This document contains the AGREED resolutions to issues related to the IPP/1.0 Model and Semantics, dated June 30, 1998. A few resolutions also affect the IPP/1.0 Transport and Encoding, dated June 30, 1998 (referred to as PRO).

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

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

All references are to the June 30, 1998 drafts.

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

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

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

AGREED Resolutions to the IPP Issues List - Model only

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 a new issue is raised, it is copied to a new document called:
56
57 **PENDING IPP Issues List - Model only**
58
59 which is available at:
60
61 ftp://ftp.pwg.org/pub/pwg/ipp/proposed-clarifications/ipp-issues-list-mod-1.n.doc
62
63 where n is the version number.
64

AGREED Resolutions to the IPP Issues List - Model only

TABLE OF CONTENTS

AGREED Resolutions to the IPP Issues List - Model only

119

120

121

1   Change History for Model and Encoding/Transfer documents

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

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

128   The following substantive changes and major clarifications have been made to this
129   document from the June 30, 1998 version based on the interoperability testing that took
130   place September 23-25 1998. These changes are the ones that might affect
131   implementations. Clarifications that are unlikely to affect implementations are not listed.
132   The issue numbers refer to the IPP Issues List.
133

Section	Description
3.1.2 16.3.3	Clarify that the IPP object SHOULD NOT validate the range of the request-id being 1 to 2**31-1, but accepts and returns any value. Clients MUST still keep in the range though. (Issue 1.36)
3.3.3	Clarified that Cancel-Job MUST be rejected if the job is in 'completed', 'canceled', or 'aborted' job states. (Issue 1.12)
4.1.1.3 4.1.2.3	Added sections about comparing textWithLanguage and textWithoutLanguage indicating that the explicit language MUST match the implicit language. Same for comparing nameWithLanguage and nameWithoutLanguage. A keyword value never matches either type of value, even if the language is 'en-us'. (Issue 1.33 and 1.34)
4.1.5	Clarified regarding the case-insensitivity of URLs to refer only to the RFCs that define them. (Issue 1.10)
4.4.18 and 4.4.19	Clarified that the "document-format-default" and "document-format-supported" Printer Description attributes are REQUIRED. (Issue 1.4)
4.4.21	Changed "queued-job-count" from OPTIONAL to RECOMMENDED. (Issue 1.14)
8.5	Added a new section RECOMMENDING listing non-IPP jobs using Get-Jobs whether or not assigning them a job-id and job-uri. Also RECOMMENDED generating, if possible, job-id and job-uri and supporting other IPP operations on foreign jobs as an implementer option. (Issue 1.32)
14.1.2.2 and Get-xxx	Clarified that an IPP object MUST return 'successful-ok-ignored-or-substituted-attributes' (0x1), rather than 'successful-ok' (0x0), when a client supplied unsupported attributes as values of the 'requested-attributes'

AGREED Resolutions to the IPP Issues List - Model only

operations	operation attribute. (Issue 1.24)
14.1.5.9	Added a new error code 'server-error-job-canceled' (0x0508) to be returned if a job is canceled by another client or aborted by the IPP object while the first client is still sending the document data. (Issue 1.29)

134

135 **2   Model & Semantics**

136

Question	**1.1 xxx-supported and PDL-only supported features** For each job template attribute there is the associated default and supported values. I have a question about the xxx-supported values. Imagine a printer that say supports binding which may be controlled by various PDL commands, but does not support controlling binding via the IPP finishings job template attribute. Should the printer response to finishings-supported include binding or not? I assume that it should not include binding as this would give the idea to the client that binding can be controlled with the finishings attribute. Thus, xxx-supported is not intended to indicate printer capabilities, but rather support for the IPP attributes. Is this correct? **Stuart Rowley**
Discussion	Should we add a new series of Printer Description attributes of the form: "xxx-supported-in-document-data?
Answer 8/19/1998	Correct. The values of "xxx-supported" attributes MUST not include values that are only supported in the PDL data stream. The values do include values that are supported in both the protocol and the PDL data stream, as well as values that are supported only in the protocol. The values MAY also include actions carried about manually by an operator on a completed job, such as stapling or bursting. Yes, further attributes may be added in the future. Capability might be provided by post processing outside the printer. No change to MOD. Add question and answer to FAQ

137

138

Question	**1.2 Identifying document-format dependent JT attributes**
	It looks like the problem discussed in "document-format-supported" [MOD needs clarification], http://www.findmail.com/list/ipp/showthread.html?num=3864 was addressed in the new MOD, ftp://ftp.ietf.org/internet-drafts/draft-ietf-ipp-model-10.txt June 30, 1998. The new words say: "If the Printer object does distinguish between different sets of supported values for each different document format specified by the client, this specialization applies only to the following Printer object attributes: - Printer attributes that are Job Template attributes ("xxx-default" "xxx-supported", and "xxx-ready" in the Table in Section 4.2), - "pdl-override-supported", - "compression-supported", - "job-k-octets-supported", - "job-impressions-supported, - "job-media-sheets-supported" - "printer-driver-installer", - "color-supported", and - "reference-uri-schemes-supported" "The values of all other Printer object attributes (including "document-format-supported") remain invariant with respect to the client supplied document format (except for new Printer description attribute as registered according to section 6.2). While this new wording gets around the problem, I think it presents a poor model. It blatantly violates Second Normal Form, in that some Printer attributes depend on the (Printer identifier, document-format) tuple, while others depend only on the Printer identifier. The model says that all these attributes, including those that vary with document-format (e.g., number-up), are attributes of the Printer class of objects. But the implication is that each real-wold printer maps to a whole set of Printer object instances, selected by document-format. Attributes (e.g. printer-name) which don't vary with document-format are redundantly stored in each instance. Updates to attributes that don't vary with document-format (e.g. printer-state) require visiting all the instances. A better model would split the existing Printer into two classes of objects: 1) a new, reduced Printer, and 2) something else that could be called "Interpreter". Then the attributes can be normalized between these two new classes. Attributes that don't vary with document-format are assigned to the Printer. Each real-world printer maps to one instance of Printer.

	Attributes that do vary with document-format are assigned to Interpreter. Each Printer instance contains one or more Interpreter instances, selected by document-format. I know that IPP doesn't claim to be truly object-oriented. But I think considerations like this are important for a few reasons: - IPP looks object-oriented, with terms like Object, and attribute, and Operation bandied about. It will lead to confusion if the IPP model is anti-object-oriented. Let's not call Printer an object if it represents something other than what an object is commonly understood to be. - Many implementors are likely to use OO methods. (How about a poll of current implementors?) It would sure be nice if the IPP model could map easily to an OO design and implementation. - Although an implementor's design could split up these classes internally and still meet the existing spec, there is some value in having the implementation, the design, and the model trace cleanly back to the real world. **Carl Kugler**
Discussion	
Answer 8/19/1998	In IPP v1.0, other objects are "hidden." We might consider this for a future version. No change to MOD.

139

140

Question	**1.3 Validating type 3 keyword | name attributes**
	In the Job Template Attributes there are attributes that can be a type3 keyword or a name (job-hold-until, job-sheets, and media). As I read the spec, these attributes are usually type3 keywords but can optionally be changed at the printer to a name type. Is this correct or did I miss something in the spec? My question is how does an IPP client know which type to send? If the wrong type is sent, what should the expected reply be? **Rajesh Chawla**
Discussion	My understanding, based on my reading of the spec and questions I've asked here in the past: Those attributes can be typed, and tagged as any of the following: 0x36 nameWithLanguage 0x42 nameWithoutLanguage 0x44 keyword In general, an IPP Object may send any one of the three types, and must accept any one of the three. However, for any 'name' attribute in the request that is in a different natural language than the value supplied in the "attributes-natural-language", the sender must use the nameWithLanguage form. Type 3 keywords have standard, registered values. If the wrong type is sent in a request, according to MOD section 16.4.3, the response should be 'client-error-request-value-too-long'. Quote: "IF NOT any single 'keyword' or 'name' value less than or equal to 255 octets, REJECT/RETURN 'client-error-request-value-too-long'.") **Carl Kugler**
Answer 8/19/1998	Section 16.4.3 needs to be clarified. The sentence should only be talking about the case of a value that is too long, but is one of the expected attribute syntaxes (keyword, nameWithLanguage, or nameWithoutLanguage). After examining the question, the group does not agree with Carl Kugler's last paragraph as an attempted answer. Bob Herriot will draft a proposed response for this issue, and submit it to for consideration by the group.

