IPP Bake-Off 2 Results Summary

Table of Contents

1 Revision Log

3/16/99	pjz	Started with first Bake-Off summary. Updated issues list.
3/17/99	pjz	Updated tables.
3/18/99	pjz	added additional issues captured by Tom Hastings
3/19/99	pjz	Completed updates

2 Overview

The second IPP Bake-Off held March 9th to 12th. It was hosted by Novel in Orem Utah. The Bake-Off was an unqualified success. The 24 participating organizations were:

Apple	AUCO,	Axis, Canon	EFI
Epson	Extended Systems	Fuji Xerox	Genoa
Hewlett Packard	i-data	IBM	JCI
Komatsu	Kyocera	Lexmark	Microsoft
Netsilicon	Novell	Ricoh	Shinesoft
Sun	Tektronix	Xerox	

There were 11 IPP Client implementations and 27 IPP Printer implementations. Out of the 297 possible Client/Printer combinations, 296 were successful in completing a simple "print-job" operation. That is a success rate of 99.66%. The previous Bake off had 8 Clients and 16 Printer with a success rate of 96.9%.

Other aspects of IPP were tested for the first time including HTTP chunking and security. We tested 127 combinations of chunking clients with IPP Printers. Of those tested 118 worked fine resulting in a success rate of 92.9%. We knew going into this test that some Printers did not yet support chunking. The limited security testing that was done was a complete success. There were 43 combinations supporting Basic authentication. They all worked. There were 2 combinations supporting Digest authentication and they worked. (One printer knew going in its Digest authentication did not work. The printer participated only to prove it was broken.) Three combinations supported SSL. One combination used Basic authentication over SSL. These also worked.

All the IPP operations, Printer attributes, Job Description attributes and attribute syntaxes were tested. The majority of the Job Description attributes and Operational attributes were tested. We have not strenuously tested the semantics and interactions of these operations and attributes. There are just too many participants and a lack of automated tools to accomplish such a rigorous test.

Twenty six issues were raised over the four day event. These issues will be brought to the IPP mailing list and discussed.

The major benefit of the Bake-Off was bringing together the implementers of IPP from across the industry. The cooperation between the engineers was remarkable. All were sharing their IPP expertise and working together for the benefit of all. Every participating vendor will have an improved implementation of IPP as a direct result of this event.

The First day of the Bake-off was for setup. This was accomplished in less than half a day. Some time was spent discussing methodology and prioritizing objectives. A prioritized list and the IPP Test Plan was used as the basis for Bake-off testing. The URL for the prioritized list is "ftp://ftp.pwg.org/pub/pwg/ipp/new_TES/Bake-Off2-Official-List.pdf" . The URL for the Test Plan is "ftp://ftp.pwg.org/pub/pwg/ipp/new_TES/IPP-Test-Plan-990219.pdf"
The remaining time was spent testing basic printing, chunking, security, print by reference, create-job/send-document and the set 1 operations.

3 Summary Tables

3.1 Print Submission Interoperability Matrix

This matrix shows the results of submitting a simple print job from IPP Clients to IPP Printerss. The participants of the Bake-Off agreed on rules of anonymity. One of the rules was to use designations to identify implementations. In the following table IPP Clients are on the horizontal and IPP Printers are on the vertical. A key at the bottom of the table is provided to interpret the results.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C11	C12	S/F
P1	P	P	P	P	P	P	P	P	P	P	P	11/0
P2	P	P	P	P	S	S	P	P	P	P	S	11/0
P3	S	S	S	S	S	S	S	S	S	S	S	11/0
P4	P	P	P	P	S	S	P	P	P	P	S	11/0
P5	P	P	P	P	P	P	P	P	P	P	P	11/0
P6	S	S	S	S	S	S	S	S	S	S	S	11/0
P7	P	P	P	P	P	P	S	P	P	P	S	11/0
P8	S	S	S	S	S	S	S	S	S	S	S	11/0
P9	S	P	P	P	P	P	P	P	P	P	S	11/0
P10	P	P	P	P	S	P	P	P	P	P	P	11/0
P11	P	P	P	P	S	P	P	P	P	P	P	11/0
P12	S	S	S	S	S	S	S	S	S	S	S	11/0
P13	P	P	S	P	P	P	P	P	P	S	P	11/0
P14	P	P	P	P	P	P	P	P	P	P	P	11/0
P15	S	P	S	S	S	P	S	S	S	S	S	11/0
P16	S	S	S	S	S	S	S	S	S	S	S	11/0
P17	S	S	P	P	S	S	P	P	P	P	P	11/0
P18	S	S	S	S	S	S	S	S	S	S	S	11/0
P19	P	P	P	P	P	P	P	P	P	P	S	11/0
P20	P	P	P	P	S	P	P	P	P	P	S	11/0
P21	S	S	S	S	S	F	S	S	P	S	S	10/1
P22	P	P	P	P	P	P	P	P	P	P	P	11/0
P23	S	S	S	S	S	S	S	S	S	S	S	11/0
P24	P	P	P	P	P	P	P	P	P	P	S	11/0
P25	S	S	S	S	S	S	S	S	S	S	S	11/0
P26	P	P	P	P	P	P	P	P	P	S	S	11/0
P27	S	S	S	S	S	S	S	S	S	S	S	11/0
S/F	27/0	27/0	27/0	27/0	27/0	26/1	27/0	27/0	27/0	27/0	27/0	**296/1**

