

# Resolution of Notification Issues

September 23, 1999

This document contains the resolution of each issue 1-20 as decided at the IPP meeting in Denver, plus issues 21-22. This leaves only issues 1, and 21-22 left to resolve. Please check issue 20, since it was done quickly at the end of the meeting.

Both Notification documents (with dates of September 9 and 10) that were posted on 9/13/99 have the same set of issues (issue 1-8). The Alternative spec that aligns Per-Job subscriptions with Per-Printer subscription (Sept 10 spec) has an additional issue (see issue 5a below).

Carl-Uno will bring copies of this issues list to the meeting.

Issues 9 and 10 added during the 9/15/99 IPP telecon and issues 11-14 added as a result of producing the MIB for IPP SNMP notification. The page numbers and section numbers refer to the Notification Alternative spec, dated September 10, 1999, since that alternative had considerable support during the telecon. So agreeing to go with the alternative that aligns Per-Job with Per-Printer subscriptions should be the first issue to be considered. Therefore, I've numbered it ISSUE 0:

Pages 1-47:

**ISSUE 0** - Should we adopt the alternative Notification which aligns the Per-Job subscription mechanism with the Per-Printer subscription mechanism and uses Subscription objects for both? Both of the Notification specs posted on 9/15 (with 9/9/99 and 9/10/99 dates in the document) are at the same level of development, so switching to the alternative does not require any additional specification writing.

**Resolution:** We adopted the alternative notification mechanism.

Page 19, section 5.1, lines 551-553:

**ISSUE 1:** Once a number of delivery solutions have been developed and evaluated, we may want to make one or several of them REQUIRED for implementation to ensure a minimum set of interoperability. Which one or ones should be REQUIRED?

**Resolution:** Still an issue

Page 22, section 5.3, lines 677-683:

**ISSUE 2** - Should "notify-content-type (mimeType)" be changed to notify-content-type (type2 keyword), with values: 'human-consumable' and 'machine-readable'? Then content types that are neither 'application/ipp' or 'text/plain; charset=utf-8' could be handled without having to be registered. For example, the 'snmp-ipp' is a trap MIB Machine Consumable format wouldn't need to be registered (or have some special rule since the 'snmp-ipp' value doesn't have a registered MIME type and doesn't fit either 'application/ipp' or 'text/plain'. Another advantage is that the charset would always be indicated in the "notify-attributes-charset" attribute, even for plain text.

**Resolution:** We decided on a new solution. The notification specification will specify the single machine-readable format for each defined notify-scheme (i.e. delivery method). It is possible for a notify-scheme to have no machine-readable format. The notification specification will also specify whether there can be a human-readable format for each defined notify-scheme. But it will not define them. Rename "notify-content-type" to "notify-text-format" attribute. If the client doesn't supply this attribute, the machine-readable format is used that is defined for each notify-scheme, if defined for that notify-scheme. If a machine-readable format is not defined for the requested notify-scheme and the client fails to supply the "notify-text-format" attribute, the Printer selects one of the value of its "notify-text-format-supported" attribute in an implementation-defined manner for use in the notification.

46 An implementation MUST support the attribute "notify-text-formats-supported" (perhaps find a better  
47 name) if it supports any human-readable format, e.g. "text/plain", or "text/html", or 'none'. The value of this  
48 attribute is a 1setOf mimeType. If an implementation supports a particular human-readable format, it  
49 must support this format for all notify-schemes that have a human-readable format.

50 A client can determine what formats are supported for a notify-scheme as follows. The implementation  
51 contains a hard coding of the machine-readable format (obtained from the specification). If the hard coding  
52 (obtained from the specification) prohibits a human-readable format for the notify-scheme, or the printer  
53 doesn't support "notify-text-format-supported" attribute, there are no human-readable formats for any  
54 notify-scheme. Otherwise, the attribute "notify-text-formats-supported" specifies the supported mimeType  
55 for those delivery schemes that are supported and permit a human-readable format.