141

142

Question	**1.4 Are "document-format-default" and "document-format-supported" REQUIRED Printer Description attributes?** The table in Section 4 says that "document-format-default" and "document-format-supported" are REQUIRED, but the descriptions of those attributes in sections 4.4.18 and 4.4.19 do not say REQUIRED. I believe that 4.4.18 and 4.4.19 should be fixed by adding REQUIRED to agree with the table, like the other attributes that are REQUIRED. These two attributes are so fundamental to the description of a Printer object that the fix should NOT be to remove REQUIRED from the table. **Tom Hastings**
Discussion	
Answer 8/19/1998	Update sections 4.4.18 and 4.14.19 to indicate that the "document-format-default" and "document-format-supported" Printer Descriptions attributes are REQUIRED to agree with the table in Section 4. The group agreed to Tom Hastings's suggestion proposed in the Question.

143

144
145

Question	**1.5 What charset conversion is required for Get-xxx requests?**
	How should the server handle the situation where the "attributes-charset" of the response itself is "us-ascii", but one or more attributes in that response is in the "utf-8" format?
	Consider a case where a client sends a Print-Job request with "utf-8" as the value of "attributes-charset" and with the "job-name" attribute supplied. Later another client sends a Get-Job-Attribute or Get-Jobs request. This second request contains the "attributes-charset" with value "us-ascii" and "requested-attributes" attribute with exactly one value "job-name".
	According to the IPP-Mod document (section 3.1.4.2), the value of the "attributes-charset" for the response of the second request must be "us-ascii" since that is the charset specified in the request. The "job-name" value, however, is in "utf-8" format. Should the request be rejected even though both "utf-8" and "us-ascii" charsets are supported by the server? or should the "job-name" value be converted to "us-ascii" and return "successful-ok-conflicting-attributes" (0x0002) as the status code?
	Van Dang
Discussion	My understanding: in this situation the Printer is required to convert the job-name value from utf-8 charset to us-ascii. If it can't, it shouldn't advertise us-ascii as a attributes-charset-supported.
	One implementation strategy is to convert all incoming text and name values to a Unicode internal representation. This is 16-bit and virtually universal. Then convert to the specified operation attributes-charset on output.
	Carl Kugler
Answer 8/19/1998	An IPP object that supports both utf-8 (REQUIRED) and us-ascii, the second paragraph of section 3.1.4.2 applies so that the IPP object MUST accept the request, perform code set conversion between these two charsets with "the highest fidelity possible" and return 'successful-ok', rather than a warning 'successful-ok-conflicting-attributes, or an error.
	Also we observed that is would be smarter for a client to ask for 'utf-8', rather than 'us-ascii' and throw away characters that it doesn't understand. The current document addresses this Question already. The printer will do the best it can to convert between each of the character sets that it supports--even if that means providing a string of question marks because none of the characters are representable in US ASCII. [Some people noted

	that the problem is not likely to occur in most practical situations.] No change to MOD. Add the above discussion to the IIG.

146

147

Question	**1.6 Should we add "pages-per-minute" Printer Description attribute to IPP-MOD, Directory, and SLP?** I recently noticed there is no pages-per-minute attribute in IPP. I noticed this first when reviewing the draft printer scheme for SLP (draft-ietf-srvloc-printer-scheme-02.txt). The printer scheme seems to inherit it's attribute definitions from IPP. I think ppm is one of the most fundamental attributes in terms of printer selection. I'm sure this must have been discussed at some point during IPP development, probably at a time when I wasn't paying much attention to the mail list. I do remember a discussion about a cost attribute that was eliminated because it was deemed too qualitative. But ppm is quantitative and universal in advertising printers. So, can someone explain why it is not an IPP printer attribute? And, for those familiar with the SLP printer scheme effort, why is it not part of the SLP printer scheme? **Angelo Caruso**
Discussion	You could make this a directory attribute, but I don't think its absolutely necessary to support it in IPP. Besides, its in the printer MIB ;) **Randy Turner** I think that we discussed this at some stage and found that it was not clear that we could come up with a single value. For example, depending on the type of printer, the speed is often dependent on whether you run in "draft" mode vs. "quality" mode, and whether you run B/W or color. So we would have ended up with some kind of conditions and several values to cover all cases. **Carl-Uno Manros**
Answer 8/19/1998	Such an attribute should be registered. Perhaps call it "pages-per-minute". Also clarify that the number used is not exact, but is what is used in the promotional literature to describe the device. Even devices that are not page printers are described in pages per minute in such literature. That attribute should also be added to the list of directory attributes in section 17 of IPP-MOD, "APPENDIX E: Generic Directory Schema. That attribute should also be added to the SLP Schema too. [The group feels that this Question does not belong in the Issues List. The Question will be removed.] Because the definition of "pages-per-minute" is so varied--based on quality, color, page content, etc.--a single-valued attribute will not be added. Instead, people are encouraged to generate a proposal for addressing this issue as a future registration proposal.

148
149

AGREED Resolutions to the IPP Issues List - Model only

Question	**1.7 Should Validate-Job remain a REQUIRED operation?** Is it really necessary to keep the "Validate-Job" operation as a MUST to implement? The "Get-Printer-Attributes" operation seems to provide all the functionality that is needed. **Carl-Uno Manros**
Discussion	Validate job is intended to pertain to more than just printer attributes. It should also cover print job attributes (like n-up, for example). Isn't Validate-Job akin to checking the "job ticket" whereas Get-Printer-Attributes is akin to determining the device configuration? **Harry Lewis**
Answer 8/19/1998	Keep Validate-Job as a REQUIRED operation. The September '98 bake off confirmed that every implementation had implemented it. The intention is that the Print-Job code can be re-used for Validate-Job, with the only difference being that no data is sent and no job attributes are returned. Yes, it is really necessary to keep the "Validate-Job" operation as a MUST to implement. No change to MOD.

150
151

Question	**1.8 Is it ok for an IPP Printer to restrict Create-Job, Send-Document, and Send-URI to one document?** Can you implement the operations "Create-Job", "Send-Document" and "Send-URI", without the need to support multiple documents? This could be useful for environments where you have long jobs, but do not need support for multiple documents. **Carl-Uno Manros**
Discussion	The model document supports the notion of a Create-Job operation followed by only one Send-Document operation as semantically equivalent to a Print-Job operation. It cautions regarding performance, however. If you are asking is it ok to support Creat-Job, Send-Doc with only one document - Yes. If you are asking is it ok to support Create-Job but LIMIT Send-Doc to only one document... I'd say that would be a non-no! **Harry Lewis**
Answer 8/19/1998	If you support Create-Job, Send-Document (and Send-URI), then you MUST support multiple documents. Thus a client can determine if an IPP Printer supports multiple documents by querying the Printer's "operations-supported" attribute. No change to MOD.