S = Success
P = Success (Paper output)
F = Failure

3.2 Chunking Interoperability Matrix

This matrix shows the results of submitting a simple print job from IPP Clients to IPP Printers. The print request is sent using HTTP chunking. IPP Clients are on the horizontal and IPP Printers are on the vertical. A key at the bottom of the table is provided to interpret the results.

	C1	C3	C4	C7	C8	C12	S/F
P1	S	S		S		S	4/0
P2	S	S	S	S	S	S	6/0
P3	S	S	S	S	S	S	6/0
P4	S	S	S	S	S	S	6/0
P5	S	S	S	S	S	S	6/0
P6							0/0
P7	S	S	S	S	F (500)	S	5/1
P8	S	S	S		S	S	5/0
P9	S	S		S	S	S	5/0
P10		F	F		S		1/2
P11	S	S	S	S	S	S	6/0
P12	S	S	S	S	S	S	6/0
P13	S	S	S	S	S	S	6/0
P14	F (400)	S	F (Inval req)	S	F (no msg)	S	3/3
P15	S	S	S	S	S	S	6/0
P16							0/0
P17	S	S	S	S	S	S	6/0
P18	S	S			S	S	4/0
P19					F (400)		0/1
P20	S	S	S	S	S	S	6/0
P21	S	S	S	S	S	S	6/0
P22		S	S		S	S	4/0
P23	S	S	S	F		S	4/1
P24					F		0/1
P25	S	S	S		S	S	5/0
P26	S	S	S	S	S	S	6/0
P27	S	S	S	S	S	S	6/0
S/F	20/1	22/1	18/2	17/1	19/4	22/0	118/9

S = Success
F = Failure
(*) = error code/message

3.3 Security Interoperability Matrix

This matrix shows the results of issuing an IPP request from IPP Clients to IPP Printers. IPP Clients are on the horizontal and IPP Printers are on the vertical. The security method is listed with the results. Note that P1's Digest implementation was not part of the test. It was known to be incorrect and tested only to verify it was indeed broken.
All combinations tested succeeded.

	C2	C4	C5	C6	C8	C9
P1	Basic=S Digest=F	Basic=S Digest=F	Basic=S Digest=F	Basic=S Digest=F	Basic=S Digest=F	Basic=S Digest=F
P4	Basic=S	Basic=S	Basic=S		Basic=S	Basic=S
P5	Basic=S	Basic=S Digest=S	Basic=S	Basic=S SSL=S	Basic=S	Basic=S Digest=S
P9	Basic=S	Basic=S			Basic=S	Basic=S
P10	Basic=S	Basic=S			Basic=S	Basic=S
P12	Basic=S				Basic=S	
P13			Basic=S	Basic=S SSL=S Basic+SSL=S	Basic=S	
P15						
P18	Basic=S	Basic=S	Basic=S			
P19	Basic=S	Basic=S			Basic=S	
P22	Basic=S	Basic=S				Basic=S
P24	Basic=S	Basic=S		Basic=S	Basic=S	Basic=S
P27				SSL=S		

3.4 Create-Job Interoperability Matrix

This matrix shows the results of issuing of 'create-job' and 'send-document' requests from IPP Clients to IPP Printers. IPP Clients are on the horizontal and IPP Printers are on the vertical.
All combinations tested succeeded.

	C1	C2	C4	C9	C11	S/F
P5	S	S	S	S	S	5/0
P8		S				1/0
P9	S	S	S	S		4/0
P12	S	S	S	S	S	5/0
P19	S	S	S		S	4/0
P23		S	S			2/0
S/F	4/0	6/0	5/0	3/0	3/0	21/0

3.5 Print-Uri Interoperability Matrix

This matrix shows the results of issuing of 'create-job' and 'send-document' requests from IPP Clients to IPP Printers. IPP Clients are on the horizontal and IPP Printers are on the vertical.
All combinations tested succeeded.

