

1. 1
1. 2 **PENDING IPP Issues List - Model only**
1. 3

1. 4 **Editor: Carl-Uno Manros and Tom Hastings**

1. 5 File: ipp-issues-list-mod-1.73.doc

1. 6 **Directory:** <ftp://ftp.pwg.org/pub/pwg/ipp/proposed-clarifications/>

1. 7 Version: 1.73

1. 8 **Date:** ~~October~~ **November 56, 1998**
1. 9

1. 10 This document contains the **PENDING** issues related to the IPP/1.0 Model and
1. 11 Semantics, dated June 30, 1998. ~~A few resolutions also affect the IPP/1.0 Transport and~~
1. 12 ~~Encoding, dated June 30, 1998 (referred to as PRO).~~
1. 13

1. 14 This document is prepared by the Printer Working Group (PWG), in accordance with the
1. 15 editing rules that apply to PWG documents. The information in this document will be
1. 16 continuously updated and replaced as decided in the meetings, telecons, and e-mail
1. 17 discussions of the PWG. The document is made freely available also to non-members of
1. 18 the PWG, but no guarantee is given that the content of this document is fully correct and
1. 19 consistent with the official documents on IPP from the IETF.
1. 20

1. 21 This version includes questions raised on the IPP DL between July 1 and September 30,
1. 22 1998 including the Bake-Off held September 23-25, 1998.
1. 23

1. 24 *All references are to the June 30, 1998 drafts.*
1. 25

1. 26 The purpose of this document is to collect information about implementation questions
1. 27 and issues against the current IPP draft documents. Allowable questions and issues are
1. 28 about things like suspected errors, inconsistencies, or needs for further clarifications.
1. 29 Questions about extensions or functional changes to the drafts are dealt with in the
1. 30 overall IPP development activities and are outside the scope of this document. Please
1. 31 note that even if a question does get listed, the PWG might decide that it is outside the
1. 32 scope of the IPP Issues List and remove it in a later version.
1. 33

1. 34 A separate IPP Implementer's Guide (IIG) will be developed which contains advice to
1. 35 implementers that supplements the standards track documents. It will contain advice to
1. 36 implementers that goes beyond the exact IPP conformance requirements, e.g. how to
1. 37 ensure interoperability with earlier versions of Internet components, or even early
1. 38 implementations of IPP itself. Section 16 of MOD and most of section 4 of PRO will be
1. 39 moved to the IPP. Also the conformance language of MUST, SHOULD, and MAY will
1. 40 be removed from the IPP. The publication of the IIG may be as an informational RFC
1. 41 along with the other IPP documents, or may remain as a PWG document. Which form of
1. 42 publication is TBD.
1. 43

1. 44 When the disposition of a question or issue in the IPP Issues List is of the form of
1. 45 information suitable for the IIG, rather than clarifications of the IPP standard (MOD or
1. 46 PRO), it will be put into the IIG.

IPP Issues List - Model only

1. 47
1. 48 Each new **Question** on the IPP DL has been listed in a separate table. Added in the table
1. 49 is also one section called **Discussion**, which reflects comments back from other IPP DL
1. 50 participants. When the PWG has come up with an agreed Answer to the Question, it is
1. 51 reflected in the **Answer** section of the table. Before an issue is completely resolved, the
1. 52 exact text for the MOD, PRO, or IIG will be included in the Answer section for review
1. 53 and approval, including which document(s) will be changed.

1. 54
1. 55 When an issue is approved, it is copied to a new document called:

1. 56
1. 57 **AGREED Resolutions to the IPP Issues List - Model only**

1. 58
1. 59 which is available at:

1. 60
1. 61 ftp://ftp.pwg.org/pub/pwg/ipp/approved-clarifications/ipp-agreed-fixes-yymmdd.*

1. 62
1. 63 where yymmdd is the year month day of the file.