152

153

Question	**1.9 Requirements for "printer-up-time" versus "time-at-creation", "time-at-processing", and "time-at-completed?** What was the rationale for making the "printer-up-time" attribute a REQUIIRED attribute, considering that the other 3 attributes "time-at-creation", "time-at-processing", and "time-at-completed", with which it is associated, are all OPTIONAL? **Carl-Uno Manros**
Discussion	Don't know for sure but I suspect this attempts to make a running "time marker" available for monitoring, tracking accounting etc... without mandating all the possible time recording points on each IPP device. This is somewhat analogous to the sysUpTime concept in MIB-II. **Harry Lewis** Should we make at least one of the Job Description attribute REQUIRED? How about "time-at-creation"? **Tom Hastings**
Answer 8/19/1998	The group agreed that Harry's response (contained in the document) will be re-worded and used as the answer. No change to MOD.

154

155

Question	**1.10 Case sensitivness in URLs**
	Which parts of a URL are case-insensitive and which parts are case-sensitive?
	IPP Bake Off

Discussion	9/30/98: Characters from "http://" to the first "/" are <u>case insensitive</u>, any characters after that are <u>case sensitive</u>.
	10/7/98: We agreed that IPP MOD will not specify anything further about the case sensitivity, but simply refer to the relevant standards. A new URI draft standard, RFC 2396 will be referred to.
	
	There are some additional rules for URI comparison in [RFC2068] (beyond those in [RFC2396]) that are relevant here, since we are using HTTP/1.1 as a transport layer:
	
	3.2.3 URI Comparison
	  When comparing two URIs to decide if they match or not, a client
	  SHOULD use a case-sensitive octet-by-octet comparison of the entire
	  URIs, with these exceptions:
	   o A port that is empty or not given is equivalent to the default
	     port for that URI;
	   o Comparisons of host names MUST be case-insensitive;
	   o Comparisons of scheme names MUST be case-insensitive;
	   o An empty abs_path is equivalent to an abs_path of "/".
	  Characters other than those in the "reserved" and "unsafe" sets (see
	  section 3.2) are equivalent to their ""%" HEX HEX" encodings.
	  For example, the following three URIs are equivalent:
	    http://abc.com:80/~smith/home.html
	    http://ABC.com/%7Esmith/home.html
	    http://ABC.com:/%7esmith/home.html
	
	Might want to cite that spec, too.
	
	 -Carl
	
	For the IIG:
	Harry has proposed some text for the IPP Implementor's Guide (IIG) which is non-standards track companiion document to the MOD and PRO documents.
	
	Carl has given the reason that IPP doesn't want to be any more restrictive than HTTP/1.1 and the URI specifications: so that off-the shelf components may be used.
	
	Paul has suggested that two printers SHOULD not differ only in case.

AGREED Resolutions to the IPP Issues List - Model only

Answer 10/12/1998	Agreements reached by 10/12 with no further discussion:

	MOD Section 4.1.5 'uri' (attribute syntax):
	1. Don't attempt to make any further restrictions from other standards about the case-sensitivity or case-insensitivity of URLs or URIs.
	2. Don't even attempt to paraphrase other standards about the case-sensitivity or case-insensitivity of URLs or URIs. Just refer to them for case-sensitiveness and case-insensitivity.
	3. Now that there is finally a URI Draft IETF standard, replace all references to RFC 1630, RFC 1738 and RFC 1808 with RFC 2396.
	4. Keep references to HTTP/1.1 [RFC 2068] too, since it contains some specifications for comparing URLs.
	SPECIFIC PROPOSED CLARIFICATIONS TO JUNE 1998 IPP MODEL DOCUMENT:
	Change Section 4.1.5 'uri' from:
	4.1.5 'uri'
	The 'uri' attribute syntax is any valid Uniform Resource Identifier or URI [RFC1630]. Most often, URIs are simply Uniform Resource Locators or URLs [RFC1738] [RFC1808]. The maximum length of URIs used within IPP is 1023 octets. Although most other IPP syntax types allow for only lower-cased values, this syntax type allows for mixed-case values. The URI and URL standards allow for mixed-case values that are case-sensitive.
	to:
	4.1.5 'uri'
	The 'uri' attribute syntax is any valid Uniform Resource Identifier or URI [RFC2396]. Most often, URIs are simply Uniform Resource Locators or URLs. The maximum length of URIs used as values of IPP attributes is 1023 octets. Although most other IPP attribute syntax types allow for only lower-cased values, this syntax conforms to the case-sensitive and case-insensitive rules specified in [RFC2396].
	Throughout MOD and PRO, replace references to RFC 1630, RFC 1738 and RFC 1808 with just RFC 2396.
	In Section 9 References, make the following changes:
	1. remove RFC 1630, since RFC 2396 refers to it as "historical".
	2. also remove RFC 1738 and 1808, since RFC 2396 says the it replaces RFC 1738 and 1808 (in the Abstract, though the status says "updates" 1738 and 1808, instead of "supersedes").
	3. finally, add:
	[RFC2396]

<table>
<tr><td rowspan="7"></td><td>Berners-Lee, T., Fielding, R., Masinter, L.,
"Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.

Proposed text for IIG:

IPP client and server implementations must be aware of the diverse uppercase/lowercase nature of URIs. RFC 2396 defines URL schemes and Host names as case insensitive but reminds us that the rest of the URL may well demonstrate case sensitivity. When creating URL's for fields where the choice is completely arbitrary, it is probably best to select lower case. However, this cannot be guaranteed and implementations MUST NOT rely on any fields being case-sensitive or case-insensitive in the URL beyond the URL scheme and host name fields.

The reason that the IPP standard does not make any restrictions on URIs, is so that implementations of IPP may use off-the-shelf components that conform to the standards that define URIs, such as RFC 2396 and the HTTP/1.1 specifications [RFC2068]. See these specifications for rules of matching, comparison, and case-sensitivity.

It is also recommended that that System Administrators and implementations avoid creating URLs for different printers that differ only in their case. For example, don't have Printer1 and printer1 as two different IPP Printers.</td></tr>
</table>

156
157

Question	**1.11 No response to a Cancel-Job operation** Some implementations do not send back an HTTP response to the Cancel-Job operation. **IPP Bake Off**
Discussion	No where in the spec does it allow or mention not returning a Cancel-Job response.
Answer 9/30/1998	Not returning a response to a Cancel-Job operation is a bug in the implementation. No change will be made to the MOD, PRO, or IIG documents.

158

159

Question	**1.12 Cancel-Job response to a 'completed' job** Implementations react differently to "Cancel-Job". Some return a client-error-not-possible error as IPP-MOD says. Some return success-ok and leave the job in the 'completed' state. Some return success-ok and delete the job immediately, removing it from the job history. What is correct response when job is already completed? Should Cancel-Job result in deletion of job history? **IPP Bake Off**
Discussion	If we change the spec so that a Cancel-Job is accepted, instead of rejected when the job is in the 'completed', 'canceled', or 'aborted' state, a job that a user thinks is still processing, may just have completed before issuing the Cancel-Job. Thus the user will be mis-led into thinking that the job was prevented from printing if Cancel-Job returned 'successful-ok' when the job was already 'completed'. The current spec does not have a way for the user to indicate that a job is to be retained after completing. Thus Cancel-Job is not needed to get rid of such a retained job. When we figure out how to allow a user to submit (or modify) a job to be retained (for a period of time as in DPA), we will also figure out how to allow the client to change that period of time to get rid of the job. A retained job can be indicated using the job's "job-state-reasons" attribute with the 'job-restartable' value.

Answer 9/30/1998	Keep Cancel-Job spec as MOD section 3.3.3.2 says:
	    If the job is already in the 'completed', 'aborted', or 'canceled' state, or the 'process-to-stop-point' value is set in the Job's "job-state-reasons" attribute, the Printer object MUST reject the request and return the 'client-error-not-possible' error status code.
	