	C1	C2	C4	C9	C11	S/F
P5	S	S	S	S	S	5/0
P9	S	S	S			3/0
P19	S	S				2/0
P23		S				1/0
P26		S	S	S		3/0
P27		S	S	S		3/0
S/F	3/0	6/0	4/0	3/0	1/0	17/0

3.6 Set 1 Interoperability Matrix

This matrix shows the results of testing the Set One operations from IPP Clients to IPP Printers. IPP Clients are on the horizontal and IPP Printers are on the vertical.
It is unclear what level of testing was performed. The test plan had several steps to perform. It is not known if these steps were followed. This was one of the last tests performed and not carefully monitored. All combinations tested reported success.

NOTE: The Bake-Off was for IPP v1.0. The 'Set One' operations are not part of the v1.0 specification.

	C2	C5	C6	C9	S/F
P7	S	S	S	S	4/0
P8					0/0
P9	S				1/0
P13	S				1/0
P115	S	S		S	3/0
P18					0/0
P19	S				1/0
S/F	5/0	2/0	1/0	2/0	10/0

3.7 Ipp:// scheme support Matrix

This matrix shows the results of testing the 'ipp://' scheme from the C9 IPP Client to the IPP Printers. This was tested using IPP v1.0 and IPP v1.1 in the IPP header. The 'accept' column indicates that the version and scheme was accepted in a 'print-job' operation. The 'return' column indicates the 'ipp://' scheme was returned for the job URL in a 'get-jobs' operation. NOTE: I am unsure what 'Rel' indicates and counted those as untested.

NOTE: The Bake-Off was for IPP v1.0. The 'ipp://' scheme is not part of the v1.0 specification.

	IPP 1.0 Accept ipp://	IPP 1.0 Return ipp://	IPP 1.1 Accept ipp://	IPP 1.1 Return ipp://
P1	N	N	N	N
P2	Y	N	Y	N
P3				
P4	Y	Rel	Y	Rel
P5	Y	Y	Y	Y
P6	N	N	N	N
P7	N	N	N	N
P8	Y	N	N	N
P9	Y	Y	N	N
P10	Y	N	Y	N
P11				
P12	Y	N	Y	N
P13				
P14	Y	N	Y	N
P15	Y	N	N	N
P16	N	N	N	N
P17	N	N	N	N
P18	N	N	N	N
P19				
P20	Y	Y	Y	Y
P21				
P22	Y	N	Y	N
P23	N	N	N	N
P24	Y	N	Y	N
P25	Y	Y	Y	Y
P26	Y	Y	N	N
P27	Y	Y	N	N
Y/N	15/7	6/15	10/12	3/18

3.8 HTTP get on IPP URL support Matrix

This matrix shows the results of issuing an HTTP 'Get' on the IPP URL. This was tested using Netscape Navigator without a proxy and Internet Explorer with a proxy. The 'Y' indicates the printer returned a page of information. Any other indication indicates a failure to return a page.

ID	Netscape Navigator No Proxy	Internet Explorer Proxy
P1	Y	Y
P2	Doc cont. no data	Time out
P3	Y but redir	Y but redir.
P4	N 404	N 404
P5	N 405	N 405
P6	N 404	N 404
P7	Y CGI	Y CGI
P8	Time out	Time out
P9	Y	Y CGI
P10	Doc cont. no data	Unknown HTTP Error
P11	Y	Y
P12	Y Exe file	Y Exe file
P13	Doc cont. no data	N Invalid. response
P14	Y	Y
P15	Y	Y
P16	N	N
P17	N 405	Time out
P18	Y	Y
P19	Y	Y
P20	Y for HTTP	Y for HTTP
P21	Y	Time out
P22	Time out	Time out
P23	Time out	Time out
P24	N Object not found	N Object not found
P25	Doc contains no data	Invalid response
P26	N 400	N 400
P27	N 400	N 400
Y/N	12/15	11/16

3.9 Operation Coverage

An attribute, group or operation is considered tested when two independent implementations are able to interoperate. For the purposes of the Bake-off, that means at least two IPP Printers and two IPP Clients understand the attribute, group or operation. Partial successes are listed giving the number of printers and clients that interoperate. A value of 'No' in the 'Tested' column indicates that no printer supported the operation, attribute or group.