1. 64

1. 65 TABLE OF CONTENTS

| | | |
|-------|---|---|
| 1. 66 | | |
| 1. 67 | 1 | Change History for Model and Encoding/Transfer documents..... 3 |
| 1. 68 | | Change History for the IPP Model and Semantics document..... 3 |
| 1. 69 | 2 | Model & Semantics 4 |
| 1. 70 | | <u>1.19 REVISITED - What error to return when an unsupported charset is requested.</u> |
| 1. 71 | | 1.28 What MUST an IPP object do if Create-Job never gets an Add-Document or |
| 1. 72 | | Send-Document with 'last-document' set to 'true'? 4 |
| 1. 73 | | 1.33 Equality between different syntaxes? 9 |
| 1. 74 | | 1.34 Equality between “natural language” tags? 12 |
| 1. 75 | | 1.46 NLO 2 of 4: Clarification that Natural Language Override MAY be used |
| 1. 76 | | redundantly 12 |
| 1. 77 | | 1.47 NLO 3 of 4: Vote to simplify Get-Jobs 15 |
| 1. 78 | | 1.48 NLO 4 of 4: Vote to always use the Natural Language Overrid e mechanism 17 |
| 1. 79 | | <u>1.49 Standard values are keywords, not names 19</u> |
| 1. 80 | | 1.50 What are the errors for each operation? 20 |
| 1. 81 | | 1.51 Can Get-Jobs redundantly contain job-level NLO?..... 21 |
| 1. 82 | | 1.52 Can Get-Jobs attribute-natural-language occur twice? 21 |
| 1. 83 | | <u>1.53 Should we make document-format-supported REQUIRED for directories? 20</u> |
| 1. 84 | | <u>1.54 Can't put one staple through multiple documents that start on new sheets 21</u> |

IPP Issues List - Model only

1. 85

1. 86 **1 Change History for Model and Encoding/Transfer documents**

1. 87 We agreed that the Model and Semantics (MOD) and the Encoding/Transfer documents
1. 88 (PRO) should have a change history that lists the substantive changes from the June 30
1. 89 document. It should also contain major clarifications, but not list every minor
1. 90 clarification. This section contains copies of those change histories.

1. 91 **Change History for the IPP Model and Semantics document**

1. 92 The following substantive changes and major clarifications have been made to this
1. 93 document from the June 30, 1998 version based on the interoperability testing that took
1. 94 place September 23-25 1998. These changes are the ones that might affect
1. 95 implementations. Clarifications that are unlikely to affect implementations are not listed.
1. 96 The issue numbers refer to the IPP Issues List.

1. 97

| Section | Description |
|-------------------------|---|
| 3.1.4.1 | Clarified Section 3.1.4.1 Request Operation Attributes that a client MAY use the attribute level natural language override (text/nameWithLanguage) redundantly in a request. (Issue 1.46) |
| 3.1.4.2 | Clarified Section 3.1.4.2 Response Operation Attributes that an IPP object MAY use the attribute level natural language override (text/nameWithLanguage) redundantly in a response. (Issue 1.46) |
| 3.2.6.2 | Deleted the job-level natural language override from Section 3.2.6.2 Get-Jobs Response. (Issue 1.47) |
| 3.3.1 | 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) |
| 4.1.2.3 | Added that nameWithoutLanguage plus the implicit natural language matches nameWithLanguage, if the values and natural languages are the same. Also added that keyword never matches nameWithLanguage or nameWithoutLanguage. Clarified that if both have countries, that the countries SHOULD match as well. If either do not, then the country field SHOULD be ignored. (Issues 1.33 and 1.34) |
| 4.2.* | Added brief descriptions of each status code to each operation description. (Issue 1.50) |
| 4.2.4 | Added the single-document-new-sheet' value to Section 4.2.4 multiple-document-handling. (Issue 1.54) |
| 4.4.28 | Clarified that the "multiple-operation-time-out" SHOULD be between 30 and 240 seconds, though the administrator can set values outside this |

IPP Issues List - Model only

| | |
|------------------|--|
| | <u>range. (Issue 1.28)</u> |
| <u>5.1</u> | <u>Clarified Client Conformance that if a client supports an attribute of 'text' or 'name' attribute syntax, that it MUST support both the nameWithoutLanguage and the nameWithLanguage forms. (Issue 1.48)</u> |
| <u>14.1.4.14</u> | <u>Clarified that the 'client-error-charset-not-supported' SHOULD take precedence over all other errors, unless the request syntax is so bad, that the client's requested charset cannot be determined. (Issue 1.19 REVISITED)</u> |
| <u>17</u> | <u>Changed "document-format-supported" to REQUIRED for directory schema, to agree with Printer object. (Issue 1.53)</u> |