	The first line of MOD section 3.3.3 will be changed from:
	    This REQUIRED operation allows a client to cancel a Print Job any time after a create job operation.
	
	to:
	    This REQUIRED operation allows a client to cancel a Print Job from the time the job is created up to the time it is completed, canceled, or aborted.
	
	so that it does not appear to contradict section 3.3.3.2.

160
161

Question	**1.13 Job-attribute response to Hold-Job, Release-Job, Restart-Job** The Set 1 Spec specifies that the three job operations (Hold-Job, Release-Job, and Restart-Job) MUST return the "job-state", and, if supported, the "job-state-reasons" attributes. However, implementations did not return any job attributes in the response. Should we change the spec to not require any job attributes to be returned? Should we allow any to be returned? Should a Restart-Job implementation be required to return the same job attributes that Print-Job returns ("job-uri", "job-id", neither of which can change, "job-state" which could be 'pending', 'pending-held', or 'processing') Should Restart-Job implementation be allowed to return the same optional job attributes that Print-Job returns ("job-state-reasons", "job-state-message", and "number-of-intervening-jobs")? **IPP Bake Off**
Discussion	None of the current implementations of Set 1 return neither job attributes for the three Job operations nor printer attributes for the three Printer operations. Some implementations may have difficulty returning the new job state or printer state after the operation, since the job or printer state may be changing. A client can query the Job object using Get-Job-Attributes or the Printer object using the Get-Printer-Attributes after getting the response, if it wants to display the new job or printer state.
Answer 9/30/1998	Remove Group 3 from the spec for the responses for all six operations so that none of them return job or printer attributes. The IIG cannot mention these Set 1 operations, since the IIG is going to go become an Internet-Draft along with MOD and PRO, but Set 1 will become and Internet-Draft after IPP 1.0 is approved by the IESG.

162

163

Question	**1.14 Should "queued-job-count" be REQUIRED?** Should we make the printer description attribute "queued-job-count" a required attribute? **IPP Bake Off**
Discussion	Then a client could depend on the "queued-job-count" as being a fast way to determine whether the printer has a long queue or not? However, since there are some implementations that didn't implement "queued-job-count", it seemed problematic to REQUIRE it.
Answer 9/30/1998	Recommend that "queued-job-count" be implemented. Change MOD Section 4.4 Table to add SHOULD to last column entry for "queued-job-count". Add the word "RECOMMENDED" as the second word in the first sentence of MOD section 4.4.21. In the IIG, indicate that the reason that "queued-job-count" is RECOMMENDED, is that some clients look at that attribute alone when summarizing the status of a list of printers, instead of doing a Get-Jobs to determine the number of jobs in the queue.

164
165

Question	**1.15 Should "queued-job-count" not include 'pending-held' jobs?** The current Model document specifies that "queued-job-count" includes jobs that are in the 'pending-held' state, as well as 'pending', 'processing', and 'processing-stopped'. But these jobs are not in competition (yet) for the printer, until a client performs a Release-Job operation on them. **IPP Bake Off**
Discussion	The PWG Job Monitoring MIB does not include 'pending-held' jobs in its jmGeneralNumberOfActiveJobs object, only 'pending', 'processing', and 'processing-stopped'. On the other hand, there is good reason to have both numbers: the total number of jobs in the queue and the number that are in the 'pending-held' state, for implementations that support that state. So a future registration could be to add a "held-job-count" (or an "active-job-count") Printer Description attribute.
Answer 9/30/1998	No change to MOD. The IIG will indicate that the "queued-job-count" is not a good measure of how busy the printer is when there are held jobs. Also indicate that a future registration could be to add a "held-job-count" (or an "active-job-count") Printer Description attribute.

166
167

Question	**1.16 Empty Job Template attribute group in a Print-Job request**
	If a client does not have any job template attributes to send (or does not support ANY job template attributes), does it still have to send the empty group for job template attributes?
	IPP Bake Off
Discussion	Probably needs clarification in both MOD and PRO.
Answer 9/30/1998	An IPP object MUST accept both forms in a request and that a client MUST accept both forms in a response. PRO lines 24-267:
	The syntax allows an xxx-attributes-tag to be present when the xxx-attribute-sequence that follows is empty. The syntax is defined this way to allow for the response of Get-Jobs where no attributes are returned for some job-objects. Although it is RECOMMENDED that the sender not send an xxx-attributes-tag if there are no attributes (except in the Get-Jobs response just mentioned), the receiver MUST be able to decode such syntax.
	There doesn't seem to be any reason to specify in MOD whether or not empty groups can be omitted by a sender, since a different syntax might have different rules about empty groups. Therefore, no changes to either MOD or PRO.
	The IIG will indicate that the terms "sender" means client for a request and IPP object for a response. Also that an IPP object SHOULD be forgiving in accepting requests in order to work with the most clients. On the other hand, clients should be conforming in requests so that they will work with the most IPP objects.

168
169

Question	**1.17 Empty groups in responses**
	MAY an IPP object omit an empty group, such as a Job Attributes or Printer Attributes group entirely in a response for any operation if there are no attributes to return?
	IPP Bake Off
Discussion	In the Get-Printer-Attributes operation, if e.g. you do not have any job template attributes to send back, why do you have to send an empty printer group in the "requested attributes' test case 2.8 with TS1?
	Probably needs clarification in both MOD and PRO.
Answer 9/30/1998	No change to MOD or PRO, see Issue 1.17.

170

171

Question	**1.18 Returning Unsupported attributes in Get-Xxxx operations** Inconsistent wording in the Model & Semantics document about whether you must return unsupported attributes in Get-Printer-Attributes, Get-Job-Attributes, and Get-Jobs in the Unsupported Attributes group. **IPP Bake Off**
Discussion	One the one hand, the request contained an operation attribute with unsupported values, namely "requested-attributes" with a values of 'xxx' and 'yyy' that are unsupported, so the IPP object returns the unsupported values. On the other hand, the Group 3 text specifies that the IPP object "ignores" any unsupported attributes that are requested. Does it help a client to know which attributes it has requested that are not supported? Or is it sufficient for the client to discover which are unsupported because they were not returned in the Job or Printer attributes group? See also Issue 1.24 which concerns the status code returned.

Answer 9/30/1998	An IPP object MAY return requested attributes that are unsupported in Group 2 in Get-Printer-Attributes, Get-Jobs, and Get-Job-Attributes responses, but a client cannot depend on it.
	
	Add the following sentence:
	
	    The response NEED NOT contain the "requested-attributes" operation attribute with any supplied values (attribute keywords) that were requested by the client but are not supported by the IPP object.
	
	to MOD 3.2.5.2 Get-Printer-Attributes response, 3.2.6.2 Get-Jobs response, and 3.3.4.2 Get-Job-Attributes response:
	
	    Group 2: Unsupported Attributes
	
	        This is a set of Operation attributes supplied by the client (in the request) that are not supported by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16).
	
	Add a statement to the IIG that the client cannot depend on getting unsupported attributes returned in the Unsupported Attributes group of Get-Xxxx responses that the client requested, but are not supported by the IPP object. However, such unsupported requested attributes will not be returned in the Job Attributes or Printer Attributes group (since they are unsupported).

