1
2 # PENDING IPP Issues List - Model only
3
4 **Editor: Carl-Uno Manros and Tom Hastings**
5 File: ipp-issues-list-mod-1.53.doc
6 Directory: ftp://ftp.pwg.org/pub/pwg/ipp/proposed-clarifications/
7 Version: 1.53
8 **Date: October 306, 1998**
9
10 This document contains the PENDING issues related to the IPP/1.0 Model and
11 Semantics, dated June 30, 1998. A few resolutions also affect the IPP/1.0 Transport and
12 Encoding, dated June 30, 1998 (referred to as PRO).
13
14 This document is prepared by the Printer Working Group (PWG), in accordance with the
15 editing rules that apply to PWG documents. The information in this document will be
16 continuously updated and replaced as decided in the meetings, telecons, and e-mail
17 discussions of the PWG. The document is made freely available also to non-members of
18 the PWG, but no guarantee is given that the content of this document is fully correct and
19 consistent with the official documents on IPP from the IETF.
20
21 This version includes questions raised on the IPP DL between July 1 and September 30,
22 1998 including the Bake-Off held September 23-25, 1998.
23
24 *All references are to the June 30, 1998 drafts.*
25
26 The purpose of this document is to collect information about implementation questions
27 and issues against the current IPP draft documents. Allowable questions and issues are
28 about things like suspected errors, inconsistencies, or needs for further clarifications.
29 Questions about extensions or functional changes to the drafts are dealt with in the
30 overall IPP development activities and are outside the scope of this document. Please
31 note that even if a question does get listed, the PWG might decide that it is outside the
32 scope of the IPP Issues List and remove it in a later version.
33
34 A separate IPP Implementer's Guide (IIG) will be developed which contains advice to
35 implementers that supplements the standards track documents. It will contain advice to
36 implementers that goes beyond the exact IPP conformance requirements, e.g. how to
37 ensure interoperability with earlier versions of Internet components, or even early
38 implementations of IPP itself. Section 16 of MOD and most of section 4 of PRO will be
39 moved to the IPP. Also the conformance language of MUST, SHOULD, and MAY will
40 be removed from the IPP. The publication of the IIG may be as an informational RFC
41 along with the other IPP documents, or may remain as a PWG document. Which form of
42 publication is TDB.
43
44 When the disposition of a question or issue in the IPP Issues List is of the form of
45 information suitable for the IIG, rather than clarifications of the IPP standard (MOD or
46 PRO), it will be put into the IIG.

47
48   Each new **Question** on the IPP DL has been listed in a separate table. Added in the table
49   is also one section called **Discussion**, which reflects comments back from other IPP DL
50   participants. When the PWG has come up with an agreed Answer to the Question, it is
51   reflected in the **Answer** section of the table. Before an issue is completely resolved, the
52   exact text for the MOD, PRO, or IIG will be included in the Answer section for review
53   and approval, including which document(s) will be changed.
54
55   When an issue is approved, it is copied to a new document called:
56
57   <u>**APPROVED IPP Issues List - Model only**</u>
58
59   which is available at:
60
61   ftp://ftp.pwg.org/pub/pwg/ipp/approved-clarifications/ipp-appproved-issues-list-mod-
62   1.n.doc
63
64   where n is the version number.

IPP Issues List - Model only

65   TABLE OF CONTENTS
66

82

83

1 Change History for Model and Encoding/Transfer documents

85 We agreed that the Model and Semantics (MOD) and the Encoding/Transfer documents
86 (PRO) should have a change history that lists the substantive changes from the June 30
87 document. It should also contain major clarifications, but not list every minor
88 clarification. This section contains copies of those change histories.