ID	Operation	Mandatory	Tested
OC01	print-job	Yes	Yes
OC04	validate-job	Yes	Yes
OC05	get-printer-attributes	Yes	Yes

ID	Operation	Mandatory	Tested
OC06	get-jobs	Yes	Yes
OC07	get-job-attributes	Yes	Yes
OC08	cancel-job	Yes	Yes
OC09	print-uri	No	Yes
OC10	create-job	No	Yes
OC11	send-document	No	Yes
OC12	send-uri	No	Yes
OC13	hold-job	No	Yes
OC14	release-job	No	Yes
OC15	pause-printer	No	Yes
OC16	resume-printer	No	Yes
OC17	purge-printer	No	Yes
OC18	restart-job	No	Yes

3.10 Operational Attributes Coverage

An attribute, group or operation is considered tested when two independent implementations are able to interoperate. For the purposes of the Bake-off, that means at least two IPP Printers and two IPP Clients understand the attribute, group or operation. Partial successes are listed giving the number of printers and clients that interoperate. A value of 'No' in the 'Tested' column indicates that no printer supported the operation, attribute or group.

Operational Attribute Coverage					
ID	**Operational Attribute**	**Group**	**Comment**	**Mandatory**	**Tested**
OA01		job-attribute	Tested by get-jobs	Yes	Yes
OA02		printer-attribute	Tested by get-printer-attributes	Yes	Yes
OA03		unsupported-attributes	Tested by print-job with unsupported attributes	Yes	Yes
OA04		operational-attribute	See OA08 to OA30	Yes	Yes
OA05	version-number	preamble	Test with any operation	Yes	Yes
OA06	operation-id	preamble		Yes	Yes
OA07	request-id	preamble	Test with any operation	Yes	Yes
OA08	attributes-charset	operational-attribute	Must support utf-8	Yes	Yes
OA09	attributes-natural-language	operational-attribute	Test imposes en-us requirement	Yes	Yes
OA10	printer-uri	operational-attribute	This or OA11 must be 3rd attribute	Conditional	Yes
OA11	job-uri	operational-attribute	This or OA10 must be 3rd attribute	Conditional	Yes

Operational Attribute Coverage					
ID	**Operational Attribute**	**Group**	**Comment**	**Mandatory**	**Tested**
OA12	job-id	operational-attribute	This must be 4th attribute	Conditional	Yes
OA13	job-name	operational-attribute	Only for print & validate operations	Optional	Yes
OA14	requesting-user-name	operational-attribute		Optional	Yes
OA15	document-uri	operational-attribute	Only for print-uri	Conditional	Yes
OA16	last-document	operational-attribute	Only for send-uri and send-document	Conditional	Yes
OA17	status code	preamble	Test with any response	Yes	Yes
OA18	status-message	operational-attribute	Test with any response	Optional	Yes
OA19	compression	operational-attribute	Only for print & validate operations	Optional	Yes
OA20	document-natural-language	operational-attribute	Only for print & validate operations	Optional	No
OA21	ipp-attribute-fidelity	operational-attribute	Only for print & validate operations	Optional	Yes
OA22	job-impressions	operational-attribute	Only for print & validate operations	Optional	No
OA23	job-k-octets	operational-attribute	Only for print & validate operations	Optional	No
OA24	job-media-sheets	operational-attribute	Only for print & validate operations	Optional	2 Clients 1 Printer
OA25	limit	operational-attribute	Only for get-jobs operations	Optional	Yes
OA26	message	operational-attribute	Only for cancel operations	Optional	Yes
OA27	my-jobs	operational-attribute	Only for get-jobs operations	Optional	Yes
OA28	requested-attributes	operational-attribute	Only for get-* operations	Optional	Yes
OA29	Document-format	operational-attribute	Only for print operations	Optional	Yes
OA30	which-jobs	operational-attribute	Only for get-jobs operations	Optional	Yes

3.11 Attribute Coverage

An attribute, group or operation is considered tested when two independent implementations are able to interoperate. For the purposes of the Bake-off, that means at least two IPP Printers and two IPP Clients understand the attribute, group or operation. Partial successes are listed giving the number of printers and clients that interoperate. A value of 'No' in the 'Tested' column indicates that no printer supported the operation, attribute or group.