172
173

AGREED Resolutions to the IPP Issues List - Model only

Question	**1.19 What charset to return when an unsupported charset is requested?** What character set should a server use for the value when returning the value of an unknown or badly formed attribute? Should it be the IPP Printer's configured charset or UTF-8? <center>**IPP Bake Off**</center>
Discussion	While clients SHOULD support UTF-8, they NEED NOT. Only IPP objects are required to support UTF-8. Since there is a specific client-error-charset-not-supported status code, the client can determine the error even if it doesn't understand the charset that the IPP object is configured for. While we thought that MOD specified that any 'text' or 'name' attributes returned in the error response, such as "status-message" MUST use the "charset-configured" charset, I could not find it in MOD. Discussion for the IIG from 10/13/1998 by Hugo Parra and Carl Kugler: I've inserted some comments in Carl's text below. > -Hugo > > >>> "Carl Kugler" <kugler@us.ibm.com> 10/12 12:43 PM >>> > > 1.19 - When an unsupported char set is requested, what character set > should a server use when returning the unknown attribute? > > The server should use it default character set as currently stated > in the spec. > > There is actually a larger question here. There are other cases in which > a request can be REJECTED before the "attributes-charset" is known > and validated. > > I interpret MOD as saying that the request validation steps can occur in = > any order, and that the Printer is free to REJECT a request as soon as it > finds a reason to do so: > > [[[HParra]]] One exception to this may need to be explicitly noted, > namely, the reject response resulting from a job creation request > specifying unsupported attributes and ipp-attribute-fidelity TRUE. > It should be strongly recommended/mandated that the response include > the list of all unsupported attributes that caused the operation to fail, so > users don't have to try submitting a job multiple times to find out > exactly what all attributes caused the job to fail. > > >MOD>16.3 Suggested Operation Processing Steps for All Operations > In order to determine whether or not to accept or reject the request, the > IPP object SHOULD execute the following steps. The order of the steps

> may be rearranged and/or combined, including making one or multiple
> passes over the request.
>
> However, validating "attributes-charset" requires processing the whole
> request:
>
> >...it is an error for the "attributes-charset" or "attributes-natural-lang
> uage" attribute to be omitted in any operation request, or for an
> Operation attribute to be supplied in a Job Template group or a Job
> Template attribute to be supplied in an Operation Attribute group in a =
> create request. It is also an error to supply the "attributes-charset"
> attribute twice.
>
> So, in general, a request MAY be REJECTED before the request
"attributes-charset" has been read and/or validated. Examples:
>
> 1) Bad version number.
> 2) Unsupported Operation identifier
> 3) "attributes-charset" was omitted.
> 4) "attributes-charset" was supplied more than once.
>
> My simple implementation solution was to use the Printer's default
charset =
> whenever REJECTing a request. But this approach fails some of the
test scripts.
>
> I'd prefer a simple rule saying something like "a Printer MAY use its
> default charset in a rejection response". Or at least for some subset of
> responses like "client-error-bad-request" and "client-error-charset-not-
supported". Otherwise we have a mess of special cases:
>
> - The order of the steps may be rearranged and/or combined, including
> making one or multiple passes over the request, except that "attributes-
> charset" has to be processed before the request is rejected, so at least
> one complete pass is required.
>
> [[[HParra]]] Is a complete pass really required? Section 3.1.4 of the
> MOD states, "The 'attributes-charset' attribute MUST be the first
> attribute in the (Operation Attributes) group and the
> 'attributes-natural-language' attribute MUST be the second attribute in
> the group." Once an IPP printer has validated this information it should
> be free to reject a request at any point it deems appropriate.

But the operation identifier, for example, precedes the 'attributes-charset'.
So, for a Printer that processes the data stream as it arrives, in the case of
an unsupported operation, the Printer would have to remember that

problem and continue to process the request trying to find a valid 'attributes-charset' to use for the response.

And what if it can't validate the request 'attributes-charset'?

Section 16.3.5, Validate the values of the REQUIRED Operation attributes, says:
>attributes-charset (charset)
>IF NOT any single non-empty 'charset' value less than or equal to 63 octets, REJECT/RETURN 'client-error-request-value-too-long'.

What "attributes-charset" should be used for the 'client-error-request-value-too-long' response?

>If an IPP object receives a request with required attributes missing or repeated from a group, the IPP object REJECTS the request and RETURNS the 'client-error-bad-request' status code.

Suppose "attributes-charset" is repeated in the operation attributes group. What "attributes-charset" must be used in the 'client-error-bad-request' response? The first one? Either one?

> If a duplicate 'attribute-charset' is
> specified later on, the printer rejects the request when it runs into it,
> if it makes it that far before rejecting the request for any other reason.
> What am I missing?

It all depends on how strongly you want to validate 'attributes-charset' before returning the response. Lately, 'attributes-charset' seems to have a kind of elevated status as a special attribute in a special position with special rules applied to it. Recent interpretations seem to imply that 'attributes-charset' errors take precedence over other errors, since you can't form a valid response except for 'client-error-charset-not-supported' unless you can get a valid, supported 'attributes-charset' from the request.

>
> - The IPP object NEED NOT find all attribute errors before returning an
=
> error, but it must process the entire request to validate that "attributes-
> charset" is a single non-empty 'charset' value less than or equal to 63
> octets,=20
> and in the Printer object's "charset-supported" attribute, before
> returning an error.
> - The Printer object checks to see if the "operation-id" attribute
> supplied by the client is supported as indicated in the Printer object's
> "printer-operations-supported" attribute. If not, the Printer REJECTS

	> the request and returns the 'server-error-operation-not-supported' status > code in the response AFTER processing the rest of the operation > attributes group to read and validate the "attributes-charset". > - For rejection responses, the error status codes returned may differ > between implementations, but "attributes-charset" errors take > precedence. - etc. > > -Carl

Answer 9/30/1998	The IPP object returns any 'text' or 'name' attributes using the Printer's "charset-configured" charset and the 'client-error-charset-not-supported' error status code.
	
	Clarify MOD section 3.1.4.1 third paragraph by adding:
	    and any 'text' or 'name' attributes using the Printer's "charset-configured" charset.
	to the end of:
	    If the Printer object does not support the client supplied charset value, the Printer object MUST reject the request and return the 'client-error-charset-not-supported' status code.
	
	Clarify MOD section 14.1.4.14 'client-error-charset-not-supported' by replacing:
	    the Printer MUST reject the operation and return this status (see Section 3.1.4.1).
	with:
	    the Printer MUST reject the operation and return this status and any 'text' or 'name' attributes using the Printer's "charset-configured" charset (see Section 3.1.4.1).
	
	Add to the IIG: Since such an error is a client error, rather than a user error, the client should check the status code first so that it can avoid displaying any other returned 'text' and 'name' attributes that are in an unexpected charset. Also add to the IIG from the above discussion.

174

175

Question	**1.20 The 'resolution' attribute syntax is not two bytes** IPP-MOD says that resolution should be two bytes. This is wrong, see syntax. **IPP Bake Off**
Discussion	The MOD integer data type is 4 octets long, so don't use the term integer for anything that isn't 4 octets long.

Answer 9/30/1998	MOD section 4.1.15 'resolution' says:
	       It consists of 3 integers:
	 Since the third integer is only a byte according to PRO, change the above MOD sentence to:
	       It consists of 3 values:

176
177

Question	**1.21 Position of the target operation attributes in requests** Although IPP-MOD says that target (Job-URI, Print-URI plus Job-Id or Printer-URI) MUST be the 3rd operation attribute, several implementations do not have it in that place or not at all. Can we relax that requirement or should it be strictly enforced? **IPP Bake Off**
Discussion	The reason for redundantly having the target in the MIME body, os to that the IPP MIME data is transport independent. Appendix D MOD Section 16.3.4.3 describes the IPP object checking and rejecting the request if the target is not present or not in the proper order.
Answer 9/30/1998	Keep MOD requiring the client to supply the target operation attribute and in the correct position. However, the IPP object SHOULD NOT check for it being present and in the correct position, following the philosophy that clients should be conforming and servers should be forgiving. Move Section 16 (Appendix D) to the IIG. Keep the error check as something that a test suite for clients might include, but remove the error check for recommended IPP object behavior.