89 **Change History for the IPP Model and Semantics document**

90 The following substantive changes and major clarifications have been made to this
91 document from the June 30, 1998 version based on the interoperability testing that took
92 place September 23-25 1998. These changes are the ones that might affect
93 implementations. Clarifications that are unlikely to affect implementations are not listed.
94 The issue numbers refer to the IPP Issues List.
95

Section	Description
3.2.4	Clarified that an IPP Printer that supports the Create-Job operation MUST handle the situation when a client does not supply Send-Document or Send-URI operations within a one- to four-minute time period. Also clarified that a client MUST send documents in a multi-document job without undue or unbounded delay. (Issue 1.28)

96

2 Model & Semantics

98

Question	**1.28 What MUST an IPP object do if Create-Job never gets an Add-Document or Send-Document with 'last-document' set to 'true'?** Should the IPP object close the job after some period of time and: 1. move the job to the 'aborted' state with the 'aborted-by-system' job-state-reasons value set 2. move the job to the 'pending-held' state (with some new job-state-reason indicating an incomplete job, or 3. move the job to the 'pending' state and print the job? What if the job never had any Add-Document or Send-Document operations, so that the job has no documents? **IPP Bake Off**
Discussion	The IPP object should close the job after some period of time and: 1. For spooling applications - move the job to the 'aborted' state with the 'aborted-by-system' job-state-reasons value set. 2. For non-spooling applications - move the job to the 'pending-held' state with a job-state-reason of "incomplete-job" and an administratively set

time-out (probably somewhere between 30sec and 4 min.).
3. As a fallback - move the job to the 'pending' state and print the job? (A form of natural aging)

These notions should be described in the IIG. This basically addresses system latencies that may occur during the process of performing a create job based job submission. In general, the Create-Job form of submission is intended to flow as a rapid sequence of operations without large discontinuities in time between related operations. We should note the caution that we are defining a tuning attribute, here, and thereby may effect overall system performance. The notion here is that it is not our intent for the sever to keep partially constructed job submissions on hold for long periods of time. We couldn't actual agree on a figure but we expect it to be somewhere between 30 sec to 4 mins. The real number should be determined empirically and information updated in the IIG.

The editor found the following discussion in Section 3.3.1 Send-Document Operation, including a reference to a "multiple-operation-timeout" Printer attribute which has not been defined:

Since the Create-Job and the send operations (Send-Document or Send-URI operations) that follow can occur over arbitrarily long periods of time, each Printer object must decide how long to "wait" for the next send operation. The Printer object OPTIONALLY supports the "multiple-operation-timeout" attribute. This attribute indicates the maximum number of seconds the Printer object will wait for the next send operation. If the Printer object times-out waiting for the next send operation, the Printer object MAY decide on any of the following semantic actions:

1. Assume that the Job is an invalid job, start the process of changing the job state to 'aborted', and clean up all resources associated with the Job. In this case, if another send operation is finally received, the Printer responds with an "client-error-not-possible" or "client-error-not-found" depending on whether or not the Job object is still around when it finally arrives.

2. Assume that the last send operation received was in fact the last document (as if the "last-document" flag had been set to 'true'), close the Job object, and proceed to process it (i.e., move the Job's state to 'pending').

3. Assume that the last send operation received was in fact the last document, close the Job, but move it to the 'pending-held' to allow an operator to determine whether or not to continue processing the Job by moving it back to the 'pending' state.

Each implementation is free to decide the "best" action to take depending on local policy, the value of "ipp-attribute-fidelity", and/or any other piece

<table>
<tr>
<td></td>
<td>of information available to it. If the choice is to abort the Job object, it is possible that the Job object may already have been processed to the point that some media sheet pages have been printed.

From the October 14 telecon minutes:

We discussed that we had forgotten that the June Model and Semantics document contains a "multiple-operations-time-out" Printer Description (see section 4.4.28) that allows the IPP Printer to indicate the length of time before it closes down multi-document jobs that haven't had another operation performed on them.

We agreed to the following:

1. Clarify that "multiple-operations-time-out" is a "minimum", not a promise to close the job after exactly that much time.

2. We reconfirmed that it is a requirement of the IPP Printer to clean up such jobs, not the client.

3. The "multiple-operations-time-out" attribute is an OPTIONAL attribute, but that an IPP Printer MUST support the "multiple- operations-time-out" Printer Description attribute if it supports the Create-Job and Send-Document operations, i.e., if it supports multi-document jobs.

