

Resource objects counter proposal

From Paul Moore and Tom Hastings

11/27/00

File: resource-objects-counter-proposal-001127-rev.doc

This document is a counter proposal to the one that Ira McDonald and Tom Hastings prepared for the September 2000 PWG meeting that defined a general Resource object with a "resource-type" attribute for sub-typing and operations to query, create, and delete Resource object instances. See ftp://ftp.pwg.org/pub/pwg/ipp/new_RES/draft-ietf-ipp-get-resource-01.pdf

Table of Contents

1	Introduction and Summary.....	1
2	Get-Printer-Collection-Rows operation.....	2
3	Operations to Add, Delete, or Modify Collection rows.....	3
3.1	Add-Printer-Collection-Row.....	3
3.2	Delete-Printer-Collection-Row.....	4
3.3	Modify-Printer-Collection-Row operation.....	4
4	Get-Printer-Collection-Row-Data, Set-Printer-Collection-Row-Data operations.....	5
5	Observations.....	5
6	Suggested member attributes.....	6
7	Possible '1setOf collection' Printer attributes.....	6
8	Examples.....	7
8.1	Input-trays.....	7
8.2	Media descriptions.....	7
8.3	Images.....	7

1 Introduction and Summary

Use '1setOf collection' attributes to describe fonts, media, paper trays, downloaded JPEGs, ICC Color Profiles, macros, ... Some of these resources can be downloaded into the Printer, some can be installed by means outside the IPP protocol, and some can be properties or characteristics of the Printer as it comes from the vendor or is configured by the administrator when the Printer is installed. Some of these resources can have associated opaque binary data, such as font data, while others consist solely of attributes.

These collection attributes are retrieved using the regular GetPrinterAttributes operation, since they are ordinary collection attribute with an attribute syntax of '1setOf collection'. More than one can be asked for in a request. All of the members of all of the rows are

36 returned (as per the current collections spec). Note: according to the collection spec the
37 "xxx-supported" Job Template attributes usually have the attribute syntax: '1setOf type2
38 keyword', rather than '1setOf collection'. The keywords indicate which member
39 attributes are supported for the collection and the corresponding "xxx-supported" indicate
40 the values supported for each "xxx" member attribute. For such Job Template attributes,
41 a new naming convention is introduced: "xxx-rows" for the Printer attributes with the
42 attribute syntax of '1setOf collection'.

43 The "xxx-rows" (1setOf collection) attributes are never returned by Get-Printer-Attributes
44 unless they are explicitly asked for (i.e., they are never included in groups or 'all', since
45 there would be too much data in the response). A single row can be queried using a new
46 Get-Printer-Collection operation described in the next section.

47 The following new operations are defined for use with '1setOf collection' attributes:

- 48 • Get-Printer-Collection-Rows - return rows of a '1setOf collection' Printer attribute
49 based on a simple filter supplied by the client
- 50 • Add-Printer-Collection-Row - add a row to a '1setOf collection' Printer attribute
- 51 • Delete-Printer-Collection-Row - delete a row of a '1setOf collection' Printer
52 attribute
- 53 • Modify-Printer-Collection-Row - modify a row of a '1setOf collection' Printer
54 attribute
- 55 • Get-Printer-Collection-Row-Data - same as Get-Printer-Collection-Row, and in
56 addition get the row's associated opaque data.
- 57 • Set-Printer-Collection-Row-Data - same as Set-Printer-Collection-Row, and in
58 addition set the row's associated opaque data.

59 For consistency all six operations have an Operation Attributes Group and a Printer
60 Attributes Group in each request and response. The response always includes *all* of the
61 member attributes of each row returned. In addition to the usual request operation
62 attributes for a Printer operation, all six operations MUST include:

63 "collection-attribute" (type2 keyword) - which identifies the collection attribute to
64 be affected. For example: "collection-attribute" = 'font-rows-supported'
65 or "collection-attribute" = 'tray-rows-supported'

66 **2 Get-Printer-Collection-Rows operation**

67 The Get-Printer-Collection-Rows operation retrieves one or more values of a 1setOf
68 collection attribute. Each collection value is called a "row". The rows are selected on the
69 basis of a filter specified in the operation. Only one Filter Attribute is permitted and it is
70 expressed as the only attribute in the Printer Attributes group. The Printer matches the
71 Filter Attribute against all the member attribute of all of the rows in the collection value.
72 The attribute name, syntax, and value of the Filter Attribute MUST be the same as the
73 member attribute in the 1setOf collection, in order to match. A value match occurs if all
74 of the values of the Filter Attribute are a subset of the member attribute of a row..