56 The rationale behind this decision is that the machine-readable format seems easy to pick for each notify-  
57 scheme and thus easy to document. It is easy for a printer to support most machine-readable formats. It is  
58 much harder to support a human-readable format because of localization issues. Once the code is written to  
59 support a particular human-readable format, it is easy to transmit it on any of the supported notify-schemes.  
60 Thus, if a vendor decides to support a notify-scheme, it has already committed to implement the machine  
61 readable-format. This may be simple if existing code can be reused, e.g. application/ipp and or more  
62 difficult if new code must be written, e.g. it is SNMP.

63

64 Page 22, section 5.3, lines 684-689:

65 **ISSUE 3** - There is no good way for a client to discover which "notify-content-type" values are supported  
66 for each delivery method "notify-schemes-supported". Adding "notify-content-type-supported" doesn't  
67 work, since the delivery methods have to be paired with the content types. Instead, how about only having  
68 each scheme support one or the other content type? Then we don't need "notify-content-type" as all. For  
69 the 'mailto' scheme, we specify it is human-consumable, i.e., text, and then register a 'mailto-ipp' scheme  
70 which is the 'application/ipp' scheme.

71 **Resolution:** See Issue 2 for the resolution of this.

72

73 Page 23, section 5.5, lines 703-708:

74 **ISSUE 4** - Should the Subscription object attributes "notify-attributes-charset (charset)" and "notify-  
75 attributes-natural-language (naturalLanguage)" be renamed to simply "attributes-charset (charset)" and  
76 "attributes-natural-language (naturalLanguage)" so that all IPP requests and objects have the same names?  
77 Then the "notify-attributes-charset (charset)" and "notify-attributes-natural-language (naturalLanguage)"  
78 would only be operation attributes that set or reflect the "attributes-charset (charset)" and "attributes-  
79 natural-language (naturalLanguage)" Subscription object attributes.

80 **Resolution:** We changed the name for the Subscription object attribute. Because these attributes specify the  
81 charset and language for whatever goes in the event notification, and the event notification contents may be  
82 attributes or may be text, we decided to name the attributes in the subscription object and operation  
83 attributes to be "notify-charset" and "notify-natural-language". The "attributes-charset" and "notify-  
84 attributes-natural-language" remain as REQUIRED operation attributes for all operation requests and  
85 responses.

86

87 Page 25, section 6, lines 780-782:

88 **ISSUE 5** - Shouldn't we add a "number-of-printer-subscriptions" Printer Description attribute, since the  
89 client is NOT assured that it can determine the number of outstanding Per-Printer Subscriptions by doing  
90 Get-Printer-Subscriptions because of access control?

91 **Resolution:** No. We couldn't find a real use-case for this attribute.

92

93 Page 25, section 6, lines 783-786:

94 **ISSUE 5a** - Shouldn't we add a "number-of-job-subscriptions" Job Description attribute, since the client is  
95 NOT assured that it can determine the number of outstanding Per-Job Subscriptions by doing Get-  
96 Subscriptions because of access control and since there are no Job Descriptions attributes associated with  
97 notification.

98 During the 9/15 telecon, there were four alternatives for Job Description attributes for notification

- 99 a. "number-of-job-subscriptions (integer(0:MAX))" - the number of Per-Job subscriptions for this job  
100 b. "job-subscriptions (boolean)" - whether or not this job has any Per-Job subscriptions  
101 c. "job-subscription-ids (1setOf (integer(1:MAX)))" - the set of Per-Job subscription ids for the  
102 Subscription objects associated with this job. In order to indicate that there are no Per-Job  
103 subscriptions, we need to decide on several alternatives: (1) don't return this attribute, (2) return  
104 this attribute with the 'no-value' out-of-band attribute value (see [ipp-mod] section 4.1), (3) return  
105 this attribute with a single integer 0 value (since 0 isn't a legal Subscription ID value), (4) return  
106 new '0setOf' attribute syntax in which 0 or more values can be represented.  
107 d. Don't have any Job Description attributes for notification (as in the current alternative spec).

