

# Resource objects summary proposal

From Ira McDonald and Tom Hastings

11/27/00

**File: resource-objects-summary-proposal-001127-rev.doc**

This document is a summary proposal of 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](ftp://ftp.pwg.org/pub/pwg/ipp/new_RES/draft-ietf-ipp-get-resource-01.pdf). In order to compare the resource object approach more directly with Paul Moore's counter proposal, this document is shows revisions to Paul's proposal to make the Resource object proposal. Also non-essential differences, such as filtering have been made the same, so that only real differences remain.

## Table of Contents

1	Introduction and Summary.....	1
2	Get-Resource-Attributes and Get-Resources operations .....	2
3	Operations to Add, Delete, or Modify Resource object instances .....	3
3.1	Create-Resource .....	4
3.2	Delete-Resource .....	4
3.3	Set-Resource-Attributes.....	4
3.4	Renew-Resource .....	5
4	Get-Resource-Data, Create-Resource operations .....	5
5	Observations .....	6
6	Suggested Resource attributes .....	6
7	Possible Resource types.....	7
8	Examples .....	7
8.1	Input-trays .....	7
8.2	Media descriptions .....	7
8.3	Images .....	8

## 1 Introduction and Summary

Use a polymorphic generic Resource object type with sub-typing to describe fonts, media, paper trays, downloaded JPEGs, ICC Color Profiles, macros, ... Some of these resources can be down-loaded 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

36 of these resources can have associated opaque binary data, such as font data, while others  
37 consist solely of attributes.

38 These Resource object attributes are retrieved using the (new) Get-Resource-Attributes  
39 and Get-Resources operations which are modeled on the IPP/1.1 Get-Job-Attributes and  
40 Get-Jobs operations and the Get-Subscription-Attributes and Get-Subscriptions  
41 operations. Resource objects that can be loaded are defined to have Resource Template  
42 attributes (just like Job and Subscription objects), so that there are "xxx" Resource  
43 attributes and "xxx-supported" Printer attributes.

44 The following new operations are defined for use with Resource objects:

- 45 • Get-Resource-Attributes - returns the requested attributes of the identified  
46 Resource object instance.
- 47 • Get-Resources - return the requested attributes of the Resource object instances  
48 based on a simple filter supplied by the client
- 49 • Create-Resource - add a Resource object instance to a Printer
- 50 • Delete-Resource - delete a Resource object instance from the Printer
- 51 • (new) Set-Resource-Attributes - modify a Resource object instance of a Printer
- 52 • Get-Resource-Data - same as Get-Resource-Attributes, and in addition get the  
53 object instance's associated opaque data.
- 54 • Create-Resource - same operation sets the object instance's associated opaque  
55 data.
- 56 • Renew-Resource - update the lease time for the Resource object instance for those  
57 Resource types that have leases.

58 For consistency all seven operations have an Operation Attributes Group and a Resource  
59 Attributes Group in each request and response. The response always includes the  
60 requested Resource object attributes. In addition to the usual request operation attributes  
61 for a Printer operation, all six operations MUST include:

62 "resource-type" (type2 keyword) - which indicates the type of Resource, e.g.,  
63 'media', 'font', 'image', 'input-tray', 'output-bin', etc.

64 Either "resource-name" (name(127)) or "resource-id" (integer(1:MAX)) -  
65 identifies the resource object instance. The Printer MUST support both.

## 66 **2 Get-Resource-Attributes and Get-Resources** 67 **operations**

68 The Get-Resource-Attributes and Get-Resources operations for Resource objects follow  
69 the pattern established by the IPP/1.1 Get-Job-Attributes and Get-Jobs operations for Job  
70 objects and the Get-Subscription-Attributes and Get-Subscriptions operations for  
71 Subscription objects. The Get-Resource-Attributes operation retrieves requested  
72 attributes from one Resource object instance specified by the Key Attribute supplied by  
73 the client.

74 The Get-Resources operation retrieves requested attributes from one or more Resource  
75 object instances. The Resource objects instances are selected on the basis of a filter  
76 specified in the operation. Only one Filter Attribute is permitted and it is expressed as the  
77 only attribute in the Resource Attributes group. The Printer matches the Filter Attribute  
78 against all the attributes of all of the Resource object instances. The attribute name,  
79 syntax, and value of the Filter Attribute MUST be the same as one of the attributes in the  
80 Resource object instance, in order to match. A value match occurs if all of the values of  
81 the Filter Attribute are a subset of the Resource object attribute.