1. 98

1. 99 2 Model & Semantics

1. 100

| | |
|--------------------------|---|
| <u>Question</u> | <p><u>1.19 REVISITED - What error to return when an unsupported charset is requested.</u></p> <p><u>What is the precedence between the 'client-error-bad-syntax' and 'client-error-charset-not-supported'? What if both errors actually occur in a request?</u></p> <p style="text-align: center;"><u>Carl Kugler</u></p> |
| <u>Discussion</u> | <u>This is important for the SLP Printer template discussion and review, so that it is mandatory for directory entries. Also which document formats that a printer supports is pretty important to a user in order to select a printer.</u> |
| <u>Answer 11/08/1998</u> | <p><u>Add the following sentences to section 14.1.4.14 'client-error-charset-not-supported':</u></p> <p><u>This error SHOULD take precedence over any other error, so that the client will know that the returned charset is not the one requested. Therefore, the IPP object SHOULD endeavor to determine the "attribute-charset" operation attribute in the request. Of course, if the syntax of the request is so bad that the IPP object cannot find the "attributes-charset", then the IPP object has no choice but to return the 'client-error-bad-syntax' status code.</u></p> |

1. 101

1. 102

| | |
|-----------------|--|
| <u>Question</u> | <p><u>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'?</u></p> <p>Should the IPP object close the job after some period of time and:</p> <ol style="list-style-type: none"> 1. move the job to the 'aborted' state with the 'aborted-by-system' job-state-reasons value set |
|-----------------|--|

IPP Issues List - Model only

| | |
|--------------------------|--|
| | <p>2. move the job to the 'pending-held' state (with some new job-state-reason indicating an incomplete job, or</p> <p>3. move the job to the 'pending' state and print the job?</p> <p>What if the job never had any Add-Document or Send-Document operations, so that the job has no documents?</p> <p style="text-align: center;">IPP Bake Off</p> |
| <p>Discussion</p> | <p>The IPP object should close the job after some period of time and:</p> <ol style="list-style-type: none"> 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) <p>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.</p> <p>The editor found the following discussion in Section 3.3.1 Send-Document Operation, including a reference to thea "multiple-operation-timeout" Printer attribute which has not been is defined in Section 4.4.28 of the June Model spec:</p> <p style="padding-left: 40px;">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:</p> <ol style="list-style-type: none"> 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 |

IPP Issues List - Model only

finally received, the Printer responds with a "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 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

IPP Issues List - Model only

| | |
|---|---|
| <p>Answer 9/3011/2/19 98</p> | <p><u>another two week review.</u></p> <p>Replace the last two paragraphs and three actions in MOD 3.3.1 (<u>see Discussion above for the current text</u>) with:</p> <p>Since the Create-Job and the send operations (Send-Document or Send-URI operations) that follow can<u>could</u> occur over <u>an</u> arbitrarily long periods of time <u>for a particular job, a client MUST send another send operation within an IPP Printer defined minimum time interval after the receipt of the previous request for the job. If a Printer object supports multiple document jobs, the Printer object</u> OPTIONALLY MUST <u>supports</u> the "multiple-operation-time-out" attribute (<u>see section 4.4.28</u>). This attribute indicates the maximum<u>minimum</u> number of seconds the Printer object will wait for the next send operation <u>before taking some recovery action.</u></p> <p><u>An IPP object MUST recover from an errant client that does not supply a send operation with a "last-document" set to 'true', sometime after the minimum time interval specified by the Printer object's "multiple-operation-time-out" attribute. Each Printer object must decide how long to "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</u>Such recovery <u>MAY include any of the following recovery</u> actions:</p> <ol style="list-style-type: none"> 1. Assume that the Job is an invalid job, start the process of changing the job state to 'aborted', <u>add the 'aborted-by-system' value to the job's "job-state-reasons" attribute (see section 4.3.8), if supported,</u> 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 <u>the send operation</u> 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' <u>and add the 'submission-interrupted' value to the job's "job-state-reasons" attribute (see section 4.3.8), if supported. This action</u> to <u>allows the user or</u> an operator to determine whether or not to continue processing the Job by moving it back to the 'pending' state <u>or to cancel the job.</u> <p>Each implementation is free to decide the "best" action to take depending on local policy, the value of "ipp-attribute-fidelity",</p> |
|---|---|

IPP Issues List - Model only

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.