4. The system administrator can set the "multiple-operations-time-out" attribute to any value. He/she is not restricted to a one to four minute value. Instead, the one to four minute value will be the RECOMMENDED default value for this attribute.

ACTION ITEM (Tom): Update the proposed text for Issue 1.28 for another two week review.</td>
</tr>
<tr>
<td>Answer 9/30/1998</td>
<td>Replace the last two paragraphs and three actions in MOD 3.3.1 with:

Since the Create-Job and the send operations (Send-Document or Send-URI operations) that follow could occur over an arbitrarily long periods of time for a particular job, a client MUST send another send operation within an IPP Printer implementation-defined time interval after the receipt of the previous request for the job. An IPP object MUST recover from an errant client that does not supply a send operation with a "last-document" set to 'true', sometime within this implementation-defined time interval after the most recent Create-Job or send operation has been received for the job. The implementation-defined time period MUST be within one to four minutes.

Such recovery MAY include any of the following recovery</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td>

actions:

      1. Assume that the Job is an invalid job, start the process of changing the job state to 'aborted', adding the 'aborted-by-system' value to the job's "job-state-reasons" attribute, if supported, and clean up all resources associated with the Job. In this case, if another send operation is finally received, the Printer responds with an "client-error-not-possible" or "client-error-not-found" depending on whether or not the Job object is still around when the send operation finally arrives.

      2. Assume that the last send operation received was in fact the last document (as if the "last-document" flag had been set to 'true'), close the Job object, and proceed to process it (i.e., move the Job's state to 'pending').

      3. Assume that the last send operation received was in fact the last document, close the Job, but move it to the 'pending-held' to allow an operator to determine whether or not to continue processing the Job by moving it back to the 'pending' state.

Each implementation is free to decide the "best" action to take depending on local policy, the value of "ipp-attribute-fidelity", whether any documents have been added, whether the implementation spools jobs or not, and/or any other piece of information available to it. If the choice is to abort the Job object, it is possible that the Job object may already have been processed to the point that some media sheet pages have been printed.

</td>
</tr>
</table>

99

100

Question	**1.33 Equality between different syntaxes?** When checking for equality or containment (e.g., "IF NOT in the Printer object's 'job-hold-until-supported' attribute ...") is value type considered? Is a value of type 'nameWithoutLanguage' considered equal to a value of type 'nameWithLanguage' if the default language for the context of the 'nameWithoutLanguage' value is the same as the language explicit in the 'nameWithLanguage' value? Can a 'name' match a 'keyword'? IF a 'nameWithoutLanguage' value in the appropriate natural language context CAN match a 'nameWithLanguage' value, is there any harm (other than a negligible increase in network bandwidth consumption) in an application promoting ALL 'name' and 'text' attribute values to 'nameWithLanguage' and 'textWithLanguage' values? **Carl Kugler**
Discussion	When checking for equality or containment (e.g., "IF NOT in the Printer object's 'job-hold-until-supported' attribute ...") is value type considered? Is a value of type 'nameWithoutLanguage' considered equal to a value of type 'nameWithLanguage' if the default language for the context of the 'nameWithoutLanguage' value is the same as the language explicit in the 'nameWithLanguage' value? (Yes, under these circumstances, but not if the defaults are different because then the semantics implied by the values may not match). Can a 'name' match a 'keyword'? (Yes, possibly, under these circumstances but not in general). (Need clarification on the question). IF a 'nameWithoutLanguage' value in the appropriate natural language context CAN match a 'nameWithLanguage' value, is there any harm (other than a negligible increase in network bandwidth consumption) in an application promoting ALL 'name' and 'text' attribute values to 'nameWithLanguage' and 'textWithLanguage' values? No harm... Another way to state the question is if a client sends an attribute then queries it back must the tagging be identical in the response... We said no. Keywords are intended to be localized by the client. Keywords on the wire are not localized, however. If the server also supports some administratively defined names, the client realizes these are already localized by the server. Administrator has defined a name and the client can supply that either with or without language. From the October 7 telecon minutes: Reviewing the proposed Answer section of Issue 1.33 in the Issue list,