82 The client MUST also supply the Filter Attribute as the only attribute in a separate  
83 Resource Attributes group. For example:

84           "font-point-size" = '12'

85 The Printer returns the requested attributes of all Resource object instances that match the  
86 Filter Attribute. Each row is returned in a separate Resource Attributes group in the  
87 response (like Get-Jobs response). If no rows match then the status code 'client-error-not-  
88 found' error is returned. **ISSUE: Or should the status code be 'successful-ok' (0), with an  
89 empty Resource Attributes group returned to be more like Get-Printer-Attributes?**

90 When a Resource object type is defined, the definition SHOULD specify an 'identifying  
91 attribute', called the Key Attribute that uniquely identifies an object instance. No two  
92 Resource object instances of the same type can have the same Key Attribute value. The  
93 role of the Key Attribute is the same as a primary key in a data base. The Key Attribute  
94 facilitates direct indexing into Resource object instances. Possible examples could be tray  
95 name, media name, font name, etc. In some cases the identifying member attribute could  
96 be a printer generated unique ID.

97 If a Resource object type definition does not indicate which member attribute is the Key  
98 Attribute, that Resource object type MAY still be used in the Get-Resources operation,  
99 but there is no way for the client to unambiguously request a single object instance.

### 100 **3 Operations to Add, Delete, or Modify Resource object** 101 **instances**

102 The operations defined in this section add, delete, or modify a Resource object instance  
103 for object types that is defined to have a Key Attribute.

104 These operations do not work on all Resource types– there are some Resource types that  
105 represent state or non-logical capabilities of the device (paper loaded, input trays, etc.). In  
106 this case these Resource instances are read-only (either by definition or in a particular  
107 implementation).

108 There also can exist Resource object attributes that represent objects that are software  
109 modifiable entities but that are still not updated via these operations in an  
110 implementation. For example fonts could be loaded by a specific set of font management  
111 operations, rather than these operations.

112 What this means is that the Resource object querying can be used on all entities that are  
113 represented as Resource objects but there can be many mechanisms that create those

114 instances. The definition of the Resource object type MUST indicate how the rows are  
115 created, modified, and removed.

### 116 **3.1 Create-Resource**

117 This operation adds a Resource object instance provided that the Resource object type  
118 definition defined a Key Attribute.

119 In addition to the "resource-type" and either "resource-name" or "resource-id" operation  
120 attributes, the client MUST supply the Key Attribute as the first attribute in the Resource  
121 Attributes group in the request. For example, "font-name" = 'TimesRomanItalic'. The  
122 client supplies the remaining attributes for the object instance as the remaining attributes  
123 in the Resource Attributes group. For example, "font-size" = '12', "font-style" = 'italic',  
124 etc.

125 If the object instance already exists, the Printer MUST reject the request and return the  
126 (new) 'client-error-row-already-exists'.

127 If the "resource-type" does not specify a Resource type whose definition includes a Key  
128 Attribute or the first attribute in the Resource Attributes group is not the Key Attribute  
129 defined for the Resource type, the Printer MUST reject the request with the 'client-error-  
130 bad-request'.

### 131 **3.2 Delete-Resource**

132 This operation deletes a Resource object instance provided that the Resource object type  
133 definition defined a Key Attribute.

134 In addition to the "resource-type" and either "resource-name" or "resource-id" operation  
135 attributes, the client MUST supply the Key Attribute as the only attribute in the Resource  
136 Attributes group in the request. For example, "font-name" = 'TimesRoman'.

137 If the object instance does not exist, the Printer MUST reject the request and return the  
138 'client-error-not-found' error status code.

139 If the "resource-type" does not specify a Resource type whose definition includes a Key  
140 Attribute or the only attribute in the Resource Attributes group is not the Key Attribute  
141 defined for the Resource type, the Printer MUST reject the request with the 'client-error-  
142 bad-request'.

### 143 **3.3 Set-Resource-Attributes**

144 This operation modifies an existing Resource object instance provided that the Resource  
145 object type definition defined a Key Attribute.

146 In addition to the "resource-type" and either "resource-name" or "resource-id" operation  
147 attributes, the client MUST supply the Key Attribute as the first attribute in the Resource  
148 Attributes group in the request. For example, "font-name" = 'TimesRomanItalic'. The  
149 client supplies the remaining attributes to be modified for the row as the remaining  
150 attributes in the Resource Attributes group. For example, "font-size" = '12', "font-style"  
151 = 'italic', etc. Any Resource attributes of the object instance that the client omits are  
152 unchanged.