178

179

Question	**1.22 A Paused printer may never return a response to Print-job until Resumed** Test cases 2.6-2.7 and 2.9 in TS1 seems to expect a response before all the data has been sent. This results in a deadlock situation with some printers which are still waiting for all the data to first be delivered. **IPP Bake Off**
Discussion	A paused printer (or one that is stopped due to paper out or jam, may flow control the data of a Print-Job operation, so that the client is not able to send all the document data. Consequently, the Printer will not return a response. Thus the script will never step onto the next operation and the script hangs.
Answer 9/30/1998	No change to MOD or PRO. All printers will eventually flow control a Print-Job data when its buffers and spool space, if it spools, fills up. The Printer should not return an error, since either the printer will be resumed and/or the spool space will be freed up as jobs print. Fix the script to still test sending a Print-Job while the printer is paused, but figure out a way for the script not to hang, if the Printer flow controls the script off. Add the above discussion to the IIG.

180
181

Question	**1.23 Returning job-uri and job-id when "job-template" attributes are requested.** TS1 is saying that the job attributes job-uri and job-id should be returned in the response to a Get-Jobs operation with requested-attributes of <job-template>, but job-uri and job-id are not in the job-template group. **IPP Bake Off**
Discussion	
Answer 9/30/1998	The "job-uri" and "job-id" attributes are not job-template attributes. This is a bug in the script. Fix the script.

182

183

Question	**1.24 Definition of 'successful-attributes-substituted-or-ignored' and unsupported attribute values in Get-Xxxx operations** Is it required to return a status of 01 when a bogus attribute is included as one of requested attributes of a Get-Jobs operation? Technically, this situation is not covered by the definition of status x0001. The first part of the definition says 'some attributes were ignored". The attribute being "requested-attributes" was not ignored. What was ignored is one of the bvalues (bogus-attribute) of the attribute. The second half of the definition is "unsupported values were substituted with supported values". this wasn't done either, since the unsupported value was ignored. So this status code does not apply. Recommended that the definition gets beefed up to include something like "or unsupported values were ignored". **IPP Bake Off**
Discussion	

Answer 9/30/1998	While the IPP object is NOT REQUIRED to return requested attributes that are unsupported (see Issue 1.18), it is REQUIRED to return the 'successful-attributes-substituted-or-ignored' success code, rather than 'successful-ok'.
	
	MOD 14.1.2.1 'successful-ok' change:
	    The request has succeeded.
	to:
	    The request has succeeded and no request attributes were substituted or ignored.
	
	MOD 14.1.2.2 'successful-ok-ignored-or-substituted-attributes' clarify that it is used for all requests, not just create operations, by changing:
	    The request has succeeded, but some attributes were ignored or unsupported values were substituted with supported values in order to process the job without rejecting it.
	to:
	    The request has succeeded, but some attributes were ignored or unsupported values were substituted with supported values or were ignored in order to perform the operation without rejecting it. These unsupported attributes or values are returned in the Unsupported Attributes group of the response. In the case of Get-Xxxx operations when supplied values of the "requested-attributes" operation attribute are requesting attributes that are not supported, the IPP object MUST return this status code and MAY return the "requested-attributes" attribute in the Unsupported Attribute response group (with the unsupported values only).

184
185

AGREED Resolutions to the IPP Issues List - Model only

Question	**1.25 Can new attribute groups be added through registration?** **Tom Hastings**
Discussion	Section 6 lists keyword attribute values, enum attribute values, attributes, attribute syntaxes, operations, and status codes, but does not mention new attribute groups. What about private attribute groups?
Answer 9/30/1998	Yes, so add the following section to Section 6 after Section 6.4 Operation Extensibility: Attribute groups passed in requests and responses may be registered following the type2 procedures described in Section 6.1. The tags that identify each of the attribute groups are assigned in [IPP-PRO]. For attribute groups, the IPP Designated Expert in consultation with IANA assigns the next attribute group tag code in the appropriate range as specified in [IPP-PRO]. IANA will publish approved attribute group registration specifications as separate files: ftp.isi.edu/iana/assignments/ipp/attribute-group-tags/xxx-yyy-tag.txt where 'xxx-yyy-tag' is the new attribute group tag name.

186

187

Question	**1.26 What about unsupported attribute syntaxes?** Does the implementation respond as if the attribute or value were not supported? If so, then Section 3.2.1.2 should add this condition to the list. **Tom Hastings**
Discussion	There are two situations: 1. The attribute syntax is not supported for any attribute (because the implementation doesn't support any attributes that use that syntax or a new attribute syntax was registered that the implementation doesn't support). 2. The attribute syntax is not supported for a particular attribute (because another new or existing attribute syntax was added to the specification of the attribute) as a registered extension. If an attribute is specified to have more than one attribute syntax, such as (type3 keyword | name), then an implementation MUST support both (all) attribute syntaxes specified in order to support that attribute. Remember that the 'name' attribute syntax is really a short hand for (nameWithLanguage | nameWithoutLangage) and 'text' is a shorthand for (textWithLanguage | textWithoutLanguage).

Answer 9/30/1998	Clarify the following three categories of unsupported attributes in section 3.2.1.2:
	    1. The Printer object does not support the named attribute (no matter what the value).
	    2. The Printer object does support the attribute, but does not support some or all of the particular values supplied by the client (i.e., the Printer object does not have those values in the corresponding supported values attribute).
	by replacing the above with:
	    1. The Printer object does not support the supplied attribute (no matter what the attribute syntax or value).
	    2. The Printer object does support the attribute, but does not support some or all of the particular values supplied by the client (i.e., the Printer object does not have those values in the corresponding supported values attribute) or does not support some or all of the particular attribute syntaxes supplied by the client for the value(s) of the named attribute.
	
	Clarify the following paragraph in Section 3.2.1.2:
	    In the case of a supported attribute with one or more unsupported values, the Printer object simply returns the client-supplied attribute with the unsupported values as supplied by the client. This indicates support for the attribute, but no support for that particular value. If the client supplies a multi-valued attribute with more than one value and the Printer object supports the attribute

	but only supports a subset of the client supplied values, the Printer object MUST return only those values that are unsupported. by replacing "values" with "attribute syntaxes or values" to make: In the case of a supported attribute with one or more unsupported attribute syntaxes or values, the Printer object simply returns the client-supplied attribute with the unsupported attribute syntaxes or values as supplied by the client. This indicates support for the attribute, but no support for that particular attribute syntax or value. If the client supplies a multi-valued attribute with more than one value and the Printer object supports the attribute but only supports a subset of the client supplied attribute syntaxes or values, the Printer object MUST return only those attribute syntaxes or values that are unsupported. Clarify that when the spec for an attribute specifies more than one attribute syntax, then all such specified attribute syntaxes are required to be supported in order to support that attribute. So add the following sentence to the last paragraph of section 4.1: If an attribute specification includes more than one attribute syntax in the sub-section heading, all such attribute syntaxes are required to be supported in order to support the attribute.

188

189

Question	**1.27 How staple multiple documents as one document, but start each document on a new sheet?** The 'single-document' value of "multiple-document-handling" requires that each document not be forced to start on a new sheet. **IPP Bake Off**
Discussion	How about adding a new value? Perhaps called 'single-document-with-new-sheets'
Answer 9/30/1998	Deferred. Such a value can be registered in the future for use with the "multiple-document-handling" Job Template attribute.

190
191

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 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.
Answer 9/30/1998	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 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.

192

193

Question	**1.29 What does an IPP Printer return in a Print-Job response if the job was canceled by another client before the first client had supplied all of the data?** Presumably, the IPP Printer returns an error code that rejects the request, the job does not come into existence? Must the "job-id" and "job-uri" not be re-used (for the next job)? **IPP Bake Off**
Discussion	

Answer 9/30/1998	Add a new server error status code by adding the following new section:
	      14.1.5.9 server-error-job-canceled (0x0508)
	
	      An error indicating that the job has been canceled by an operator or the system while the client was transmitting the data to the IPP Printer. If a job-id and job-uri had been created, then they are returned in the Print-Job, Send-Document, or Send-URI response as usual; otherwise, no job-id and job-uri are returned in the response.