Change the description for Section 4.4.28 "multiple-operation-time-out" from:

4.4.28 multiple-operation-time-out (integer(1:MAX))

This Printer attributes identifies how long (in seconds) the Printer object waits for additional Send-Document or Send-URI operations to follow a still-open multi-document Job object before taking one of the actions indicated in section 3.3.1.

to:

4.4.28 multiple-operation-time-out (integer(1:MAX))

This Printer attributes identifies how long the minimum time (in seconds) that the Printer object waits for additional Send-Document or Send-URI operations to follow a still-open multi-document Job object before taking one of the actions indicated in section 3.3.1.

It is RECOMMENDED that vendors supply a value for this attribute that is between 60 and 240 seconds. A system administrator MAY set this attribute to any value, including values outside this range.

1. 103

IPP Issues List - Model only

1. 104

| | |
|-------------------|--|
| Question | <p><u>1.33 Equality between different syntaxes?</u></p> <p>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'?</p> <p>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?</p> <p>Carl Kugler</p> |
| Discussion | <p>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).</p> <p>Can a 'name' match a 'keyword'? (Yes, possibly, under these circumstances but not in general). (Need clarification on the question).</p> <p>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?</p> <p>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.</p> <p>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.</p> <p>Administrator has defined a name and the client can supply that either with or without language.</p> <p><u>From the October 7 telecon minutes:</u> <u>Reviewing the proposed Answer section of Issue 1.33 in the Issue list,</u></p> |

IPP Issues List - Model only

| | |
|--|---|
| | <p><u>V1.3, we agreed:</u></p> <ol style="list-style-type: none"><u>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.</u><u>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.</u><u>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.</u><u>4. change from MAY to SHOULD that a language without a country matches a language with a country.</u> |
| <p><u>Answer</u> <u>10/30/1998</u></p> | <p><u>Don't change 'text', since 'text' isn't compared. Only add to the 'name' attribute syntax:</u></p> <p><u>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)</u></p> <p><u>The following text is to be added to make a new section under 4.1.2 'name':</u></p> <p><u>4.1.2.3 Matching 'nameWithLanguage' and 'nameWithoutLanguage'</u></p> <p><u>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:</u></p> <ol style="list-style-type: none"><u>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 'en-us'.</u> |

IPP Issues List - Model only

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'.

1. 105

IPP Issues List - Model only

1. 106

| | |
|---|---|
| <p>Question</p> | <p><u>1.34 Equality between “natural language” tags?</u></p> <p>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?</p> <p>Carl Kugler</p> |
| <p>Discussion</p> | |
| <p>Answer 9/30/1998</p> | <p>If the country part of the natural language <u>are both present and</u> differ then they don't match. If one country part is omitted and the other is explicit, then whether they match depends on implementation<u>they SHOULD match</u>. See answer to 1.33.</p> |

1. 107

| | |
|---------------------------------|---|
| <p><u>Question</u></p> | <p><u>1.46 NLO 2 of 4: Clarification that Natural Language Override MAY be used redundantly</u></p> <p><u>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.</u></p> <p><u>Carl Kugler and Bob Herriot</u></p> |
| <p><u>Discussion</u></p> | <p>*****</p> <p><u>* Please reply to this e-mail message if there is any disagreement</u> <u>* on this clarification to allow a request or response to supply attribute</u> <u>* NLO even when it isn't needed, i.e., be able to supply the</u> <u>* text/nameWithLanguage with the same natural language as in the</u> <u>* "attributes-natural-language" operation attribute.</u> <u>* If no disagreements are returned by Monday,</u> <u>* November 2, it will be considered an agreed clarification.</u> *****</p> <p><u>Note: that the votes on e-mail messages (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.</u></p> <p><u>The current text in Section 3.1.4.1 Request Operation Attributes, 5th paragraph of "attributes-natural-language says:</u></p> <p><u>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.</u></p> |

IPP Issues List - Model only

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.

One problem with this clarification is that if an implementation starts to return nameWithLanguage, but the client doesn't support accepting that form, since it never generates that form, there will be a lack of interoperability.