	V1.3, we agreed: 1. change the case-insensitive matching rules for attributes with the 'name' attribute syntax from SHOULD to MUST, since such attributes are completely within the province of IPP, and are not the subject of other standards and are not handled by any off-the-shelf code conforming to other standards. 2. Since there are currently no 'text' matching attributes specified in MOD, that MOD would be silent on any rules for matching 'text' attributes. So the proposed resolution to Issue 1.33 only applies to the 'name' attribute syntax. 3. to remove any statement about any other equivalencies, such as accent insensitiveness or other character equivalencies, such as Unicode composed accented letters versus composite accented letters. 4. change from MAY to SHOULD that a language without a country matches a language with a country.
Answer 10/30/1998	Don't change 'text', since 'text' isn't compared. Only add to the 'name' attribute syntax: 4.1.2.3 Add sections about comparing nameWithLanguage and nameWithoutLanguage indicating that the explicit language MUST match the implicit language. A keyword value never matches either type of name value, even if the language for the name value is 'en-us'. (Issue 1.33 and 1.34) The following text is to be added to make a new section under 4.1.2 'name': 4.1.2.3 Matching 'nameWithLanguage' and 'nameWithoutLanguage' For purposes of matching 'name' values for equality in job validation, where a client-supplied value for attribute "xxx" is checked to see if the value is among the values of the Printer's corresponding "xxx-supported" attribute, the following match criteria apply: 1. The attribute syntax and value of "xxx" supplied by the client MUST be identical to the attribute syntax and value of one of the values of the corresponding Printer's "xxx-supported" attribute. For example, the client-supplied 'keyword' 'iso-a4-white' does not match the Printer's 'name' 'iso-a4-white', even if the Printer's "natural-language-configured" is 'us-en'.

9

2. For purposes of matching 'name' attributes, the attribute value comparison SHOULD include a case-insensitive algorithm.

3. For purposes of matching 'name' attributes, the implicit or explicit natural language of the "xxx" value supplied by the client MUST be the same as the implicit or explicit natural language of the Printer's "xxx-supported" attribute. For example, a client-supplied nameWithoutLanguage value with an 'en' "attributes-natural-language" operation attribute will match either a Printer's "xxx-supported value which is (1) 'en' nameWithLanguage or (2) nameWithoutLanguage with an 'en' "natural-language-configured". Similarly, a client-supplied 'en' nameWithLanguage value will match either a Printer's "xxx-supported value which is (1) 'en' nameWithLanguage or (2) nameWithoutLanguage with an 'en' "natural-language-configured".

4. An attribute value that has a country part of the natural language SHOULD match an attribute value that has no country part. So a client-supplied 'en' SHOULD match a Printer's 'en-us' or 'en-gb'. Similarly, a client's 'en-us' SHOULD match a Printer's 'en'. However, two attribute values that both have a country part that is different SHOULD NOT match. So a client-supplied 'en-gb' SHOULD NOT match a Printer's 'en-us'.

101

102

Question	**1.34 Equality between "natural language" tags?** Is natural language considered when comparing 'name' attributes (e.g., "job-originating-user-name", "media", "job-hold-until-supported")? [Assertion: ALL 'text' and 'name' attributes have an associated natural language, either explicitly or implicitly.] If so, how strict is the comparison? Does "en" match "en-us", for example? **Carl Kugler**
Discussion	
Answer 9/30/1998	If the country part of the natural language are both present and differ then they don't match. If one country part is omitted and the other is explicit, then ~~whether they match depends on implementation~~they SHOULD match. See answer to 1.33.