75 The client MUST also supply the Filter Attribute as the only attribute in a separate Printer
76 Attributes group. For example:

77 "font-point-size" = '12'

78 The Printer returns all member attributes of all rows of the collection that match the Filter
79 Attribute. Each row is returned in a separate Printer Attributes group in the response
80 (like Get-Jobs response). If no rows match then the status code 'client-error-not-found'
81 error is returned. **ISSUE: Or should the status code be 'successful-ok' (0), with an empty
82 Printer Attributes group returned to be more like Get-Printer-Attributes?**

83 When a collection attribute with a 1setOf collection attribute syntax is defined, the
84 definition SHOULD specify an 'identifying member attribute', called the Key Attribute
85 that uniquely identifies a row. No two rows can have the same Key Attribute value. The
86 role of the Key Attribute is the same as a primary key in a data base. The Key Attribute
87 facilitates direct indexing into 1setOf collection attributes. Possible examples could be
88 tray name, media name, font name, etc. In some cases the identifying member attribute
89 could be a printer generated unique ID.

90 If a collection attribute has a '1setOf collection' attribute syntax, but the definition of that
91 attribute does not indicate which member attribute is the Key Attribute, that collection
92 attribute MAY still be used in the Get-Printer-Collection-Rows operation, but there is no
93 way for the client to unambiguously request a single row.

94 **3 Operations to Add, Delete, or Modify Collection rows**

95 The operations defined in this section add, delete, or modify a row in a '1setOf collection'
96 Printer attribute that is defined to have a Key Attribute.

97 These operation do not work on all collections – there are some collections that represent
98 state or non-logical capabilities of the device (paper loaded, input trays, etc.). In this case
99 the collections are read-only (either by definition or in a particular implementation).

100 There also can exist Printer collection attributes that represent collections that are
101 software modifiable entities but that are still not updated via these operations in an
102 implementation. For example fonts could be loaded by a specific set of font management
103 operations, rather than these operations.

104 What this means is that the collection querying can be used on all entities that are
105 represented as '1setOf collection' but there can be many mechanisms that create those
106 collections. The definition of the collection MUST indicate how the rows are created,
107 modified, and removed.

108 **3.1 Add-Printer-Collection-Row**

109 This operation adds a collection row to an existing '1setOf collection' Printer attribute
110 provided that the collection definition defined a Key Attribute.

111 In addition to the "collection-attribute" operation attribute, the client MUST supply the
112 Key Attribute as the first attribute in the Printer Attributes group in the request. For
113 example, "font-name" = 'TimesRomanItalic'. The client supplies the remaining attributes

114 for the row as the remaining attributes in the Printer Attributes group. For example,
115 "font-size" = '12', "font-style" = 'italic', etc.

116 If the row already exists, the Printer MUST reject the request and return the (new) 'client-
117 error-row-already-exists'.

118 If the "collection-attribute" does not specify a '1setOf collection' attribute whose
119 definition includes a Key Attribute or the first attribute in the Printer Attributes group is
120 not the Key Attribute defined for the collection, the Printer MUST reject the request with
121 the 'client-error-bad-request'.

122 **3.2 Delete-Printer-Collection-Row**

123 This operation deletes a collection row from an existing '1setOf collection' Printer
124 attribute provided that the collection definition defined a Key Attribute.

125 In addition to the "collection-attribute" operation attribute, the client MUST supply the
126 Key Attribute as the only attribute in the Printer Attributes group in the request. For
127 example, "font-name" = 'TimesRoman'.

128 If the row does not exist, the Printer MUST reject the request and return the 'client-error-
129 not-found' error status code.

130 If the "collection-attribute" does not specify a '1setOf collection' attribute whose
131 definition includes a Key Attribute or the only attribute in the Printer Attributes groups is
132 not the Key Attribute defined for the collection, the Printer MUST reject the request with
133 the 'client-error-bad-request'.

134 **3.3 Modify-Printer-Collection-Row operation**

135 This attribute modifies an existing collection row of an '1setOf collection' Printer
136 attribute provided that the collection definition defined a Key Attribute.

