Remaining IPP/1.1 Issues, Updated during 5/27/99 meeting

File: remaining-issues-summary.doc
Date: 5/28/99

Comments on the resolutions to the following IPP/1.1 issues had been raised on the mailing list as WG Last Call comments and were discussed at the IPP WG 5/26-27 meeting. This paper shows the agreements reached at the meeting, which included a telecon with the Area Directory, Keith Moore. These agreements need confirmation on the mailing list. The editors will be updating the IPP/1.1 Model and Semantics document and the Encoding and Transport document to reflect these agreements. Any comments should be send to the mailing list by Thursday, June 10, 1999. Then the resulting documents will be sent as Internet-Drafts on Friday, June 11, 1999. The following issues were discussed:

ISSUE 17: Client display of absolute time for job attributes 1
ISSUE 30: Should "job-state-reasons" and "printer-state-reasons" be REQUIRED for an IPP/1.1 Printer2
ISSUE 31: How indicate a ripped job that is waiting for the marker? 4
ISSUE 32: Is Digest REQUIRED for an IPP Client and an IPP Printer to support? 5
ISSUE 35: What error code to return on Print-URI or Send-URI if document not accessible? 6
ISSUE 36: Don't require 1.0 support and add REQUIRED "ipp-version-supported" attribute 9

Revision marks show changes from the May Internet-Drafts

Bold Italics are explanations about following text.

ISSUE 17: Client display of absolute time for job attributes

We agreed to Alternative 1:

Alternative 1: An IPP Printer MUST support "time-at-created(integer(MIN:MAX))", "time-at-processing(integer(MIN:MAX))", and "time-at-completed(integer(MIN:MAX))" Job Description attributes ~~(and keep the 3 new date-time-at-xxx attributes as RECOMMENDED)~~.

~~Alternative 2: An IPP Printer MUST support either:~~

~~option (a) the set of 3 "time-at-xxx(integer) and "job-printer-up-time(integer) Job attributes~~

~~OR:~~

~~option (b) the set of 3 "date-time-at-xxx(dateTime)" Job attributes and the "printer-current-time(dateTime)" Printer attribute~~

~~Implementations MAY support both options (a) and (b). It is RECOMMEND that option (b) be supported, if only one option is supported.~~

ISSUE 30: What "job-state-reasons" and "printer-state-reasons" values are REQUIRED for an IPP/1.1 Printer?

The following clarification of 4.3.8 "job-state-reasons"

~~Implementation of these values is OPTIONAL, i.e., a Printer NEED NOT implement them, even if (1) the output device supports the functionality represented by the reason and (2) is available to the Printer object implementation.~~ These values MAY be used with any job state or states for which the reason makes sense. Some of these value definitions indicate conformance requirements; the rest are OPTIONAL.

The following state reasons have the indicated conformance requirements added to reflect conformance requirements already states elsewhere in the document. All of the other values are OPTIONAL:

’none’: There are no reasons for the job’s current state. This state reason is semantically equivalent to "job-state-reasons" without any value and MUST be used when there is no other value, since the 1setOf attribute syntax requires at least one value.

’document-access-error’: After accepting a Print-URI or Send-URI request, the Printer could not access one or more documents passed by reference. This reason is intended to cover any file access problem, including file does not exist and access denied because of an access control problem. Whether the Printer aborts the job and moves the job to the ’aborted’ job state or prints all documents that are accessible and moves the job to the ’completed’ job state and adds the ’completed-with-errors’ value in the job’s "job-state-reasons" attribute depends on implementation and/or site policy. This value SHOULD be supported if the Print-URI or Send-URI operations are supported. Issue 30 and 35

’job-hold-until-specified’: The value of the job’s "job-hold-until" attribute was specified with a time period that is still in the future. The job MUST NOT be a candidate for processing until this reason is removed and there are no other reasons to hold the job. This value MUST be supported if the "job-hold-until" Job Template attribute is supported.

’job-canceled-by-user’: The job was canceled by the owner of the job using the Cancel-Job request, i.e., by a user whose authenticated identity is the same as the value of the originating user that created the Job object, or by some other authorized end-user, such as a member of the job owner’s security group. This value SHOULD be supported.