103

Question	**1.46 NLO 2 of 4: Clarification that Natural Language Override MAY be used redundantly** The purpose of this clarification is to explicitly allow use of the Natural Language Override in situations where implementers thought it couldn't be used. Therefore, this clarification should not force any existing conforming implementations to change. **Carl Kugler and Bob Herriot**
Discussion	== = Please reply to this e-mail message if there is any disagreement = on this clarification. If no disagreements are returned by Monday, = November 2, it will be considered an agreed clarification. == Note: that the votes on e-mail messages (3 of 4) and (4 of 4) may remove the need for this (2 of 4) clarification. But please comment on these clarifications assuming that the changes specified in the votes do NOT happen. The current text in Section 3.1.4.1 Request Operation Attributes, 5th paragraph of "attributes-natural-language says: For any 'text' or 'name' attribute in the request that is in a different natural language than the value supplied in the "attributes-natural-language", the client MUST use the Natural Language Override mechanism (see sections 4.1.1.2 and 4.1.2.2) for each such attribute value supplied. The clarification is to add the following sentence to the end of the paragraph:

	The client MAY use the Natural Language Override mechanism even when the value is in the same natural language. The 7th paragraph says: Whenever any client queries the Job object's "job-name" attribute, the IPP object returns the attribute as stored and uses the Natural Language Override mechanism to specify the natural language, if it is different from that reported in the "attributes-natural-language" operation attribute of the response. The clarification is to add the following sentence: The IPP object MAY use the Natural Language Override mechanism even when the value is in the same natural language. The last paragraph of 3.1.4.2 contains the sentence: For any 'text' or 'name' attribute or status message in the response that is in a different natural language than the value returned in the "attributes-natural-language" operation attribute, the IPP object MUST use the Natural Language Override mechanism (see sections 4.1.1.2 and 4.1.2.2) on each attribute value returned. The clarification is to add the same following sentence: The IPP object MAY use the Natural Language Override mechanism even when the value is in the same natural language.
Answer 10/	

104
105

Question	**1.47 NLO 3 of 4: Vote to simplify Get-Jobs** This mail message proposes a change in the specification of Get-Jobs to remove an extra level of Natural Language Override at the job level. With this change Get-Jobs would be handled the same as any other operation with respect to the Natural Language Override mechanism at the attribute level. **Bob Herriot and Carl Kugler**
Discussion	** * The proposal to vote on is to delete the indicated paragraph * below from Section 3.2.6.2 Get-Jobs Response. *

* Please indicate your acceptance or rejection of this proposal
* on the mailing list by Monday, Nov 2.
****

This change will affect implementations that correctly implement the June 1998 Mode and Semantics specification. However, we suspect that many implementations may have ignored this feature, so that deleting this paragraph will have no impact on them. Implementers, is this suspicion correct?

Background:

Currently, Section 3.2.6.2 Get-Jobs Response contains the following paragraph:

> For any job submitted in a different natural language than the natural language that the Printer object is returning in the "attributes-natural-language" operation attribute in the Get-Jobs response, the Printer MUST indicate the submitted natural language by returning the Job object's "attributes-natural-language" as the first Job object attribute, which overrides the "attributes-natural-language" operation attribute value being returned by the Printer object. If any returned 'text' or 'name' attribute includes a Natural Language Override as described in the sections 4.1.1.2 and 4.1.2.2, the Natural Language Override overrides the Job object's "attributes-natural-language" value and/or the "attributes-natural-language" operation attribute value.

From the October 28 telecon, Bob Herriot wrote the following fallback proposal, in case we do not get a clear decision either way on the vote for NLO 3 of 4:

Subj: IPP> MOD -(vote clarification) NLO 3 of 4: Vote to simplify Get-Jobs

Re: elimination of the paragraph defined below (from Section 3.2.6.2 of Get-Jobs Response) so that attributes-natural-language is no longer used as a language override in a Get-Jobs response.

In today's teleconference, we decided that we could not make a well informed decision on this issue without test results from the IPP implementations. Xerox hopes to have a test suite by next week that we can use to test this feature.