3.11.1 Printer Description

Printer Description				
ID	**Attribute**	**Syntax**	**Mandatory**	**Tested**
PD01	printer-uri-supported	uri	Yes	Yes
PD02	uri-security-supported	1setOf type2 keyword	Yes	Yes
PD03	printer-name	name	Yes	Yes
PD04	printer-state	type1 enum	Yes	Yes
PD05	operations-supported	1setOf type2 enum	Yes	Yes
PD06	charset-configured	charset	Yes	Yes
PD07	charset-supported	1setOf charset	Yes	Yes
PD08	natural-language-configured	naturalLanguage	Yes	Yes
PD09	generated-natural-language-supported	1setOf naturalLanguage	Yes	Yes
PD10	printer-is-accepting-jobs	boolean	Yes	Yes
PD11	pdl-override-supported	type2 keyword	Yes	Yes
PD12	printer-up-time	integer	Yes	Yes
PD13	printer-location	text	No	Yes
PD14	printer-info	text	No	Yes
PD15	printer-more-info	uri	No	Yes
PD16	printer-driver-installer	uri	No	Yes
PD17	printer-make-and-model	text	No	Yes
PD18	printer-more-info-manufacturer	uri	No	Yes
PD19	printer-state-reasons	1setOf type2 keyword	No	Yes
PD20	printer-state-message	text	No	Yes
PD21	document-format-default	mimeMediaType	Yes	Yes
PD22	document-format-supported	1setOf mimeMediaType	Yes	Yes
PD23	queued-job-count	integer	No	Yes
PD24	printer-message-from-operator	text	No	Yes
PD25	color-supported	boolean	No	Yes
PD26	reference-uri-schemes-supported	1setOf uriScheme	No	Yes
PD27	printer-current-time	dateTime	No	Yes
PD28	multiple-operation-time-out	integer	No	Yes
PD29	compression-supported	1setOf type3 keyword	No	Yes
PD30	job-k-octets-supported	rangeOfInteger	No	Yes

Printer Description				
ID	**Attribute**	**Syntax**	**Mandatory**	**Tested**
PD31	job-impressions-supported	rangeOfInteger	No	Yes
PD32	job-media-sheets-supported	rangeOfInteger	No	Yes

3.11.2 Job Template

Job Template				
ID	**Attribute**	**Syntax**	**Mandatory**	**Tested**
JT01	job-priority	integer	No	Yes
JT02	job-priority-default	integer	No	Yes
JT03	job-priority-supported	integer	No	Yes
JT04	job-hold-until	type4 keyword | name	No	Yes
JT05	job-hold-until-default	type4 keyword | name	No	Yes
JT06	job-hold-until-supported	1setOf type4 keyword | name	No	Yes
JT07	job-sheets	type4 keyword | name	No	Yes
JT08	job-sheets-default	type4 keyword | name	No	Yes
JT09	job-sheets-supported	1setOf type4 keyword | name	No	Yes
JT10	multiple-document-handling	type2 keyword	No	Yes
JT11	multiple-document-handling-default	type2 keyword	No	Yes
JT12	multiple-document-handling-supported	1setOf type2 keyword	No	Yes
JT13	copies	integer	No	Yes
JT14	copies-default	integer	No	Yes
JT15	copies-supported	integer	No	Yes
JT16	finishings	1setOf type2 enum	No	Yes
JT17	finishings-default	1setOf type2 enum	No	Yes
JT18	finishings-supported	1setOf type2 enum	No	Yes
JT19	page-ranges	1setOf rangeOfInteger	No	Yes
JT20	page-ranges-supported	boolean	No	Yes
JT21	sides	type2 keyword	No	Yes
JT22	sides-default	type2 keyword	No	Yes
JT23	sides-supported	1setOf type2 keyword	No	Yes
JT24	number-up	integer	No	Yes
JT25	number-up-default	integer	No	Yes
JT26	number-up-supported	1setOf integer | rangeOfInteger	No	Yes

Job Template				
ID	**Attribute**	**Syntax**	**Mandatory**	**Tested**
JT27	orientation-requested	type2	No	Yes
JT28	orientation-requested-default	type2	No	Yes
JT29	orientation-requested-supported	1setOf type2	No	Yes
JT30	media	type4 keyword | name	No	Yes
JT31	media-default	type4 keyword | name	No	Yes
JT32	media-supported	1setOf type4 keyword | name	No	Yes
JT33	media-ready	1setOf type4 keyword | name	No	Yes
JT34	printer-resolution	resolution	No	Yes
JT35	printer-resolution-default	resolution	No	Yes
JT36	printer-resolution-supported	1setOf resolution	No	Yes
JT37	print-quality	type2 enum	No	Yes
JT38	print-quality-default	type2 enum	No	Yes
JT39	print-quality-supported	1setOf type2 enum	No	Yes