’job-canceled-by-operator’: The job was canceled by the operator using the Cancel-Job request, i.e., by a user who has been authenticated as having operator privileges (whether local or remote). If the security policy is to allow anyone to cancel anyone’s job, then this value may be used when the job is canceled by other than the owner of the job. For such a security policy, in effect, everyone is an operator as far as canceling jobs with IPP is concerned. This value SHOULD be supported if the implementation permits canceling by other than the owner of the job.

’job-canceled-at-device’: The job was canceled by an unidentified local user, i.e., a user at a console at the device. This value SHOULD be supported if the implementation supports canceling jobs at the console.

’aborted-by-system’: The job (1) is in the process of being aborted, (2) has been aborted by the system and placed in the ’aborted’ state, or (3) has been aborted by the system and placed in the ’pendingheld’ state, so that a user or operator can manually try the job again. This value SHOULD be supported.

’processing-to-stop-point’: The requester has issued a Cancel-Job operation or the Printer object has aborted the job, but is still performing some actions on the job until a specified stop point occurs or job termination/cleanup is completed.

If the implementation requires some measurable time to cancel the job in the ’processing’ or ’processing-stopped’ job states, the IPP object MUST use this value ~~This reason is RECOMMENDED to be used in conjunction with the ’processing’ job state~~ to indicate that the Printer object is still performing some actions on the job while the job remains in the ’processing’ or ’processing-stopped’ state. After all the job’s job description attributes have stopped incrementing, the Printer object moves the job from the ’processing’ state to the ’canceled’ or ’aborted’ job states.

’job-completed-successfully’: The job completed successfully. This value SHOULD be supported.

’job-completed-with-warnings’: The job completed with warnings. This value SHOULD be supported if the implementation detects warnings.

’job-completed-with-errors’: The job completed with errors (and possibly warnings too). This value SHOULD be supported if the implementation detects errors.

’job-restartable’ - This job is retained (see section 4.3..7.1 and is currently able to be restarted using the Restart-Job operation (see section 3.3.7)). If ’job-restartable’ is a value of the job’s ’job-state-reasons’ attribute, then the IPP object MUST accept a Restart-Job operation for that job. This value SHOULD be supported if the Restart-Job operation is supported.

Suggested text for addition to section 4.4.12 "printer-state-reasons":

Some of the these value definitions indicate conformance requirements; the rest are OPTIONAL.

’none’: There are not reasons. This state reason is semantically equivalent to "printer-state-reasons" without any value and MUST be used, since the 1setOf attribute syntax requires at least one value.

’moving-to-paused’: Someone has paused the Printer object using the Pause-Printer operation (see section 3.2.7 or other means, but the device(s) are taking an appreciable time to stop. Later, when all output has stopped, the "printer-state" becomes ’stopped’, and the ’paused’ value replaces the ’moving-to-paused’ value in the "printer-state-reasons" attribute. This value MUST be supported, if the Pause-Printer operation is supported and the implementation takes significant time to pause a device in certain circumstances.

’paused’: Someone has paused the Printer object using the Pause-Printer operation (see section 3.2.7 or other means and the Printer object’s "printer-state" is ’stopped’. In this state, a Printer MUST NOT produce printed output, but it MUST perform other operations requested by a client. If a Printer had been printing a job when the Printer was paused, the Printer MUST resume printing that job when the Printer is no longer paused and leave no evidence in the printed output of such a pause. This value MUST be supported, if the Pause-Printer operation is supported.

’spool-area-full’: The limit of persistent storage allocated for spooling has been reached. The Printer is temporarily unable to accept more jobs. The Printer will remove this value when it is able to accept more jobs. This value SHOULD be used by a non-spooling Printer that only accepts one or a small number jobs at a time or a spooling Printer that has filled the spool space. Issue 20 Issue 30 and Issue 31

ISSUE 31: How indicate a ripped job that is waiting for the marker?

Simplified "printer-state" descriptions that preserves IPP/1.0 compatibility while allowing multiple rips with a marker:

4.4.14 printer-state (type1 enum)

This REQUIRED Printer attribute identifies the current state of the device. The "printer-state reasons" attribute augments the "printer-state" attribute to give more detailed information about the Printer in the given printer state.

A Printer object need only update this attribute before responding to an operation which requests the attribute; the Printer object NEED NOT update this attribute continually, since asynchronous event notification is not part of IPP/1.1. A Printer NEED NOT implement all values if they are not applicable to a given implementation.