I've talked to several implementers who are reluctant to take advantage of this clarification for fear the some clients will not be able to accept the nameWithLanguage form.

For example, the client supplies the "job-name" operation attribute using nameWithoutLanguage, but the implementation returns it using nameWithLanguage. If the client just blindly displays the value, it will be

IPP Issues List - Model only

| | |
|---|---|
| | <p><u>corrupted, since the value has two binary numbers and the natural language as well as the actual job name text.</u></p> <p><u>Because we don't have a test tool that tests clients, we can't verify that the clients will be able to accept nameWithLanguage on any attribute whose attribute syntax is 'name'.</u></p> <p><u>Even if the client only supports one natural language, it could test itself with an IPP object that is configured for a different natural language, because then that IPP object would be forced into returning nameWithLanguage. Of course, if all implementations are only supporting en-us, then even that test is impossible.</u></p> <p><u>So before implementations start taking advantage of this proposed clarification, we need to verify that clients are conforming by supporting accepting in a response:</u></p> <ol style="list-style-type: none"><u>1. BOTH the nameWithoutLanguage and the nameWithLanguage forms for 'name' attributes</u><u>2. BOTH the textWithoutLanguage and the textWithLanguage forms for 'text' attributes</u> <p style="text-align: center;"><u>Tom Hastings</u></p> |
| <p><u>Answer</u> <u>11/4/1998</u></p> | <p><u>The current text in Section 3.1.4.1 Request Operation Attributes, 5th paragraph of "attributes-natural-language says:</u></p> <p style="text-align: center;"><u>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.</u></p> <p><u>The clarification is to add the following sentence to the end of the paragraph:</u></p> <p style="text-align: center;"><u>The client MAY use the Natural Language Override mechanism even when the value is in the same natural language.</u></p> <p><u>The 7th paragraph says:</u></p> <p style="text-align: center;"><u>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.</u></p> |

IPP Issues List - Model only

| | |
|--|---|
| | <p><u>The clarification is to add the following sentence:</u></p> <p><u>The IPP object MAY use the Natural Language Override mechanism even when the value is in the same natural language.</u></p> <p><u>The last paragraph of 3.1.4.2 contains the sentence:</u></p> <p><u>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.</u></p> <p><u>The clarification is to add the same following sentence:</u></p> <p><u>The IPP object MAY use the Natural Language Override mechanism even when the value is in the same natural language.</u></p> |
|--|---|

- 1. 108
- 1. 109

| | |
|---------------------------------|---|
| <p><u>Question</u></p> | <p><u>1.47 NLO 3 of 4: Vote to simplify Get-Jobs</u></p> <p><u>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.</u></p> <p><u>Bob Herriot and Carl Kugler</u></p> |
| <p><u>Discussion</u></p> | <p>*****</p> <ul style="list-style-type: none"> * <u>The proposal to vote on is to delete the indicated paragraph</u> * <u>below from Section 3.2.6.2 Get-Jobs Response that requires the IPP</u> * <u>object to return the job's "attribute-natural-language" as the first job</u> * <u>attribute if it is different from the value being returned as the Get-Jobs</u> * <u>response "attribute-natural-language" operation attribute.</u> * <u>_____</u> * <u>Please indicate your acceptance or rejection of this proposal</u> * <u>on the mailing list by Monday, Nov 2.</u> <p>*****</p> <p><u>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?</u></p> <p><u>Background:</u></p> |

IPP Issues List - Model only

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.

IPP Issues List - Model only

| | |
|--|--|
| | <p>-----</p> <p><u>Currently, Section 3.2.6.2 Get-Jobs Response contains the following paragraph:</u></p> <p><u>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.</u></p> |
| <p><u>Answer</u> <u>11/04/1998</u></p> | <p><u>Delete the following paragraph from Section 3.2.6.2 Get-Jobs Response:</u></p> <p><u>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.</u></p> |

- 1. 110
- 1. 111

| | |
|---------------------------------|---|
| <p><u>Question</u></p> | <p><u>1.48 NLO 4 of 4: Vote to always use the Natural Language Override mechanism</u></p> <p><u>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.</u></p> <p><u>Carl Kugler</u></p> |
| <p><u>Discussion</u></p> | <p>*****</p> <p><u>* The proposal to vote on is to require all attributes to always</u></p> |

IPP Issues List - Model only

* 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

IPP Issues List - Model only

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.