153 If the object instance does not exist, the Printer MUST reject the request and return the  
154 'client-error-not-found' error status code.

155 If the "resource-type" does not specify a Resource type whose definition includes a Key  
156 Attribute or the first attribute in the Resource Attributes groups is not the Key Attribute  
157 defined for the Resource type, the Printer MUST reject the request with the 'client-error-  
158 bad-request'.

### 159 **3.4 Renew-Resource**

160 This operation renews the lease for the specified Resource object instance provided that  
161 the Resource object type definition defined a Key Attribute.

162 In addition to the "resource-type" and either "resource-name" or "resource-id" operation  
163 attributes, the client MUST supply the Key Attribute as the first attribute in the Resource  
164 Attributes group in the request. For example, "font-name" = "TimesRomanItalic". The  
165 client supplies the "xxx-lease-duration" attributes as the remaining attribute in the  
166 Resource Attributes group. For example, "font-lease-duration" = nnn.

167 If the object instance does not exist, the Printer MUST reject the request and return the  
168 'client-error-not-found' error status code.

169 If the "resource-type" does not specify a Resource type whose definition includes a Key  
170 Attribute or the first attribute in the Resource Attributes groups is not the Key Attribute  
171 defined for the Resource type or the Resource type definition does not include an "xxx-  
172 lease-duration" attribute, the Printer MUST reject the request with the 'client-error-bad-  
173 request'.

## 174 **4 Get-Resource-Data, Create-Resource operations**

175 Some Resource types may have data associated with each instance. In this case then one  
176 choice available to the designer of the Resource type is to use the Get-Resource-Data and  
177 Create-Resource operations to read and write opaque blobs (as well as the Resource  
178 object attributes).

179 For the Get-Resource-Data, the client supplies the "resource-type" and the Key Attribute  
180 as in the other Resource operations. The requested Resource object attributes are returned  
181 in the Resource Attributes Group, followed by the data as a data stream in the response  
182 (packaged the same way the print-job's data is following the 'end-of-attributes-tag').

183 The data is sent in the same way using the Create-Resource operation.

184 Note that for some Resource types it might be possible to read the data but not write it  
185 (uploading font metrics from ROM for example). Also it might be possible to write it but  
186 not read it (macros are not intended to be used outside the printer so there is not point in  
187 providing read capabilities).

188 Alternative design #1. Resource types that have associated data have explicit object  
189 creation operations (Load-Font operation for example) but the data is read by HTTP get  
190 or IPP Get-Resources and Get-Resource-Attributes operations. This alternative  
191 overcomes the non-atomic nature of adding a row then uploading the data.

## 192 **5 Observations**

193 Some Resource types may have read-only object instances and read-write object  
194 instances (fonts supported may include ROM fonts and soft fonts).

195 Jobs and Subscriptions could have been done using Resource objects of type 'job' and  
196 'subscription', but we already have operations defined for Jobs and Subscription objects.

197 Driver down loading could have been done usingResource objects of type 'driver', but we  
198 have a specification that uses Get-Printer-Attributes and a new Get-Client-Print-Support-  
199 Files operation.

200 Expiration times for Resource types can be specified in the Resource type definition if  
201 that is what the Resource type needs. For example, if users are allowed to down load  
202 images into the Printer for a period of time.

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

## 205 **6 Suggested Resource attributes**

206 In order to get some consistency in definition of Resource types, the following attribute  
207 names and attribute syntaxes are suggested if the attribute is appropriate for the resource  
208 type. However, none of these attributes are REQUIRED for a definition.

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

210 xxx-name (name(127)) or xxx-key (name(127) | type3 keyword) - Key Attribute  
211

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

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

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

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

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

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

219 xxx-data-compression (type3 keyword) - data compression  
220

221 For READ-ONLY attributes populated by the Printer:

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

224 xxx-create-user-name (name(MAX)) - user name who added the resource to the  
225 Printer

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

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

231 xxx-data-uri (1setOf uri) - uri of the data when supplied by the Printer  
232

233 **7 Possible Resource types**

Collection	Key Attribute	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, description	Image data

234

235 **8 Examples**

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

240 **8.1 Input-trays**

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

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

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

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

253 **8.2 Media descriptions**

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

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

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

264 The rows of this collection are read using Get-Printer-Collection-Rows. Rows are created  
265 using the Add-Printer-Collection-Row operation and deleted using the Delete-Printer-  
266 Collection-Row operation.

### 267 **8.3 Images**

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

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

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

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

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

284 The image data is not readable externally.

285