108

109 **Resolution:** No. We couldn't see any reasonable use cases, so "d" is the choice.

110

111 Page 34, section 8.2.1, lines 1027-1032:

112 **ISSUE 6** - Instead of Create-Printer-Subscription response returning "notify-lease-time-granted" which is  
113 NOT a Subscription object attribute, it is cleaner for implementation to return the "notify-printer-up-time"  
114 Subscription attribute instead, which is a Subscription object attribute.

115 The client subtracts the "notify-printer-up-time" from the "notify-lease-expiration-time" to get the number  
116 of second remaining in the lease, just as it would for a Get-Subscribe-Attributes response. OK to replace the  
117 "notify-lease-time-granted" with the "notify-printer-up-time" attribute in the response?

118 **Resolution:** We agreed that a subscription response should contain "notify-printer-up-time" and not  
119 "notify-lease-time-granted" in order to have uniformity.

120

121 Page 34, section 8.2.1, lines 1036-1037:

122 **ISSUE 7** - Should "notify-persistence-granted" be a Subscription object attribute too, so that all attributes  
123 returned in a response are Subscription object attributes?

124 **Resolution:** We agreed to add "notify-persistence-granted" as a new subscription object attribute.

125

126 Page 37, section 8.2.4, lines 1086-1089:

127 **ISSUE 8** - Instead of Renew-Subscription response returning "notify-lease-time-granted" which is NOT a  
128 Subscription object attribute, it is cleaner for implementation to return the "notify-printer-up-time"  
129 Subscription attribute instead, which is a Subscription object attribute. Ok to replace the "notify-lease-time-  
130 granted" with the "notify-printer-up-time" attribute in the response?

131 The following issues are not listed in the Notification specification, but were added later during email and  
132 telecon discussions:

133 **Resolution:** Decision is the same as Issue 6, namely "yes".

134

135 Page 18, section 5, Table 2 (and a new 5.n section):

136 **ISSUE 9** - Should we add the "job-id" Subscription Description attribute to the Subscription object? Or is  
137 it better to leave it out, so that implementations that need such a link would do so internally? The client  
138 doesn't need such a link, since when it queries using Get-Subscriptions, it supplies either the "job-id" it  
139 wants Per-Job subscriptions or it omits the "job-id" and gets all of the Per-Printer subscriptions. If we do  
140 add the "job-id" to the Subscription object, what about Per-Printer Subscription objects? See choices in  
141 issue 5a: Don't return "job-id", return "job-id" with a 0 (invalid subscription id value), return 'no-value' out-  
142 of-band value.

143 **Resolution:** No. We could see no reason for a client to need this information because a client can only get  
144 subscriptions by asking for a particular job-id. See issue 17 for more.

145

146 Page 31, section 8.1.1 and Table 7:

147 **ISSUE 10** - Should we remove the last use of collection in the notification spec (see Ira McDonald mail  
148 suggestion), i.e., remove the "job-notify (1setOf collection)" operation attribute in the Job Creation  
149 operations (and Validate-Job)?

