sub-units-by-name" or "which-sub-units-by-index" then the client MUST supply this "which-sub-unit-types" attribute. If the client supplies neither the "which-sub-unit-types" nor the "which-sub-units-by-name" nor the "which-sub-units-by-index" attributes, the Printer SHALL respond will all contained Sub-Unit objects.

"which-sub-units-by-name" (1setOf name):
The client OPTIONALLY supplies this attribute. The Printer object MUST support this attribute. It indicates (by name) which sub-unit objects SHALL be returned by the Printer object. If the client does not supply this attribute, the Printer object SHALL respond with all Sub Unit objects of the types specified in the client supplied "which-sub-unit-types" attribute. If the client supplies both the "which-sub-units-by-name" attribute and the "which-sub-unit-types" attributes there must be only a single value for "which-sub-unit-types". In other words, if both a type and one or more objects are specified, then they all must be of the same type (the value supplied in the "which-object- types" attribute).

"which-objects-by-index" (1setOf integer):
See text for "which-sub-units-by-name" above and the same applies to this attribute (except that the object specification is done by index not name).

5.1.1.2 Get-Sub-Units Response

The Printer object returns all of the Sub-Unit objects that match the criteria as defined by the attribute values supplied by the client in the request. It is possible that no Sub-Unit objects are returned since, for example, there may literally be no Output objects at the Printer and the client only asks for Output objects.

Group 1: Operation Attributes

Status Code and Message:
The response includes the MANDATORY status code and an OPTIONAL "status-message" (text) operation attribute as described in section 3.1.5.

Natural Language and Character Set:
The "attributes-charset" and "attributes-natural-language" attributes as described in section 3.1.4.2.

Group 2: Unsupported Attributes

This is a set of Operation attributes supplied by the client (in the request) that are not supported by the Printer object or that conflict with one another (see sections 3.2.1.2 and 15.3).

Groups 3 to N: Sub-Unit Object Attributes

The Printer object responds with one set of Sub-Unit Object Attributes for each returned MIB object. The Printer object ignores (does not respond with) any requested attribute or value which is not supported or which is restricted by the security policy in force. However, the Printer object SHALL respond with the 'unknown' value for any supported attribute (including all MANDATORY attributes) for which the Printer object does not know the value, unless it would violate the security policy. See the description of the "out-of-band" values in the beginning of Section 4.1. The Printer MUST always respond with the following attributes for each object in this order:

```
prt-sub-unit-type
prt-sub-unit-name
prt-sub-unit-index
```

For all returned Sub-Unit objects, any Sub-Unit object attributes which is of type ‘text' or ‘name' must be in the same natural language and charset as defined in the"attributes-natural-language" operation attribute unless each is overridden using the Natural Language Override as described in the sections 4.1.2 and 4.1.4.

Sub-Unit objects can be returned in any order, however if multiple object types are returned, then objects of similar type must be returned together. In other words, if a client finds in a response an Input object, and then an Output object, it MUST assume that it will find no more Input objects in the same response.

6. Conformance

TBD

6.1 Client Conformance Requirements

TBD

6.2 Server Conformance Requirements

TBD

6.2.1 Extensions

TBD

7. Security Considerations

TBD

8. References

[IPP-MOD]

[RFC1759]

9. Copyright Notice

None,

10. Author's Address

Scott A. Isaacson (Editor)
Novell, Inc.
122 E 1700 S
Provo, UT 84606

Phone: 801-861-7366
Fax: 801-861-2517
e-mail: sisaacson@novell.com

Tom Hastings
Xerox Corporation
701 S. Aviation Blvd.
El Segundo, CA 90245

Phone: 310-333-6413
Fax: 310-333-5514
e-mail: hastings@cp10.es.xerox.com