The following standard enum values are defined:

Value Symbolic Name and Description

’3’ ’idle’: Indicates that new jobs can start processing without waiting. ~~If a Printer receives a job (whose required resources are ready) while in this state, such a job is likely to transit into the ’processing’ state immediately, unless required resources are not ready. If the "printer-state-reasons" attribute contains any reasons, they MUST be reasons that would not prevent a job from transiting into the ’processing’ state immediately, e.g., ’toner-low’.~~

~~If a Printer can interpret one or more jobs while marking a job, then it is idle if it is available to interpret jobs even while marking a job. Issue 31~~

~~If a Printer controls more than one output device, the above definition implies that a Printer is ’idle’ if at least one output device is idle, i.e., the IPP Printer is available to immediately start processing a job if a client submitted it.~~

’4’ ’processing’: Indicates that jobs are processing; new jobs will wait before processing. ~~Indicates that the system is running and that one or more jobs are in the ’processing’ state. If a Printer receives a job (whose required resources are ready) while in this state, such a job MUST transit into the ’pending’ state immediately. Such a job MUST transit into the ’processing’ state only after jobs ahead of it complete. If the "printer-state-reasons" attribute contains any reasons, they MUST be reasons that do not prevent the current job from printing, e.g. ’toner-low’.~~

~~If a Printer can interpret one or more jobs while marking a job and receives a job (whose required resources are ready) while in this state, such a received job MAY~~

~~transit into the ’processing’ state along with the job that is being marked, if any. Issue 31~~

~~If a Printer controls more than one output device, the above definition implies that a Printer is ’processing’ if at least one output device is processing, and none is idle.~~

’5’ ’stopped’: Indicates that no jobs can be processed and intervention is required. ~~If a Printer receives a job (whose required resources are ready) while in this state, such a job MUST transit into the ’pending’ state immediately. Such a job MUST transit into the ’processing’ state only after some human fixes the problem that stopped the printer and after jobs ahead of it complete processing. Issue 30 The "printer-state-reasons" attribute MUST contain at least one reason, e.g. ’media-jam’, which prevents it from either processing the current job or transitioning a ’pending’ job to the ’processing’ state.~~

~~If a Printer can interpret one or more jobs while marking a job and receives a job (whose required resources are ready) while in this state, such a submitted job MAY transit into the ’processing’ state in order to be interpreted even while the Printer is in the ’stopped’ state. However, before such a job can be completed, a human needs to fix the problem. Issue 31~~

~~If a Printer controls more than one output device, the above definition implies that a Printer is ’stopped’ only if all output devices are stopped.~~

~~Note: it is tempting to define ’stopped’ as when a sufficient number of output devices are stopped and leave it to an implementation to define the sufficient number. But such a rule complicates the definition of ’stopped’ and ’processing’. For example, with this alternate definition of ’stopped’, a job can move from ’pending’ to ’processing’ without human intervention, even though the Printer is stopped.~~

Values of "job-state-reasons", such as ’spool-area-full’~~[new] queue-full’~~ and ’stopped-partly’, MAY be used to provide further information.

Put the above deleted definitions for simple, multi-rip, and fan-out configurations into the IIG. This deleted text appeared in the May Internet-Drafts. Also discuss the difficulty of the simple IPP/1.1 Printer object abstraction to represent a set of devices that have widely differing capabilities, i.e., a color and a black and white printer.

Clarification of 4.4.12 "printer-state-reasons":

Some of these values indicate conformance requirements; the rest are OPTIONAL. Issue 30

Text for clarification of ’spool-area-full’ "printer-state-reasons" value:

~~’queue-full’~~’spool-area-full’: The limit of persistent storage allocated for spooling has been reached. The Printer is temporarily unable to accept more jobs. The Printer will remove this value when it is able to accept more jobs. This value MAY be used by a non-spooling Printer that only accepts one or a small number jobs at a time or a spooling Printer that has filled the spool space.

ISSUE 32: Is Digest REQUIRED for an IPP Client and an IPP Printer to support?

Text for Section 5.1 Client Conformance:

A client MUST~~/SHOULD~~ ~~[which is to be determined in consultation with the Area Director]~~ support Client Authentication as defined in the IPP/1.1 Encoding and Transport document [ipp-pro]. A client SHOULD support Operation Privacy and Server Authentication as defined in the IPP/1.1 Encoding and Transport document [IPP-PRO]. See also ~~[Ipp-Mod]~~ section 8 of this document.