150 Alternatives:

151

- 152 a. Don't change. Keep "job-notify (1setOf collection)" as an operation attribute (for the client to  
153 supply and the Printer to support) and 'collection' as a new attribute syntax.
- 154 b. Keep "job-notify (1setOf collection)" as an operation attribute (for the client to supply and  
155 'collection' as a new attribute syntax. However, allow the Printer to be able to support the  
156 notification spec, but NEED NOT support the 'collection'. Instead the Printer would return the  
157 "job-notify" as an unsupported attribute, but would support the notification member attributes. In  
158 this case, the Printer is only supporting one Per-Job subscription in the Job Creation operations.
- 159 c. Remove the 'collection' attribute syntax all together (until another use of it is justified). Instead of  
160 one "job-notify (1setOf collection)" operation attribute in the Job Creation operations (and  
161 Validate-Job), we'd make the 6 member attributes be operation attributes instead  
162  
163 notify-recipient (uri)  
164 notify-events (1setOf type2 keyword)  
165 notify-content-type (mimeMediaType) [See ISSUE 3 which would remove this one]  
166 subscriber-user-data (octetString(63))  
167 notify-charset (charset)  
168 notify-natural-languages (1setOf naturalLanguage)  
169
- 170 d. Same as c but in addition, REQUIRE that an implementation that supports notification also support  
171 the OPTIONAL Hold-Job and Release-Job operations with at least the 'indefinite' value. Then a  
172 client could count on add additional Per-Job subscriptions by submitting the job with "job-hold" =  
173 'indefinite', perform additional Create-Subscription operations (up to the limit of the number of  
174 Per-Job subscriptions supported per job)  
175

176 **Resolution:** No. Use solution "a". We believe it is better to add the general purpose "collection". In the  
177 internet environment we see a potentially greater need for multiple subscriptions per job, e.g. a person  
178 submitting a job might want to receive notification of completion and might want the recipient notified too.  
179 We believe that the collection is a better solution than one requiring a client to hold a job and then to add  
180 job subscriptions, and finally to release the job. All three of these operations are optional.

181

182 Page 20, section 5.2, lines 573-579:

183 **ISSUE 11** - Should the 'job-completed' event which is defined to be job completion, job aborted, and job  
184 canceled be separated into three separate events, i.e., replace the single 'job-completed' event with 'job-  
185 completed', 'job-aborted' and 'job-canceled'? In the current spec, there is no way to subscribe just to aborted

186 or canceled; the 'job-completed' event means all three conditions. On the other hand, breaking them apart  
187 will increase the number of events that are supplied in a subscription. The Notification Recipient could  
188 filter between the three kinds of completions by looking at the "job-state" attribute that is passed in the  
189 Notification content.

190 **Resolution:** No. We could see no reasonable use-case where a person doesn't want to be notified in all three  
191 cases. Normally a person wants to know the job disposition, whatever it is. There might be exceptional case  
192 where someone listens for errors and not normal cases. The client could contain a filtering program for  
193 such cases.

194

195 Pages 28-29, section 7.2, Table 4, "12. trigger-date-time" row:

196 **ISSUE 12** - Shouldn't 12.trigger-date-time be OPTIONAL in 'job-progress', rather than not specified, i.e.,  
197 add "O" to the 'job-progress' column in the 12.trigger-date-time row? The trigger-date-time will be needed  
198 for QUALDOCs, though maybe page level won't be REQUIRED for QUALDOCs.

199 **Resolution:** No. We will wait for a real need before making this change.

200

201 Pages 28-29, section 7.2, Table 4, "15. subscriber-user-data" row:

202 **ISSUE 13** - Shouldn't 15. subscriber-user-data be REQUIRED for 'job-progress', since it may be important  
203 for certain notification methods, i.e., change the <blank> to "R" in the 'job-progress' column for the "15.  
204 subscriber-user-data" row?

205 **Resolution:** Yes. We believe that subscriber-user-data should required for all events if it is to be useful.

206

207 Pages 28-30, section 7.2, Table 4 and section 7.3 Table 5:

208 **ISSUE 14** - Agreed that the following attributes are NOT sent in the 'job-progress' event notification  
209 (because the table entry is blank in the "job-progress" column), since these attributes aren't likely to change  
210 from one page to the next:

211 "printer-name"

212 "job-name"

213 "subscriber-user-name"

214 "job-state"

215 "job-state-reasons"

216

217 **Resolution:** We agreed to delete the first 3 attributes above, but to keep the last two "job-state" and "job-  
218 state-reasons".

219 The following issues were raised at the meeting in Denver.

220

221 **ISSUE 15** – Should we require that a Printer always accept the number of subscriptions that it advertises as  
222 its maximum. If not, how does it inform a user about the rejection of a subscription.