3.11.3 Job Description

Job Description				
ID	**Attribute**	**Syntax**	**Mandatory**	**Tested**
JD01	job-uri	uri	Yes	Yes
JD02	job-id	integer	Yes	Yes
JD03	job-printer-uri	uri	Yes	Yes
JD04	job-name	name	Yes	Yes
JD05	job-originating-user-name	name	Yes	Yes
JD06	job-state	type1 enum	Yes	Yes
JD07	attributes-charset	charset	Yes	Yes
JD08	attributes-natural-language	naturalLanguage	Yes	Yes
JD09	job-more-info	uri	No	No
JD10	job-state-reasons	1setOf type2 keyword	No	Yes
JD11	job-state-message	text	No	1 Printer 2 Client
JD12	number-of-documents	integer	No	Yes
JD13	output-device-assigned	name	No	1 Printer 2 Client
JD14	time-at-creation	integer	No	Yes
JD15	time-at-processing	integer	No	Yes
JD16	time-at-completed	integer	No	Yes
JD17	number-of-intervening-jobs	integer	No	Yes
JD18	job-message-from-operator	text	No	1 Printer 2 Client
JD19	job-k-octets	integer	No	Yes
JD20	job-impressions	integer	No	No
JD21	job-media-sheets	integer	No	1 Printer 2 Client
JD22	job-k-octets-processed	integer	No	Yes

Job Description				
ID	**Attribute**	**Syntax**	**Mandatory**	**Tested**
JD23	job-impressions-completed	integer	No	1 Printer 2 Client
JD24	job-media-sheets-completed	integer	No	Yes

3.12 Syntax Coverage

Syntax Coverage				
ID	**Syntax**	**Attribute**	**Mandatory**	**Tested**
SC01	text	printer-state-message	No	Yes
SC02	textWithLanguage	Printer-location	No	Yes
SC03	name	printer-name	Yes	Yes
SC04	nameWithLanguage	printer-name	Yes	Yes
SC05	keyword	pdl-override-supported	Yes	Yes
SC06	enum	printer-state	Yes	Yes
SC07	uri	printer-uri-supported	Yes	Yes
SC08	UriScheme (1 setOf)	reference-uri-schemes-supported	No	Yes
SC09	charset	charset-configured	Yes	Yes
SC10	naturalLanguage	natural-language-configured	Yes	Yes
SC11	mimeMediaType	document-format-default	No	Yes
SC12	octetString			No
SC13	boolean	printer-is-accepting-jobs	Yes	Yes
SC14	integer	printer-up-time	Yes	Yes
SC15	rangeOfInteger	job-k-octets-supported	No	Yes
SC16	dateTime	printer-current-time	No	Yes
SC17	resolution	printer-resolution	No	Yes
SC18	1setOf X (1setOf type2 enum)	operations-supported	Yes	Yes

4 Issues

1) PROBLEM: Is 'application/octet-stream REQUIRED?
   Is application/octet-stream REQUIRED. IPP/1.0 appears not to require it, while IPP/1.1 indicated "REQUIRED".

2) ADDITION: We would like to add another operation that forces the server to generate a 401 authentication challenge.
   This is very useful for a client to be able to get into identified mode as soon as possible. Today you have to wait to be challenged by the server, which may never happen – or happens at an unpredictable time. Unless somebody has a different solution (Microsoft)

3) ISSUE: How reject down stream auto-sensed unsupported PDL?
   If auto-sensing happens AFTER the job is accepted (as opposed to auto-sensing at submit time before returning the response), what does the implementation do?
   Presumably, it is similar to encountering a mal-formed PDL. So the implementation aborts the job, puts the job in the 'aborted' state and sets the 'aborted-by-system' value in the job's "job-state-reasons", if supported.

4) PROBLEM: Client closes slow channel
   Some IPP Printer implementations, such as forwarding servers, want to accept an IPP job, even though the down stream channel is being used at the moment by another job stream that the device supports. Rejecting the job would mean that an IPP job might never get in, since these other protocols queue the request.
   However, some clients close the channel when it is flowed controlled off for too long a time?
   Suggested fix: Clients MUST NOT close the channel when flowed controlled off. Clients SHOULD do Get-Printer-Attributes and determine state of the device. Alert user if the printer is stopped. Let user decide whether to abort the job transmission or not. Add a new success-ok-but-very-busy status code?

5) PROBLEM: Client closes stopped device
   When a non-spooling printer is accepting data and putting it on media and runs into a problem, such as paper out or paper jam, what should it do?
   Returning an error is not user friendly, if fixing the problem would allow the job to complete normally.
   Suggested fix: clients must not close the channel. Clients SHOULD do Get-Printer-Attributes and determine state of the device. Alert user if the printer is stopped. Let user decide whether to abort the job transmission or not.