Text for a new sub-section to Section 5.2 IPP Object Conformance:

5.2.7 Security

An IPP Printer implementation ~~MUST/~~SHOULD ~~[which is to be determined in consultation with the Area Director]~~ contain support for Client Authentication as defined in the IPP/1.1 Encoding and Transport document [IPP-PRO]. A Printer implementation MAY allow an administrator to configure the Printer so that all, some, or none of the users are authenticated. See also ~~[IPP-MOD]~~ section 8 of this document.

An IPP Printer implementation SHOULD contain support for Operation Privacy and Server Authentication as defined in the IPP/1.1 Encoding and Transport document [ipp-pro]. A Printer implementation MAY allow an administrator to configure the degree of support for Operation Privacy and Server Authentication. See also ~~[IPP-MOD]~~ section 8 of this document.

ISSUE 35: What error code to return on Print-URI or Send-URI if document not accessible?

2. Add OPTIONAL "job-document-access-errors (1setOf text)" Job Description and "document-access-error(text)" Print-URI/Send-URI operation attribute. For protocol errors, such as HTTP or FTP errors, standard values would be the error code in parentheses, followed by the URL~~"reference-uri" 'uriScheme' followed by a hyphen (-) and an error number.~~ For example:

```
        ~~http-~~(XXX) http://ftp.pwg.org/pub/pwg/ipp/new_MOD/ipp-model-v11-990510.pdf
        ~~ftp-~~(XXX) ftp://ftp.pwg.org/pub/pwg/ipp/new_MOD/ipp-model-v11-990510.pdf
```

For example, the ~~'http-404' keyword~~ value: (404) http://... would indicate that the HTTP server could not find the resource. (Note: most Internet protocols use decimal error codes (unlike IPP), so the ASCII keyword representation is in decimal.)

3. Instead of adding "debug-info", add the "detailed-status-message (text(MAX))" operation attribute and "job-detailed-status-messages (1setOf text(MAX))" Job Description attribute which contains detailed information that is not localized.

Reorganize ~~Add to~~ and add to section 3.1.6 Operation Status Codes and Messages:

3.1.6 Operation Response Status Code~~s~~ and Status Messages:

Every operation response includes a REQUIRED "status-code" parameter and an OPTIONAL "status-message" operation attribute, and an OPTIONAL "detailed-status-message" operation attribute. The Print-URI and Send-URI response MAY include an OPTIONAL "document-access-error" operation attribute.

3.1.6.1 "status-code" (type2 enum)

The REQUIRED "status-code" parameter provides information on the processing of a request.

The status code is intended for use by automata. A client implementation of IPP SHOULD convert status code values into any localized message that has semantic meaning to the end user.

The "status-code" value is a numeric value that has semantic meaning. The "status-code" syntax is similar to a "type2 enum" (see section 4.1 on "Attributes Syntaxes") except that values can range only from 0x0000 to 0x7FFF. Section 13 describes the status codes, assigns the numeric values, and suggests a corresponding status message for each status code for use by the client when the user's natural language is English.

3.1.6.2 "status-message" (text(255))

~~A~~ The OPTIONAL "status-message" operation attribute provides a short textual description of the status of the operation. The "status-message" attribute's syntax is "text(255)", so the maximum length is 255 octets (see section 4.1.1). ~~t~~The status message is intended for the human end user. If a response does include a "status-message" attribute, an IPP client NEED NOT examine or display the messages, however it SHOULD do so in some implementation specific manner. The "status-message" is especially useful for a later version of a Printer object to return as supplemental information for the human user to accompany a status code that an earlier version of a client might not understand.

If the Printer object supports the "status-message" operation attribute, the Printer object MUST be able to generate this message in any of the natural languages identified by the Printer object's "generated-natural-language-supported" attribute (see the "attributes-natural-language" operation attribute specified in section 3.1.4.1. Section 13 suggests the text for the status message returned by the Printer for use with the English natural language.

As described in section 3.1.4.1 for any returned 'text' attribute, if there is a choice for generating this message, the Printer object uses the natural language indicated by the value of the "attributes-natural-language" in the client request if supported, otherwise the Printer object uses the value in the Printer object's own "natural-language-configured" attribute.