IPP Issues List - Model only

| | |
|--|---|
| | <p><u>I expect that condition a) fails because some implementations do support it. But it may also be the case that some implementations don't fully implement this feature.</u></p> <p><u>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 responses) 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 nameWithLanguage using the override rules.</u></p> <p><u>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.</u></p> |
| <p><u>Answer</u> <u>11/04/1998</u></p> | <p><u>No change to [IPP-MOD]. However, to further clarify that the client MUST accept 'textWithLanguage' and 'nameWithLanguage' for any 'text' and 'name' attributes it supports as specified in section 4.2.1 and 4.2.2, add the following sentence to Section 5.1, Client Conformance Requirements:</u></p> <p style="padding-left: 40px;">A client MUST be able to accept any of the attribute syntaxes defined in Section 4.1, including their full range, that may be returned to it in a response from a Printer object. <u>For each attribute that the client supports whose attribute syntax is 'text' or 'name', the client MUST accept and process both the WithoutLanguage and WithLanguage forms.</u> For presentation purposes, truncation of long attribute values is not recommended. A recommended approach would be for the client implementation to allow the user to scroll through long attribute values.</p> |

- 1. 112
- 1. 113

| | |
|---------------------------------|--|
| <p><u>Question</u></p> | <p><u>1.50 What are the errors for each operation?</u></p> <p><u>It isn't clear what condition(s) cause which error codes to be returned for each operation</u></p> <p style="text-align: center;">Bob Herriot</p> |
| <p><u>Discussion</u></p> | <p><u>Need to add one line description of each of the error codes and the reason they are used as part of each operation description.</u></p> <p><u>ACTION ITEM (Tom Hastings and Bob Herriot): work out a proposal for each status code for each operation.</u></p> |
| <p><u>Answer</u></p> | <p><u>See separate complete proposal entitled: "MOD - Issue 1.50 - Status code</u></p> |

IPP Issues List - Model only

| | |
|------------|---|
| <u>10/</u> | descriptions for each operation" posted at: ftp://ftp.pwg.org/pub/pwg/ipp/proposed-clarifications/ipp-status-code-responses.doc .pdf |
|------------|---|

1. 114

1. 115

| | |
|-----------------------------|---|
| <u>Question</u> | <p><u>1.51 Can Get-Jobs redundantly contain job-level NLO?</u></p> <p><u>If we decide to keep the job-level NLO in Get-Jobs as in the June draft (see NLO 3 of 4 - Issue 1.47), we need to decide whether it MAY be used redundantly in a Get-Jobs response (just like we had to decide whether to allow redundant attribute-level NLO (see Issue 1.46):</u></p> <p><u>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.</u></p> <p><u>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.</u></p> <p><u>Comments?</u></p> <p><u>Tom Hastings</u></p> |
| <u>Discussion</u> | |
| <u>Answer</u> <u>10/</u> | <u>No, since Issue 1.47 removes the job-level natural language override.</u> |

1. 116

1. 117

| | |
|-----------------|---|
| <u>Question</u> | <p><u>1.52 Can Get-Jobs attribute-natural-language occur twice?</u></p> <p><u>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?</u></p> <p><u>Possibilities for "attributes-natural-language" Job attribute in the Get-Jobs response:</u></p> <ol style="list-style-type: none"> <u>1. MUST occur only once and be first</u> <u>2. SHOULD occur only once and be first</u> <u>3. MAY occur twice, once first and the other with the same value any where.</u> |
|-----------------|---|