194

195

Question	**1.30 Correct "job-state" for Job-Submit?** An IPP client submits a small job via "job-submit". By the time the IPP printer/print server is putting together a response to the operation, the job has finished printing and been removed as an object from the print system. What should the job-state be in the response? **Hugo Parra**
Discussion	The Model suggests that the Printer return a response before it even accepts the document content (see sections 16.4.8 and 16.4.9). The Job Object Attributes are returned only if the IPP object returns one of the success status codes. Then the job-state would always be "pending" or "pending-held". **Carl Kugler** This issue comes up for the implementation of an IPP Printer object as a server that forwards jobs to devices. If the server is reasonably certain that the job completed successfully, then it should return the job-state as 'completed'. Also the server can keep the job in its "job history" long after the job is no longer in the device. Then a user could query the server and see that the job was in the 'completed' state and completed as specified by the job's "time-at-completed" time which would be the same as the server submitted the job to the device. An alternative is for the server to respond to the client before or while sending the job to the device, instead of waiting until the server has finished sending the job to the device. In this case, the server can return the job's state as 'pending' with the 'job-outgoing' value in the job's "job-state-reasons" attribute. If the server doesn't know for sure whether the job completed successfully (or at all), it could return the (out-of-band) 'unknown' value. All of these alternatives depend on implementation.
Answer 9/30/1998	No change to MOD. Add the above discussion to the IIG.

196

197

Question	**1.31 What is the correct syntax for multi-valued attributes?** Each value in a multi-valued attribute includes its own value-tag. It is syntactically possible then for each value in the list be of a different syntax (integer, uri, nameWithoutLangugage, etc) Is this right? Is this explicitly stated in the documentation? Does it need to be? **Hugo Parra**
Discussion	I think it's right. You can have a mixture of 'type3 keyword' and 'name' values for "job-hold-until-supported", for example. **Carl Kugler**
Answer 9/30/1998	No change to MOD. See the last paragraph of Section 4.1, just before Section 4.1.1 that contains the statement: Most attributes are defined to have a single attribute syntax. However, a few attributes (e.g., "job-sheet", "media", "job-hold-until") are defined to have several attribute syntaxes, depending on the value. These multiple attribute syntaxes are separated by the "|" character in the sub-section heading to indicate the choice. Since each value MUST be tagged as to its attribute syntax in the protocol, a single-valued attribute instance may have any one of its attribute syntaxes and a multi-valued attribute instance may have a mixture of its defined attribute syntaxes. Add question to the FAQ and discussion to the IIG.

198

199

Question	**1.32 Listing of jobs not submitted by IPP?** We've talked about list-jobs somehow differentiating between jobs submitted through IPP and other jobs. Is there a hard requirement? Is it documented? **Hugo Parra**
Discussion	The desire (and recommendation) is to list all the jobs and also have all IPP operations apply to all jobs. Thus, a job submitted via LPR could be canceled via IPP, for example. Some IPP operations (like cancel) require access control... If the user is unknown on the non-IPP job, the access could be considered anonymous (see MOD Section 8.3).
Answer 9/30/1998	Since both the Get-Jobs and Get-Job-Attributes operations refer to Section 8 for security, the following new section will be added to section 8, after Section 8.4 Restricted Queries: 8.5 Queries on jobs submitted using non-IPP protocols If the device that an IPP Printer is representing is able to accept jobs using other job submission protocols in addition to IPP, it is RECOMMEND that such an implementation at least allow such "foreign" jobs to be queried using Get-Jobs returning "job-id" and "job-uri" as 'unknown'. Such an implementation NEED NOT support all of the same IPP job attributes as for IPP jobs. The IPP object returns the 'unknown' out-of-band value for any requested attribute of a foreign job that is supported for IPP jobs, but not for foreign jobs. It is further RECOMMENDED, that the IPP Printer generate "job-id" and "job-uri" values for such "foreign jobs", if possible, so that they may be targets of other IPP operations, such as Get-Job-Attributes and Cancel-Job. Such an implementation also needs to deal with the problem of authentication of such foreign jobs. One approach would be to treat all such foreign jobs as belonging to users other than the user of the IPP client. Another approach would be for the foreign job to belong to 'anonymous'. Only if the IPP client has been authenticated as an operator or administrator of the IPP Printer object, could the foreign jobs be queried by an IPP request. Alternatively, if the security policy is to allow users to query other users' jobs, then the foreign jobs would also be visible to an end-user IPP client using Get-Jobs and Get-Job-Attributes. Amplify the above discussion in the IIG.

200
201

Question	**1.35 Names for enums?**

	Section 14 (Appendix B) of the "Model and Semantics" document includes the following: "The name of the enum is the suggested status message for US English" The name of the enum for unqualified success (0x0000) is 'successful-ok'. Shouldn't its corresponding status message be "successful-ok"? If so, there is another discrepancy in Appendix A of the "Encoding and Transport" document where "OK" is used as the status-message for 'successful-ok'. **Hugo Parra**
Discussion	Also, isn't "successful-ok" redundant? We could save a few bytes and shorten that to "successful" without losing any information. Similarly with "successful-ok-ignored-or-substituted-attributes" and "successful-ok-conflicting-attributes". **Carl Kugler**
Answer 9/30/1998	No change to MOD. Make the editorial change to PRO to change the status message from 'OK' to 'successful-ok'.

202

203

Question	**1.36 Request-id in response when validation fails?** Suppose the Printer object, while parsing an IPP requests, fails to validate the "request-id" in the incoming payload (because the packet was incomplete or because the value is not between 1 and 2**31-1). The documents indicates that the Printer object should return a 'client-error-bad-request' status code. That's fine; now my question: What request-id should the Printer object include in the response (I'm assuming that responses with error status codes must also include version, request-id, charset, etc.)? Should 0 be used to handle this cases? **Hugo Parra**
Discussion	I can't remember if the request-id is the same as a request "transaction-id", but if it is, the server never validates this field. It is set by the client and echoed back by the server in the corresponding response. There is no validity check made by the server (that I am aware of). **Randy Turner** I'm reading section 16.3.3 of the "Model and Semantics" document. **Hugo Parra** I see what you're talking about, but I don't think 16.3.3 should be in the document (just my opinion). The request-id should be an unsigned opaque value to the server. If I were you I would just return whatever value the client gave you; IMHO, it's the safe bet. **Randy Turner** If you get a request ID of 0 (which is invalid) or if the request ID is somehow otherwise unintelligible, then what should the request ID be in the response? We need a special value. 0 is not a legal value for request ID so should we return 0? Does a server really have to reject a request ID of 0? This is a MIB issue not HTTP. But what about other forms of corruption? Every IPP request needs a response. The issue is, should you validate request Ids? Randy says you can't have a corrupt request ID. If you get 4 complete bytes, you just return the ID. If you never get to the point where you have received the entire request ID then use 0 in the return. **IPP WG Meeting**
Answer 9/30/1998	Change the 2nd paragraph of Section 3.1.2: In addition, every invocation of an operation is identified by a "request-id" value. For each request, the client chooses the "request-id" which is an integer (possibly unique depending on client requirements) in the range from 1 to 2**31 - 1 (inclusive). This "request-id" allows clients to manage multiple outstanding requests. The receiving IPP object copies the client supplied "request-id" attribute into the response so that the client can match the response with the correct outstanding request.

to:

> In addition, every invocation of an operation is identified by a "request-id" value. For each request, the client chooses the "request-id" which MUST be an integer (possibly unique depending on client requirements) in the range from 1 to 2**31 - 1 (inclusive). This "request-id" allows clients to manage multiple outstanding requests. The receiving IPP object copies all 32 bits of the client supplied "request-id" attribute into the response so that the client can match the response with the correct outstanding request, even if the "request-id" is out of range. If the request is terminated before the four octets of "request-id" are received, the IPP object returns a response with a "request-id" of 0.