If the Printer object supports the "status-message" operation attribute, it SHOULD use the REQUIRED 'utf-8' charset to return a status message for the following error status codes (see section 13): 'client-error-bad-request', 'client-error-charset-not-supported', 'server-error-internal-error', 'server-error-operation-not-supported', and 'server-error-version-not-supported'. In this case, it MUST set the value of the "attributes-charset" operation attribute to 'utf-8' in the error response.

3.1.6.3 "detailed-status-message" (text(MAX))

The OPTIONAL "detailed-status-message" operation attribute provides additional more detailed technical and implementation-specific information about the operation. The "detailed-status-message" attribute's syntax is "text(MAX)", so the maximum length is 1023 octets (see section 4.1.1). If the Printer objects supports the "detailed-status-message" operation attribute, neither the Printer nor the client localizes the message, since it is intended for use by the system administrator or other experienced technical persons. Clients MUST NOT attempt to parse the value of this attribute. See "document-access-error" operation attribute (section 3.1.6.4) for additional errors that a program can process.

3.1.6.4 document-access-error(text(MAX))

This OPTIONAL operation attribute provides additional information about any document access errors encountered by the Printer before it returned a response to the Print-URI (section 3.2.2) or Send-URI (section 3.3.1) operation. For errors in the protocol identified by the URI scheme in the "document-uri" operation attribute, such as 'http:' or 'ftp:', the error code is returned in parentheses, followed by the URI. For example:

```
(404) http://ftp.pwg.org/pub/pwg/ipp/new_MOD/ipp-model-v11-990510.pdf
```

Most Internet protocols use decimal error codes (unlike IPP), so the ASCII error code representation is in decimal.

Suggested text for section 3.2.2 Print-URI Operation:

Replace the sentences:

See The Implementer's Guide [IPP-IIG] for suggested additional checks. The Printer NEED NOT follow the reference and validate the contents of the reference.

with:

The IPP Printer MAY validate the accessibility of the document as part of the operation or subsequently. If the Printer determines an accessibility problem before returning an operation response, it rejects the request and returns the 'client-error-document-access-error' status code. The Printer MAY also return a specific document access error code using the "document-access-error" operation attribute (see section 3.1.6.4).

If the Printer determines this document accessibility problem after accepting the request and returning an operation response with one of the successful status codes, the Printer adds the 'document-access-error' value to the job's "job-state-reasons" attribute and MAY populate the job's "job-document-access-errors" Job Description attribute (see section 4.3.11). See The Implementer's Guide [IPP-IIG] for suggested additional checks.

Add two new sections after section 4.3.9 job-state-message (text(MAX)):

4.3.10 job-detailed-status-messages (1setOf text(MAX))

This attribute specifies additional detailed and technical information about the job. Neither the Printer nor the client localizes the message(s), since they are intended for use by the system administrator or other experienced technical persons. Clients MUST NOT attempt to parse the value of this attribute. See "document-access-error" (section 4.3.11) for additional errors that a program can process.

4.3.11 job-document-access-errors (1setOf text(MAX))

This attribute provides additional information about each document access error for this job encountered by the Printer after it returned a response to the Print-URI or Send-URI operation and subsequently attempted to access document(s) supplied in the Print-URI or Send-URI operation. For errors in the protocol that is identified by the URI scheme in the "document-uri" operation attribute, such as 'http:' or 'ftp:', the error code is returned in parentheses, followed by the URI. For example:

```
    (404) http://ftp.pwg.org/pub/pwg/ipp/new_MOD/ipp-model-v11-990510.pdf
```

Most Internet protocols use decimal error codes (unlike IPP), so the ASCII error code representation is in decimal.

ISSUE 36: Don’t require 1.0 support and add REQUIRED "ipp-version-supported" attribute

If the major version number matches, but the minor version number does not, the Printer SHOULD accept and attempt to process the request, or MAY reject the request and return the ’server-error-version-not-supported’ status code. In all cases, the Printer MUST return the nearest version number that it supports. For example, suppose that an IPP/1.2 Printer supports versions ’1.1’ and ’1.2’. The following responses are conforming:

Client supplies	Printer Accept Request?	Printer returns
1.0	yes (SHOULD)	1.1
	no (SHOULD NOT)	1.1
1.1	yes (MUST)	1.1
1.2	yes (MUST)	1.2
1.3	yes (SHOULD)	1.2
	no (SHOULD NOT)	1.2

Put this table in the IIG.