IPP Issues List - Model only

| | |
|--|--|
| | <u>Tom Hastings</u> |
| <u>Discussion</u> | |
| <u>Answer</u> <u>10/</u> | No change to [IPP-MOD] since Issue 1.47 removed the job-level NLO from Get-Jobs Response. |
| 1. 118 | |
| 1. 119 | |
| <u>Question</u> | <p><u>1.53 Should we make document-format-supported REQUIRED for directories?</u></p> <p>Section 17 lists the REQUIRED and OPTIONAL attributes for directories entries. We have made REQUIRED, and attribute that is also required for an IPP Printer and we have made OPTIONAL any that are OPTIONAL for an IPP Printer. The single exception is that "document-format-supported" is still OPTIONAL. Agreed resolution to issue 1.4 did clarify that "document-format-supported" is REQUIRED for IPP Printers.</p> <p style="text-align: center;"><u>Tom Hastings</u></p> |
| <u>Discussion</u> | This is important for the SLP Printer template discussion and review, so that it is mandatory for directory entries. Also which document formats that a printer supports is pretty important to a user in order to select a printer. |
| <u>Answer</u> <u>11/04/1998</u> | Change the line in the table in Section 17, Generic Directory Schema that contains "document-format-supported" from OPTIONAL to REQUIRED. |
| 1. 120 | |
| 1. 121 | |
| <u>Question</u> | <p><u>1.54 Can't put one staple through multiple documents that start on new sheets</u></p> <p>The three values for "multiple-document-handling" control whether the documents are treated as one or separate. When separate, each document is forced onto a new sheet. But when the documents are treated as one, each document is not forced onto a new sheet. This causes a problem if you are stapling multiple two-sided documents with a single staple through the entire job. We need another value for single-document that does force the component documents onto new sheets.</p> <p style="text-align: center;"><u>Tom Hastings</u></p> |
| <u>Discussion</u> | <p>The current spec for 'single-document' is:</p> <p style="text-align: center;"><u>'single-document': If a Job object has multiple documents, say, the document data is called a and b, then the result of processing all the document data (a and then b) MUST be treated as a single sequence of media sheets for finishing operations; that is, finishing would be performed on the concatenation of the sequences a(*),b(*). The Printer object MUST NOT force the data in each document instance to be formatted onto a new print-stream page, nor to start a new impression on a new media sheet. If more</u></p> |

IPP Issues List - Model only

| | |
|---|---|
| | <p><u>than one copy is made, the ordering of the sets of media sheets resulting from processing the document data MUST be a(*), b(*), a(*), b(*), ..., and the Printer object MUST force each copy (a(*),b(*)) to start on a new media sheet.</u></p> <p><u>Add a new value, called, say 'single-document-forced-new-sheet' to go with the current three values: 'single-document', 'separate-documents-uncollated-copies', and 'separate-documents-collated-copies'</u></p> |
| <p><u>Answer</u> <u>11/8/1998</u></p> | <p><u>Add the following new keyword value to section 4.2.4 multiple-document-handling:</u></p> <p><u>'single-document-forced-new-sheet': Same as 'single-document', except that the Printer object MUST force the document data in each document instance to place the first impression on a new media sheet. This allows multiple documents to be stapled together with a single staple where each document starts on a new sheet.</u></p> <p><u>Also add a reference to this new value to the paragraph in section 4.2.4 which would read:</u></p> <p>The 'single-document' value is the same as 'separate-documents-collated-copies' with respect to ordering of print-stream pages, but not media sheet generation, since 'single-document' will put the first page of the next document on the back side of a sheet if an odd number of pages have been produced so far for the job, while 'separate-documents-collated-copies' always forces the next document or document copy on to a new sheet. In addition, if the "finishings" attribute specifies 'staple', then with 'single-document', documents a and b are stapled together as a single document <u>with no regard to new sheets, with 'single-document', documents a and b are stapled together as a single document, but document b starts on a new sheet</u>, but with 'separate-documents-uncollated-copies' and 'separate-documents-collated-copies', documents a and b are stapled separately.</p> <p><u>Also add a reference to this new value in the paragraph in Section 16.3, so that it would now read:</u></p> <p>3. The input to this step is a sequence of print-stream pages. This step is controlled by the "number-up" attribute. If the value of "number-up" is N, then during the processing of the print-stream pages, each N print-stream pages are positioned, as specified in section 4.2.9, to create a single impression. If a given document does not have N more print-stream pages, then the completion of the impression is controlled by the "multiple-document-handling" attribute as described in section 4.2.4; when the value of this attribute is 'single-document' <u>or 'single-document-new-sheet'</u>, the</p> |

IPP Issues List - Model only

| | |
|--|---|
| | print-stream pages of document data from subsequent documents is used to complete the impression. |
|--|---|

- 1. 122
- 1. 123