6) PROBLEM: IPP server supports deflate and gzip.
   If client sets "compression attribute = deflate" and sends gziped data, what error does IPP server return to client?

7) CONVENTION: Please implement Manufacturer make and model printer attribute and send the .INF file model name of the printer.
   If you do this we will automatically install the correct driver (if we have it) (Microsoft)

8) ISSUE: In IPP/1.0 Model and semantics 3.2.6.1, the definition for "limit", "which-jobs" and "my-jobs" is contradicting each other.
   The problem is that the definition for "which-jobs" and "my-jobs" states that a group of job MUST be returned. (Stefan Andersson Axis Communication AB)

9) PROBLEM: Customers become very unhappy when they go to the printer to pick up their job and a ream of PostScript source code is sitting in theoutput bin.
   Cause: A PostScript datastream is accidentally sent to a PCL printer.
   IPP Issue: IPP needs to clarify the standard in section 3.2.1.1 of the Model and Semantics document.
   Lines 1219-1221 state that:
   > If the client does not supply [the document format] attribute,
   > the Printer object assumes that the document data is in the
   > format defined by the Printer object's "document-format-default"
   > attribute.

   I would like to see the following clarification:
   > If the client does not supply [the document format] attribute

and the Printer object is not able to auto-sense the document format at print-job request time, the Printer object assumes that the document data is in the format defined by the Printer object's "document-format-default" attribute.

If the Printer object senses that the document format is PostScript, then job should be rejected if it is being sent to a PCL-only printer. The 'application/octet-stream' mechanism discussed in section 4.1.9 does not seem to be helpful in this case, because it appears to assume that the auto-sensing occurs at document processing time. Until the document is actually "ripped", the document format remains unknown. So it seems to me that lines 2453-2476 do not address the problem described above where the wrong document format is submitted. These lines, rather, seem to apply to the case of a printer that handles multiple document formats and assumes that the submitted document is in one of the supported formats.

10) ISSUE: How distinguish between submit vs processing auto-sense?

There are two different implementations of auto-sensing:

at print submit time BEFORE the Print-Job or Send-Document responds

at document processing (ripping) time AFTER the Print-Job or Send-Document has accepted the job.

The description of 'application/octet-stream' doesn't clarify whether one, the other or both is meant. How can a client determine which is supported?

Possible solutions:

1. Clarify that 'application/octet-stream' means either depending on implementation
2. Add a new value that means auto-sense at request time and clarify that 'application/octet-stream' means at processing time.
3. Add a new value that means auto-sense at processing time and clarify that 'application/octet-stream' means at request time.
4. Do 1 and add two new values that mean at request time and at processing time.

11) ISSUE: If a server receives a request with a document format which is not supported, it returns the client-error-document-format-not-supported (0x040A) status code. Is it also necessary to include document format in the unsupported attribute group?

We suggest text which says it need not be supplied in the unsupported group.

12) ISSUE: length fields for the "UNSUPPORTED" tag

IPP/1.0: Model and Semantics, 16 Nov 1998, 3.2.1.2, Group 2 (unsupported attributes) -- states that in the case of an unsupported attribute name, the printer object should return a substituted out of band value of "unsupported". This impression is strengthened by the reference to section 4.1, where it gives the legal out of band values, none of which is an empty string.

This appears to conflict with Internet Printing Protocol/1.0: Encoding and Transport, 16 Nov 1998, section 3.10, where it states that the value length must be 0 and the value empty. (Claudio Cordova, Wade Mergenthal Xerox Corp.)

13) PROBLEM: What job-state value should be returned in the Create-Job response? Pending, pending-held, or either depending on implementation?

The problem with 'pending' is that the job is not a "candidate to start processing" as the definition states. The 'pending-held' state seems more reasonable. Its definition is: 'pending-held': The job is not a candidate for processing for any number of reasons but will return to the 'pending' state as soon as the reasons are no longer present. The job's "job-state-reason" attribute MUST indicate why the job is no longer a candidate for processing. Also there is a "job-state-reason" value 'job-incoming' which states: 'job-incoming': The Create-Job operation has been accepted by the Printer, but the Printer is expecting additional Send-Document and/or Send-URI operations and/or is accessing/accepting document data. But "job-state-reasons" is OPTIONAL. Do we mandate it or recommend it if supporting Create-Job?

14) PROBLEM: Job-state for a forwarding server?
What job-state value should be returned in the Print-Job response for an IPP object that forwards the data over a one-way interface, such as a parallel port or LPD? pending, processing, completed, or unknown? Unknown is the strict interpretation of section 4.3.7 "job-state", but it isn't very user friendly. The "job-state" SHOULD reflect the actual job state, but these implementations have no idea when the job actually starts or finishes.
How about a new "job-state-reasons" value: 'queued-in-device' (from PWG Job Monitoring MIB)?