Changes to the text for the new attribute:

4.4.14 ipp-versions-supported (1setOf type2 keyword)

This REQUIRED attribute identifies the IPP protocol version(s) that this Printer supports, including major and minor versions, i.e., the version numbers for which this Printer implementation meets the conformance requirements. For version number validation, the Printer matches the (two-octet binary) "version-number" parameter supplied by the client in each request (see sections 3.1.1 and 3.1.8) with the (US-ASCII) keyword values of this attribute.

~~If an IPP Printer receives a request with the "version-number" parameter set to a (two-octet binary) value that does not correspond to one of the (US-ASCII) keyword values of this attribute, it MUST reject the request and return the ’server-error-version-not-supported’ status code. See Section 3.1.8.~~

The following standard keyword values are defined:

’1.0’: Meets the conformance requirement of IPP version 1.0 as specified in RFC 2566 [RFC2566] and RFC 2565 [RFC2565] including any extensions registered according to Section 6 and any extension defined in this version or any future version of the IPP "Model and Semantics" document or the IPP

"Encoding and Transport" document following the rules, if any, when the "version-number" parameter is '1.0'.

'1.1': Meets the conformance requirement of IPP version 1.1 as specified in this document and [IPP-PRO] including any extensions registered according to Section 6 and any extension defined in any future versions of the IPP "Model and Semantics" document or the IPP Encoding and Transport document following the rules, if any, when the "version-number" parameter is '1.1'.

Modification to section 3.1.8 Versions:

3.1.8 Versions

Each operation request and response carries with it a "version-number" parameter. Each value of the "version-number" is in the form "X.Y" where X is the major version number and Y is the minor version number. By including a version number in the client request, it allows the client to identify which version of IPP it is interested in using, i.e., the version whose conformance requirements the client may be depending upon the Printer to meet.

If the IPP object does not support that major version number supplied by the client, i.e., the major version field of the "version-number" parameter does not match any of the values of the Printer's "ipp-versions-supported" (see section 4.4.14), the object MUST respond~~s~~ with a status code of 'server-error-version-not-supported' along with the closest version number that is supported (see section 13.1.5.4). If the major version number is supported, but the minor version number is not, the IPP object SHOULD accept and attempt to perform the request (or reject the request if the operation is not supported), else it rejects the request and return the 'server-error-version-not-supported' status code. In all cases, the IPP object MUST return the "version-number" that it supports that is closest to the version number supplied by the client in the request.

There is no version negotiation per se. However, if after receiving a 'server-error-version-not-supported' status code from an IPP object, ~~there is nothing that prevents~~ a client SHOULD ~~from~~ try~~ing~~ again with a different version number. A client MAY also determine the versions supported either from a directory that conforms to Appendix E (see section 16) or by querying the Printer object's "ipp-versions-supported" attribute (see section 4.4.14) to determine which versions are supported. Issue 36

~~In order to conform to IPP/1.1,~~ An IPP object implementations MUST support version '1.1', i.e., meet the conformance requirements for IPP/1.1 as specified in this document and [IPP-PRO]. An IPP object implementation SHOULD support version '1.0', i.e., meet the conformance requirements for IPP/1.0 [RFC2566 and RFC2565]. and SHOULD support version '1.0', i.e., meet the conformance requirements for IPP/1.0 [RFC2566 and RFC2565]. A client MAY also determine the versions supported either from a directory that conforms to Appendix E (see section 16) or by querying the Printer object's "ipp-versions-supported" attribute (see section 4.4.14) to determine which versions are supported. Issue 36

There is only one notion of "version number" that covers both IPP Model and IPP Protocol changes. Thus the version number MUST change when introducing a new version of the Model and Semantics document [IPP-MOD] or a new version of the "Encoding and Transport" document [IPP-PRO].