137 In addition to the "collection-attribute" operation attribute, the client MUST supply the
138 Key Attribute as the first attribute in the Printer Attributes group in the request. For
139 example, "font-name" = 'TimesRomanItalic'. The client supplies the remaining attributes
140 to be modified for the row as the remaining attributes in the Printer Attributes group. For
141 example, "font-size" = '12', "font-style" = 'italic', etc. Any member attributes of the row
142 that the client omits are unchanged.

143 If the row does not exist, the Printer MUST reject the request and return the 'client-error-
144 not-found' error status code.

145 If the "collection-attribute" does not specify a '1setOf collection' attribute whose
146 definition includes a Key Attribute or the first attribute in the Printer Attributes groups is
147 not the Key Attribute defined for the collection, the Printer MUST reject the request with
148 the 'client-error-bad-request'.

149 **4 Get-Printer-Collection-Row-Data, Set-Printer-** 150 **Collection-Row-Data operations**

151 Some 'IsetOf collection' attributes may have data associated with their rows. In this case
152 then one choice available to the designer of the collection is to use the Get-Printer-
153 Collection-Row-Data and Set-Printer-Collection-Row-Data operations to read and write
154 opaque blobs.

155 For the Get-Printer--Collection-Row-Data, the client supplies the "collection-attribute"
156 name and the Key Attribute as in the other Collection Row operations. The row's
157 attributes are returned in the Printer Attributes Group, followed by the data as a data
158 stream in the response (packaged the same way the print-job's data is following the 'end-
159 of-attributes-tag').

160 The data is sent in the same way using the Set-Printer-Collection-Row-Data operation.

161 Note that for some collection it might be possible to read the data but not write it
162 (uploading font metrics from ROM for example). Also it might be possible to write it but
163 not read it (macros are not intended to be used outside the printer so there is not point in
164 providing read capabilities).

165 Alternative design #1. There is a member attribute for each row that specifies a URI for
166 the data. The Add-Printer-Collection-Row operation returns the URI generated by the
167 Printer when the row is created. The data is got and set by HTTP GET and POST to that
168 URI. One problem with using HTTP GET and PUT to get or set data is the security
169 checking. For example, ordinary users may not be able to add some resources to the
170 Printer, such as media, but may be able to add other resources, such as images, to the
171 Printer (for a leased amount of time).

172 Alternative design #2. Collections that have associated data have explicit row creation
173 and operations (Load-Font operation for example) but the data is read by HTTP get or an
174 IPP Get-Printer-Collection-Row-Data. This alternative overcomes the non-atomic nature
175 of adding a row then uploading the data.

176 **5 Observations**

177 Some collections may have read-only rows and read-write rows (fonts supported may
178 include ROM fonts and soft fonts).

179 Jobs and Subscriptions could have been done using 'IsetOf collection' Printer attributes,
180 but since we already have operations defined for Jobs and Subscription objects.

181 Driver down loading could have been done using 'IsetOf collection' Printer attributes, but
182 we have a specification that uses Get-Printer-Attributes and a new Get-Client-Print-
183 Support-Files operation.

184 Expiration times for collection rows can be specified in the collection definition if that is
185 what the collection needs. For example, if users are allowed to down load images into
186 the Printer for a period of time.

187 This mechanism is only defining a standardized ways of viewing structured data – it does
188 not imply that common mechanisms must be used by implementations.

189 **6 Suggested member attributes**

190 In order to get some consistency in definition of '1setOf collection' Printer attributes, the
191 following member attribute names and attribute syntaxes are suggested if the member
192 attribute is appropriate for the resource. However, none of these attributes are
193 REQUIRED for a definition.

194 For key attributes that a client can supply (but cannot modify):

195 xxx-name (name(127)) or xxx-key (name(127) | type3 keyword) - Key Attribute
196

197 For attributes that a client can supply (or modify):

198 xxx-info (text(127)) - general information

199 xxx-create-date-time (dateTime) - the date and time that the resource was
200 originally created, not added to the Printer.

201 xxx-lease-duration (integer(0:MAX)) - lease duration in seconds, 0 is infinite

202 xxx-data-uri (1setOf uri) - uri of the data when supplied by the client

203 xxx-data-k-octets (integer(0:MAX)) - size of the data

204 xxx-data-compression (type3 keyword) - data compression
205

206 For READ-ONLY attributes populated by the Printer:

207 xxx-id (integer(1:MAX)) - integer id for those resources that do not have a natural
208 name supplied by the client.