15) ISSUE: 'unknown' and 'unsupported' Out of band values.
It is very unclear from the spec as to whether or not you should use the word 'unknown' (or unsupported in that case) as the value for attributes that are unknown.
You can read it that you set the length equal to zero and set the type to 'unknown'. You can also read it as saying you set the value to the string'unknown'.
This is not helped by the spec saying – you must set the length to zero and then telling a client what to do with a non-zero length. (Microsoft)

16) CONVENTION: GetPrinterAttributes Polling
Some client polls printer periodically by GetPrinterAttributes without specifying requested^attributes. So printer has to reply all attributes. It consumes printer resouce. Client should specify requested-attriubtes, if it wants to get printer status.

17) ISSUE: What are clients doing with printers that don't support absolute time? How can client display an absolute time (which is what is useful for a user)?
Possible Solution
Get Uptime from printer
Get Job(s)
Calculate Display time = job tick time – uptime + local client absolute time. The down side is that the client has to get the uptime every time (Microsoft)

18) PROBLEM: Return all errors on Print-Job fidelity=true
If ipp-attributes-fidelity=true, MUST all attributes that are not supported, be returned, or can just the first error be returned?

19) ISSUE: User Performing the Operation
The Send-Document and Send-URI commands need the following clarification with regard to the user performing the operation user. In the requesting-user-name section of Send-Document add:
The user performing the Send-Document operation must be the same as for the Create-Job operation that created the job. The printer determines the user performing the operation from the requesting-user-name or the underlying authentication mechanism as described in Section 8.3 of the model document.

The wording in the Send-URI section would imply that the above change applies to Send-URI as well.

20) PROBLEM: Non-spooling printers accept/reject additional jobs
Some IPP Printer implementations reject a second Print-Job (or Create-Job) while they are processing a Print-Job. Other IPP Printer implementations, such as forwarding servers and non-spooling printers, accept some number of subsequent jobs, but flow control them off until the first job is finished.
Suggested fix: Add a new success-ok-but-very-busy status code so that clients and servers (acting as

clients) would know. Also finish our notification extension so that a device that rejects the submit could subscribe for when the device is ready to accept another job.

21) ISSUE: Does 'none' uri-security-supported mean Basic/Digest?
Section 4.4.2 "uri-security-supported" 'none' values says: 'none': There are no secure communication channel protocols in use for the given URI. Should be clarified that the REQUIRED Basic and Digest are intended for the 'none' value. (Hugo Parra)

22) ISSUE: Status code on variable-length attributes that are 'too short'
IPP defines a status code 'client-error-request-value-too-long' for a variable-length attribute that exceeds the maximum length allowed by the attribute. However, it is not clear what status code to use in the opposite case, i.e. the supplied attribute value is shorter than the requirement. In the current spec, this problem will arise when a 0-length value is supplied in 'keyword' attributes. In this case, should the request be rejected with status code 'client-error-request-value-too-long' or 'client-error-bad-request' ? Furthermore, if ipp-attribute-fidelity is false, should the request be rejected at all ? (Robert's opinion is that, the request should be accepted with the problematic value ignored, even though it violated the 'keyword' syntax) (Jason Chien-Hung Chen)

23) ISSUE: There seems to be some misunderstanding about the unsupported-attributes group.
Some implementations return the unsupported-attributes group on a get-attributes operation. The unsupported-attributes presumably contains all the attributes the implementation knows about but does not support. I do not believe this is the proper use of the unsupported-attributes group. Do we need a clarification in the specification.

24) ISSUE What status does Get-Jobs return when no jobs?
Should Get-Jobs return 'successful-ok' when there are no jobs to be returned? The client can see that the Jobs group contains no jobs from the response. Returning an error may confuse the client. Some implementations returned 'client-error-not-found' error code.

25) ISSUE - May an IPP object return more Operation attributes?
It is ok for an IPP object to return additional operation attributes in a response (as an extension to the standard)? If so, then the client MUST ignore or do something with them. (Hugo Parra)

26) ISSUE - May an IPP object return additional groups?
It is ok for an IPP object to return additional groups of attributes in a response (as an extension to the standard)? For example, returning the "job-state" and "job-state-reasons" in a Hold-Job, Release-Job, and/or Cancel-Job operation. What about newly registered groups of attributes. If so, then the client MUST ignore or do something with them. (Hugo Parra)