We can eliminate the feature from IPP 1.0 if

	a) test results show that no implementation supports the feature, or if b) the implementors of those implementations that support the feature are willing to eliminate the feature. If some implementations must continue to support this feature, then a fallback is to change the "MUST" in the paragraph below to a "MAY" for IPP 1.0. Then servers are allowed to omit support of this feature, but clients must be able to process Get-Jobs responses with this feature. This change does not invalidate any implementations that follow the June 30 specs. However, it does change the intent, and becomes the first step in deprecating this feature. --- Currently, Section 3.2.6.2 Get-Jobs Response contains the following paragraph: For any job submitted in a different natural language than the natural language that the Printer object is returning in the "attributes-natural-language" operation attribute in the Get-Jobs response, the Printer MUST indicate the submitted natural language by returning the Job object's "attributes-natural-language" as the first Job object attribute, which overrides the "attributes-natural-language" operation attribute value being returned by the Printer object. If any returned 'text' or 'name' attribute includes a Natural Language Override as described in the sections 4.1.1.2 and 4.1.2.2, the Natural Language Override overrides the Job object's "attributes-natural-language" value and/or the "attributes-natural-language" operation attribute value.
Answer 10/	

106
107

Question	**1.48 NLO 4 of 4: Vote to always use the Natural Language Overrid e mechanism** This mail messages proposes to remove the 'textWithoutLanguage' and 'nameWithoutLanguage' attribute syntaxes and require all 'text' and 'name' attributes to always explicitly include the natural language using the 'textWithLanguage' and 'nameWithLanguage' syntaxes. **Carl Kugler**
Discussion	** * The proposal to vote on is to require all attributes to always * use the 'textWithLanguage' and 'nameWithLanguage' forms * and to delete the 'textWithoutLanguage' and * 'nameWithoutLanguage' forms.

*
_
* Please indicate your acceptance or rejection of this
* proposal on the mailing list by Monday, Nov 2.
****

This change will affect implementations that correctly implement the June 1998 Mode and Semantics specification. Implementations that only support the 'textWithoutLanguage' and 'nameWithoutLanguage' would need to be changed to conform to either the June specification or this proposal (and changing to this proposal would be easier than the June specification which requires supporting both forms of 'text' and both forms of 'name').
Background:

Currently requests and responses that supply 'text' and 'name' attributes in a different natural language than that supplied for the request or response as a whole as indicated in the "attributes-natural-language" Operation attribute MUST include the different natural language explicitly as an override (and MAY include it explicitly even when they are the same -- according to the NLO 2 of 4 clarification).

This proposal is to change the Natural Language Override mechanism so that the 'text' attribute syntax is only 'textWithLanguage' and the 'name' attribute syntax is only 'nameWithLanguage'. In other words, each 'text' and 'name' attribute would always contain the natural language explicitly as part of the value. (The Encoding and Transport specification - PRO - specifies that 'textWithLanguage' and 'nameWithLanguage' values MUST be encoded as 2 octets of length, the natural-language string, 2 octets of length, and the text or name value.)

Eliminating one of the two forms of 'text' and one of the two forms of 'name' attribute syntax will simplify comparison in job validation, since the "xxx" attribute syntax code would have to match the corresponding "xxx-supported".

The PRO document would simply delete the 'textWithoutLanguage' and 'nameWithoutLanguage' attribute syntaxes.

This proposal does not change any other parts of the Model:

1. The "attributes-natural-language" operation attribute in requests MUST still be supplied by the client to indicate its preference for natural language to be returned in responses as currently specified in Section 3.1.4.1 and 3.2.1.1.

Rationale: So that an implementation that implements the OPTIONAL "status-message" response attribute will know which natural language to use.

2. For create operations, the IPP Printer MUST still copy the "attributes-natural-language" operation attribute supplied by the client to the job object as currently specified in Section 3.2.1.1.

Rationale: Subsequent communication with the submitting user, such as operator messages, notification using e-mail, and the job-sheets MAY want to use the natural language of the job submitter.

3. All responses MUST return the "attributes-natural-language" operation attribute as specified in 3.1.4.2, though it no longer has any effect on the interpretation of any of the returned attributes.