Note: Changes to the major version number of the Model and Semantics document indicate structural or syntactic changes that make it impossible for older version of IPP clients and Printer objects to correctly parse and correctly process the new or changed attributes, operations and responses. If the major version number changes, the minor version numbers is set to zero. As an example, adding the REQUIRED "ipp-attribute-fidelity" attribute to version '1.1' (if it had not been part of version '1.1'), would have required a change to the major version number, since an IPP/1.0 Printer would not have processed a request with the correct semantics that contained the "ipp-attribute-fidelity" attribute that it did not know about. Items that might affect the changing of the major version number include any changes to the Model and Semantics document [IPP-MOD] or the "Encoding and Transport" document [IPP-PRO] itself, such as:

- reordering of ordered attributes or attribute sets
- changes to the syntax of existing attributes
- adding REQUIRED (for an IPP object to support) operation attribute groups
- adding values to existing REQUIRED operation attributes
- adding REQUIRED operations

Changes to the minor version number indicate the addition of new features, attributes and attribute values that may not be understood by all IPP objects, but which can be ignored if not understood. Items that might affect the changing of the minor version number include any changes to the model objects and attributes but not the encoding and transport rules [IPP-PRO] (except adding attribute syntaxes). Examples of such changes are:

- grouping all extensions not included in a previous version into a new version
- adding new attribute values
- adding new object attributes
- adding OPTIONAL (for an IPP object to support) operation attributes (i.e., those attributes that an IPP object can ignore without confusing clients)
- adding OPTIONAL (for an IPP object to support) operation attribute groups (i.e., those attributes that an IPP object can ignore without confusing clients)
- adding new attribute syntaxes
- adding OPTIONAL operations
- changing Job Description attributes or Printer Description attributes from OPTIONAL to REQUIRED or vice versa.
- adding OPTIONAL attribute syntaxes to an existing attribute. Issue 33

The encoding of the "version-number" MUST NOT change over any version number (either major or minor). This rule guarantees that all future versions will be backwards compatible with all previous versions (at least for checking the "version-number"). In addition, any protocol elements (attributes, error codes, tags, etc.) that are not carried forward from one version to the next are deprecated so that they can never be reused with new semantics.

Implementations that support a certain version NEED NOT support ALL previous versions. As each new version is defined (through the release of a new specification), that ~~major~~ version will specify which previous versions MUST and which versions SHOULD be supported in compliant implementations. Issue 36

Change to section 5.2.4 [Conformance of] Versions:

Clients MUST support version 1.1, i.e., MUST meet the conformance requirements for clients specified in this document and [IPP-PRO] and SHOULD also support version 1.0, i.e., SHOULD meet the conformance requirements for clients as specified in [RFC2566] and [RFC2565]. IPP objects MUST support version 1.1, i.e., MUST meet the conformance requirements for IPP objects specified in this document and [IPP-PRO] and SHOULD also support version 1.0, i.e., SHOULD meet the conformance requirements for IPP objects as specified in [RFC2566] and [RFC2565].

IPP clients MUST send a version '1.1' in requests and SHOULD try alternate versions if they receive a 'server-error-version-not-supported' error return. IPP objects MUST accept a version 1.1 request (or reject the request if the operation is not supported). IPP objects SHOULD accept any request with the major version '1' (or reject the request if the operation is not supported). See section 3.1.8.

Changes to section 13.1.5.4 server-error-version-not-supported (0x0503):

13.1.5.4 server-error-version-not-supported (0x0503)

The IPP object does not support, or refuses to support, the IPP protocol version that was supplied as the value of the "version-number" operation parameter in the request. The IPP object is indicating that it is unable or unwilling to complete the request using the same major and minor version number as supplied in the request other than with this error message. The error response SHOULD contain a "status-message" attribute describing why that version is not supported and what other versions are supported by that IPP object. See sections 3.1.6 and 3.1.8. Issue 11

The error response MUST identify in the "version-number" operation parameter the closest version number that the IPP object does support. For example, if a client supplies version '1.0' and an IPP/1.1 object supports version '1.0', then it MUST respond with version '1.0' in all responses to such a request. If the IPP/1.1 object does not support version '1.0', then it ~~MUST~~ SHOULD accept the request and respond with version '1.1' or MAY reject the request and respond with this error code and version '1.1'. If the IPP/1.1 object does not support version '1.0', then it SHOULD accept the request and respond with version '1.1' or MAY reject the request and respond with this error code and version '1.1'. If a client supplies a version '1.2' the IPP/1.1 object SHOULD accept the request and return version '1.1' or MAY respond with this error code and version '1.1'. See sections 3.1.8 and 4.4.14. Issue 36