223 **Resolution:** We decided that a printer might give higher priority for jobs than subscriptions. Thus, it might  
224 accept a job and reject some subscriptions. This means that a printer must be capable of rejecting  
225 subscriptions that accompany a job. When a subscription is ignored, all of the attributes in the subscription  
226 are returned in the unsupported attributes. If a printer accepts some of the subscriptions and rejects other  
227 subscriptions, only the attributes of rejected subscriptions are in the unsupported attributes group in the  
228 response.

229 If the job is accepted and one or more subscriptions are rejected, the status code returned is 'successful-ok-  
230 subscription-rejected'. This status code is returned even if other job attributes are unsupported or in conflict.  
231 That is, if a server finds warning that would allow it to return 'successful-ok-subscription-rejected' and

232 either 'successful-ok-ignored-or-substituted-attributes' or 'successful-ok-conflicting-attributes', it must  
233 return 'successful-ok-subscription-rejected'. We should add language in the model document explaining that  
234 if a server finds warning that would allow it to return either "successful-ok-ignored-or-substituted-attributes"  
235 or 'successful-ok-conflicting-attributes', it must return 'successful-ok-conflicting-attributes'. NOTE: This  
236 success code precedence needs an update to the Model and Semantics document.

237 **Rationale:** We want to allow an implementation that wants to keep Subscription objects separately from  
238 Job objects to be able to run out of room for Subscription objects before running out of space for Jobs. So  
239 an implementation can continue to accept jobs, even though it has exhausted space for Subscription objects.

240

241 **Issue 16:** Should create-subscription be mandatory for job subscriptions.

242 **Resolution:** No. Create separate operations for printer subscription creation and job subscription creation  
243 so that a user can determine which are supported. Make printer subscription creation mandatory for a  
244 printer to support and make job subscription creation optional for a printer to support. This decision also  
245 means that a client MUST supply the "job-id" in a Create-Job-Subscription operation and MUST NOT  
246 supply it in a Create-Printer-Subscription operation, which follows a principle we have for other Job versus  
247 Printer operations.

248

249 **Issue 17:** Should the get-subscriptions operation have an option for getting all printer and all job  
250 subscriptions in a single operation?

251 **Resolution:** No. We could see no need to get all. We could barely see a need to get job subscriptions, job  
252 by job. Getting printer subscriptions is important.

253

254 **Issue 18:** What are the implementation requirements for "job-impressions-completed" and "job-media-  
255 sheets-completed" as shown in Table 5.

256 **Resolution:** Make both "Conditionally required" (an "CR") for "job-progress" and "job-completed" events,  
257 i.e., REQUIRED in a 'job-progress' event notification if the implementation supports the corresponding  
258 "job-impressions-completed" or "job-media-sheets-completed" Job attributes. The group believed that  
259 these attributes were the ones mostly likely to be implemented and should be consistent for "job-progress"  
260 and "job-completed".

261

262 **Issue 19:** What do "R" and "O" mean in the tables.

263 **Resolution:** We would like a few sentences to be added to the document to explain "R", "CR", and "O" in  
264 the tables. We assume that "R" means "REQUIRED" if the implementation supports notification, "CR"  
265 means "CONDITIONALLY REQUIRED", i.e. it is required in an event if the printer supports the attribute  
266 in the Job or Printer object, and "O" means OPTIONAL.

267

268 **Issue 20:** Should "subscriber-user-name" be in "job-progress" events.

269 **Resolution:** Yes. But it should be changed from REQUIRED ("R") to OPTIONAL ("O") for all events.

270

271 **Issue 21:** Should we change the Notification Model to allow notification delivery methods that are request  
272 and response (in addition to the current model which has only one-directional notification delivery using the  
273 'application/ipp' operation response format?

274

275

276 **Issue 22:** If the answer to Issue 21 is yes, should we change the format of the notification content using  
277 'application/ipp' to always be a Send-Notification request, whether the scheme returns a response or not?