 Rationale: no need to change this behavior, since all implementations seem to be doing it. Removing it would save only 37-40 octets per response.

From the October 28 telecon, Bob Herriot wrote the following fallback proposal, in case we do not get a clear decision either way on the vote for NLO 3 of 4:

Subj: NLO 4 of 4: Vote to always use the Natural Language Override mechanism

Re: elimination of the data types textWithoutLanguage and nameWithoutLanguage so that text and name values in IPP always include their natural language.

In today's teleconference, we decided that we could not make a well informed decision on this issue without test results from the IPP implementations. Xerox hopes to have a test suite by next week that we can use to test this feature.

We can eliminate the feature from IPP 1.0 if

a) test results show that no implementation fully supports the feature, or if b) the implementors of those implementations that support the feature are willing to eliminate the feature.

I expect that condition a) fails because some implementations do support it. But it may also be the case that some implementations don't full implement this feature.

<table>
<tr><td></td><td>If some implementations must continue to support this feature, then a fallback is to reword IPP 1.0 to state that senders (of client requests and server reponses) SHOULD always include the language with a text or name value (i.e. send textWithLanguage rather than textWithoutLanguage, and nameWithLanguage rather than nameWithoutLanguage), receivers (of requests on servers and responses on clients) MUST be able to convert textWithoutLanguage and nameWithoutLanguage into their equivalent textWithLanguage and nameWithLangage using the override rules.

The rule for receivers is unchanged from the June 30 document, though the wording may be different. This change does not invalidate any implementations that follow the June 30 specs. It does change the intent, and becomes the first step in deprecating this feature.</td></tr>
<tr><td>**Answer 10/**</td><td></td></tr>
</table>

108
109

Question	**1.49 Standard values are keywords, not names** The attributes with attribute syntax (keyword | name) list standard values, but don't specify whether those values are keywords or names. **Tom Hastings**
Discussion	Keywords are language-independent and were the intent of the standard values. I suggest that they are keywords, not names. Names are for administrators to define.
Answer 10/27/1998	For every attribute that list values, add the word "keyword" or "enum" in between the words "standard" and "values" for the phrase at the beginning of the list. So for example, "job-hold-until" change "Standard values for named time periods are:" to "Standard keyword values for named time periods are:"

110
111

Question	**1.50 What are the errors for each operation?** It isn't clear what condition(s) cause which error codes to be returned for each operation **Bob Herriot**
Discussion	Need to add one line description of each of the error codes and the reason they are used as part of each operation description. ACTION ITEM (Tom Hastings and Bob Herriot): work out a proposal for each status code for each operation.
Answer 10/	

112
113

Question	**1.51 Can Get-Jobs redundantly contain job-level NLO?** If we decide to keep the job-level NLO in Get-Jobs as in the June draft, we have the following related issue to redundant natural language override at the attribute level (see issue 1.46): Can a Get-Jobs response redundantly return a job-level "attributes-natural-language" (when not requested) which has the same natural language as the job? If yes, then it may be simpler for IPP Printer implementations to ALWAYS add the "attributes-natural-language" in the returned Job Attributes (first), whether the job is in that natural language or not. Since a client is supposed to be able to deal with job-level NLO according to the June drafts, this redundancy would not be adding any more complexity to the clients. Comments? **Tom Hastings**
Discussion	
Answer 10/	

114

Question	**1.52 Can Get-Jobs attribute-natural-language occur twice?** If we decide to keep the job-level NLO in Get-Jobs as in the June draft, what happens if a client explicitly requests a job's 'attributes-natural-language' by including it as one of the values of the "requested-attributes" operation attribute and the implementation also has to return a NLO at the job level by returning the job's "attributes-natural-language" as the first Job attribute because the job is in a different natural language than the response? Possibilities for "attributes-natural-language" Job attribute in the Get-Jobs response: 1. MUST occur only once and be first 2. SHOULD occur only once and be first 3. MAY occur twice, once first and the other with the same value any where. **Tom Hastings**
Discussion	
Answer 10/	

115
116