Also change 16.3.3 Validate the request identifier from:

> The Printer object checks to see if the "request-id" attribute supplied by the client is in range. If the value is not between 1 and 2**31 - 1 (inclusive), the Printer object REJECTS the request and returns the 'client-error-bad-request' status code in the response.

to:

> The Printer object SHOULD NOT checks to see if the "request-id" attribute supplied by the client is in range: between 1 and 2**31 - 1 (inclusive), but copies all 32 bits.

204

205

Question	**1.37 None value for empty sets**
	I have discovered what I consider to be an unfortunate decision with regard to the "none" value for empty sets?
	The model documens states that the "none" value should be used as the value of a 1SetOf when the set is empty. In most cases, sets that are potentially empty contain keywords so the keyword "none" is used, but for the 3 finishings attributes, the values are enums and thus the empty set is represented by the enum 3. Currently there are no other attributes with 1SetOf values which can be empty and can contain values that are not keywords. This exception requires special code and is a potential place for bugs. It would have been better if we had chosen an out-of-band value, either "no-value" or some new value, such as "none". At this late date, it is probably too late to change this, though I wonder if other implementations have dealt with this special case properly.
	Bob Herriot
Discussion	
Answer 9/30/1998	No change to MOD. A 'none' value for enums is different than 'none' in keywords. Put a note in the IIG about this difference in handling 'none' depending on the attribute syntax.

206

Question	**1.38 Syntax for boolean?**
	In section 4.1.11 the words say that "The 'boolean' attribute syntax is similar to an enum with only two values: 'true' and 'false'. "
	And in section 4.1.4 the words says "The 'enum' attribute syntax is an enumerated integer value that is in the range from 1 to 2**31 - 1 (MAX)."
	Does this mean, that a boolean attribute got a 32 bit size value?
	In the protocol document, it says that a boolean is a byte size!
	Henrik Holst
Discussion	
Answer 9/30/1998	Change the description for 'boolean' in Section 4.1.11 from: The 'boolean' attribute syntax is similar to an enum with only two values: 'true' and 'false'.
	to:
	The 'boolean' attribute syntax has only two values: 'true' and 'false'.

207

208

Question	**1.39 Get-Jobs, my-jobs='true', and 'requesting-user-name'?** In section 3.2.6.1 'Get-Jobs Request' I wondered, if the attribute 'my-jobs' is present and set to TRUE, MUST the 'requesting-user-name' attribute be there to, and if it's not present what should the IPP printer do? **Henrik Holst**
Discussion	If the client does not supply a value for "requesting-user-name", the printer MUST assume that the client is supplying some anonymous name, such as "anonymous". **Carl Kugler**
Answer 9/30/1998	No change to MOD. Section 8.3 describes the various cases of "requesting-user-name" being present and not for any operation. Add question to the FAQ with a pointer to Section 8.3.

209

Question	**1.40 HTTP server resource?** We've established that the "HTTP server resource" referred to in the document is either 1) an IPP Printer, or 2) an IPP Job. If we substitute the words "IPP Printer (or IPP Job)" for "HTTP Server resource" in the original sentence, we get: > Once the IPP Printer (or IPP Job) begins to process the HTTP request, it might get the reference to the appropriate IPP Printer object from either the HTTP URI (using to the context of the HTTP server for relative URLs) or from the URI within the operation request; the choice is up to the implementation. I cannot understand this sentence. What are the words "appropriate IPP Printer object" referring to in this sentence? Why would a Printer or Job object processing an IPP request need a "reference to the appropriate IPP Printer object"? What is the Printer or Job supposed to do with the reference? Note: I realize that the sentence in the document says "begins to process the HTTP request", not "IPP request". However, if the "HTTP server resource" processes only the HTTP part of the request (and not the IPP), then there is no choice to use the URI within the IPP operation request, so the sentence makes no sense. **Carl Kugler**
Discussion	I tend to follow the saying "Be conservative in what you send, and liberal in what you accept..." Whether the text says MUST or not, IMHO we should be designing clients and servers to handle a "connection: close" header whenever it is received and still function normally, albeit with possibly less performance.

	Since I am not working on a client, I cannot speak for what clients are or will actually do, but I do think the client end should drive the connection status, whevever possible. **Randy Turner** I agree that the client and server must accept the Connection: close header. I'm wondering how to satisfy the requirement that the client and server MUST include this header for the last operation in a sequence of operations. Specifically, how do the client and server know, a priori, that the current operation is the last operation in a sequence (and therefore MUST include the Co nnection: close header)? **Carl Kugler**
Answer 9/30/1998	Duplicates Issue 2.14. This is a PRO issue, not a MOD issue.

210

Question	**1.41 Empty attribute and delimiter?** Some server implementations do not add delimiters for empty attribute group, and some client implementations assumed delimiters will always be there even if the attribute group is empty. We should make it clear if delimiter is required if the corresponding attribute group is empty. **Yuji Sasaki**
Discussion	
Answer 9/30/1998	Duplicate of Issue 1.17.

211

212

Question	**1.42 Spooling jobs?** Many "print server" products...such as Intel NetPort or HP JetDirect... has limited resource(i.e memory or HDD capacity), so it is impossible to "spool" job document. They can support job commands(Get-jobs, Get-job-attributes, etc), however because of lack of spooling capabilities, they can handle only one job at a time. Until the first job is complete, the following jobs cannot be processed. But many IPP test suite assumed the server can "spool" jobs, so caused many errors on my (JCI) IPP print server implementation, which has only 128Kbyte RAM and of course no HDD. Is it required for all IPP servers to MUST be able to spool jobs? **Yuji Sasaki**
Discussion	
Answer 9/30/1998	No change to MOD. It is not required for an implementation to spool. Don't run spooling tests on non-spooling printers. Some of the scripts can be fixed so that they do not require multiple jobs.

213
214

Question	**1.43 Target URI?** The IPP specification says the "third" operation attribute MUST be the target URI, however some implementation does not include target URI at all, and some others includes the URI but not at "third" place. **Yuji Sasaki**
Discussion	
Answer 9/30/1998	REQUIRED for clients to supply in the request and in the proper place. Change Section 16 so that the IPP object is NOT REQUIRED to check the request for the target URI. See answer for Issue 1.22.

215
216

Question	**1.44 Target URI and HTTP URI?** When issuing JOB related commands, the target URI could be a printer-URI with a job-ID or simply a job-URI. But the relation between target URI and HTTP URI seems to be unclear. For example, sending a Cancel-job request to a JOB-URI(as HTTP URI) with a printer-URI and a job-ID as the target URI is OK? **Yuji Sasaki**
Discussion	
Answer 9/30/1998	Same as Issue 2.14.

217

218

Question	**1.45 Querying support of text/name with language?** Many IPP implementations did not support text/name with language attributes, and some were crashed when they received "with language" attributes. Should we have another "-supported" attribute, like "text-or-name-with-language-attributes-supported" (maybe too long ;-)? **Yuji Sasaki**
Discussion	
Answer 9/30/1998	No new attribute is needed. Implementations should be fixed to support both textWithLanguage and textWithoutLanguage as specified in Section 4.1.1 and 4.1.2 2nd paragraph; same for nameWithLanguage and nameWithoutLanguage. Need to write a script to test this.

219
220