209 xxx-create-user-name (name(MAX)) - user name who added the resource to the
210 Printer

211 xxx-create-time (integer(MIN:MAX)) - the "printer-up-time" when the resource
212 was added to the Printer. A 0 or negative value means before this Printer
213 power-up (see RFC 2911 section 4.3.14).

214 xxx-expiration-time (integer(0:MAX)) - the "printer-up-time" when the lease
215 expires

216 xxx-data-uri (1setOf uri) - uri of the data when supplied by the Printer
217

218 **7 Possible '1setOf collection' Printer attributes**

Collection	Identifying member	Members	Data
Input trays	name	Loaded media, state, capacity, level	none
Output bins	name	State, capacity, level	none
Fonts supported	Name (face-size-style)	Size, style, format	Font metrics
Media-descriptions	name	Size, weight,	none
Macros	name	Date, format	Macro data
Images	name	Date, format,	Image data

		description	
--	--	-------------	--

219

220 **8 Examples**

221 These are examples of how this proposal could be used to represent various items. The
 222 full variety of choices is used. *These are not intended as actual proposals for their*
 223 *respective collections, but rather just indicate how the mechanism proposed in this paper*
 224 *would work.*

225 **8.1 Input-trays**

226 The "input-tray-rows-supported" (1setOf collection) Printer attribute contains one row for
 227 each input tray supported by the printer.

228 The rows are identified by an "input-tray-name" (type3 keyword | name(MAX)) Key
 229 Attribute whose value is either defined by the PWG or is defined by the implementation.
 230 The values of the "input-tray-name" member attribute may be submitted in a Job Creation
 231 operation as the value of a (new) "input-tray" Job Template attribute.

232 The member attributes include "input-tray-max-capacity", "input-tray-current-level",
 233 "input-tray-status", and "input-tray-media-name" with semantics taken from the Printer
 234 MIB.

235 The rows of this collection are read using the Get-Printer-Collection-Rows operation.
 236 Rows are not created or deleted, though in some implementations, certain member
 237 attributes, such as "input-tray-media-name" can be set.

238 **8.2 Media descriptions**

239 The "media-rows-supported" (1setOf collection) Printer attribute contains one row for
 240 each supported / known media. See the PWG Production Printing Extension spec for the
 241 definition of the member attributes. The "media-col" (collection) Job Template attribute
 242 can be supplied by the client in Job Creation operations.

243 The rows are identified by a "media-key" (type3 keyword | name(MAX)) Key Attribute
 244 whose value is either defined by the PWG or is defined by the administrator. The values
 245 of the "media-key" member attribute may be submitted in a Job Creation operation as the
 246 value of the IPP/1.1 "media" Job Template attribute.

247 The member attributes include "media-size" (1setOf collection) {x-dimension, y-
 248 dimension", "media-weight", "media-color", etc.

249 The rows of this collection are read using Get-Printer-Collection-Rows. Rows are created
 250 using the Add-Printer-Collection-Row operation and deleted using the Delete-Printer-
 251 Collection-Row operation.

252 **8.3 Images**

253 The "image-rows-supported" (1setOf collection) Printer attribute contains one row for
 254 each supported / known.

255 The rows are identified by an "image-name" (name(MAX)) Key Attribute whose value is
256 either defined by its creator. The values of the "image-name" member attribute may be
257 submitted in a Job Creation operation as the value of the (new) "image" Job Template
258 attribute.

259 Member attributes include "image-size", "image-format", "image-version", etc. The
260 images are created by a (new) image-specific operation: Load-Image. This operation
261 includes all the member attributes that describe the image plus the image data as an
262 attached 'print-job' data stream. Some member attributes are derived from the image
263 (size for example).

264 The Set-Printer-Collection-Row-Data operation is not used. Instead, a (new) Load-
265 Image operation is defined and the client may specify an expiration time for the image.

266 Images may be explicitly deleted using the Delete-Printer-Collection-Row operation. The
267 Add-Printer-Collection-Row and Modify-Printer-Collection-Row operations are not
268 defined for use with images.

269 The image data is not readable externally.

270