

Bake-Off 3 results

This document contains the results from IPP Bake-Off 3 hosted by Oak Technologies October 17 to 20.

Table of Contents

1	Summary.....	3
2	Issues	5
3	Basic Interoperability	8
3.1	Mixed version Interoperability.....	8
3.2	IPP v1.1 Only	8
3.3	IPP v1.0 Only	9
3.4	IPP 1.0 Client/1.1 Printer	9
3.5	IPP 1.1 Client/1.0 Printer	9
4	Security.....	10
4.1	Basic Security.....	10
4.2	Digest Security	10
4.3	SSLv3.....	10
4.4	TLS	11
5	Notification	11
5.1	Notification results matrix	11
5.2	Sample Mailto Notification.....	11
5.2.1	Raw text.....	11
5.2.2	HTML View	12
6	Attribute Coverage	13
6.1	Attribute Coverage Summary.....	15
7	Firewalls	15
8	HTTP Proxies.....	15

1 Summary

The third IPP Bake-Off held October 17 to 20. It was hosted by Oak Technologies in Woburn Massachusetts. The Bake-Off was a success, though some participants wanted more time to test with the rich set of IPP implementations.

Participating companies: Axent Technologies Inc., Canon, Electronics for Imaging Inc., Epson, IBM, i-data International, Japan Computer Industry, McAfee.com, Microsoft, Neteon Inc., NETsilicon Inc., Novell, Oak Technologies, Quality Logic, Ricoh, SEH Computertechnik GmbH, Xerox

The 18 participants provided 17 IPP Printers, 9 IPP Clients, 2 firewalls and 2 HTTP Proxies. Out of the 153 possible combinations of Clients and Printers, 151 were tested.

- The overall success rate was 93%.
- Limiting the scope to IPP v1.1 provided a success rate of 96%.
- With IPP v1.0 Clients and v1.1 Printers, the success rate was 100%.
- The tests with v1.1 Clients and v1.0 Printers resulted in a success rate of 31%, which is not surprising given that some printer implementations explicitly disallowed that combination. Some Clients were able to retry in v1.0 mode raising the success rate to 69%. It should also be noted that for v1.0 Printers that allow v1.1 Clients the success rate was 100%.

The majority of the failures can be attributed to one of two causes. The major cause of unresolved failures was due to IPP Clients that had problems with IPP Printers that sent HTTP “100 continue” messages. This was recognized as an implementation error. The other cause was v1.0 Printers that explicitly disallowed v1.1 Clients. IPP inherently provides a mechanism that allows minor version mismatches to be gracefully handled. The minor version mismatch was recognized as an unnecessary printer restriction.

Security testing went well with both SSLv3 and TLS having no failures with their limited number of participants (8 and 7 respectively). Basic authentication had the most participants (59) with a success rate of 93%. The most common cause of failure here was the “100 continue” problem previously mentioned. Digest authentication was the poorest performer with 31 participants and a success rate of 68%. Only a few of the failures were due to the “100 continue” problem.

Firewall and HTTP proxy testing was a complete success. The testing with the firewalls demonstrated that administrators could set policy regarding IPP printing. Firewalls were able block, selectively allow or allow unrestricted printing between IPP Clients and Printers. The firewalls further demonstrated that they could add an additional layer of security requiring IPP Clients to authenticate to the firewall before allowing the IPP request through to a designated printer. The HTTP proxies operated transparently when used in IPP printing. No security interactions or caching issues were discovered.

The IPP notification testing gave early implementers a chance to shake down their implementations. Out of the 25 combinations that were reported tested, only two complete failures were noted. The remaining 23 were able to subscribe for notifications. There were 19 successful “mailto” notification tests and 4 successful “INDP” tests. At least two issues with the notification documents were identified.

The major benefit of any Bake-Off is 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.

2 Issues

There were six general issues captured during the Bake-Off and five more that were implementation specific. The general issues will be sent out on the mailing list for final disposition.

BO3-1: IPP Client failed when an unexpected HTTP “100 continue” was received. Some printers sent a “100 continue” even before the Client sent a request.

Proposed Resolution: An IPP Client must accept and handle an HTTP “100 continue” whenever it is encountered.

BO3-2: Some IPP Clients issues a zero length HTTP Post. The Client assumed that this would force a challenge if security were enabled on the Printer. The Client would have a problem if a subsequent print operation were challenged.

Proposed Resolution: The Client should use the IPP operation “validate-job” to check if a job will be accepted. This operation will cause the Printer to issue a challenge and check the print request before sending the data. The IPP Client should also be able to handle a challenge when issuing an IPP operation since there is no guarantee the connection has not been torn down.

Furthermore, a Printer should accept an empty HTTP post and issue a challenge based on the URL of the post.

Action Item: Bob Herriot: Some clients determined if a Printer requires authentication by sending an empty HTTP request. Some Printers treated this as an error. The resolution was for clients to send a ValidateJob operation and by inference to allow Printers to reject empty HTTP requests.

I raised the issue about whether a Printer should perform the authentication challenge based solely on the URL or whether it could react differently to an empty request than to a Validate-Job request.

I asked an HTTP expert and received the following information.

- 1) An HTTP server can have any policy. This means that our decision is allowable.
- 2) It is best for a client if it can associate the URL tree with the authentication space.

This means that our decision could be better. That is, we should require an IPP Printer to decide whether to issue an authentication challenge by examining the URL and nothing else, e.g. a Printer receiving a request for a particular URL, gives the same challenge to an empty request as to a Validate-Job request.

This solution allows a client to use Validate-Job to request a challenge as we decided to allow. It also allows a client to use the empty request.

The important difference between our decision and what I am proposing is that the Printer must perform an authentication challenge consistently for a URL regardless of the contents of the message body. This rule make IPP behavior consistent with good HTTP policy.

BO3-3: Do the values for “notify-uri-schemes-supported” include the ‘:’ character?

Proposed Resolution: No. See rfc2911 section 4.1.6 uri scheme data type variables

BO3-4: For get-printer-attributes, operation submitted with an unsupported “requested-attributes” value what is the return code and should an unsupported attributes group be returned containing the requested-attributes attribute and the unsupported value. There are four possibilities of status code and unsupported attribute:

- A) Successful-ok/no attributes
- B) Successful-ok/unsupported requested-attributes returned
- C) Successful-attribute-or-value-ignored/ no attributes
- D) Successful-attribute-or-value-ignored/ unsupported requested-attributes returned

The standard currently allows A, C, D. Should the standard be relaxed to include C? The implementations at the Bake-Off supported were A-11, B-1, C-3, and D-0

Proposed Resolution: Allow all combinations

BO3-5: In the subscription object is the does the mailto URL contain ‘//’. Is it <mailto://mumble> or <mailto:mumble> ?

Proposed resolution: The mailto URL does not include ‘//’.

BO3-6: Are there suffixes to “printer-state-reasons” value “none” (i.e. none-error & none-report)?

Proposed Resolution: There are no suffixes used for the value “none”.

BO3-IMP-1: Some Clients did not allow password lengths greater than eight characters. These clients should be corrected

BO3-IMP-2: Some Printers did not handle multiple operations across a single connection. IPP uses HTTP 1.1 and therefore IPP Printers must handle more than one request across a connection.

BO4-IMP-3: Some Clients did not properly decode the attribute syntax textwithlanguage. The Clients recognized that it is an implementation problem.

BO3-IMP-4: Some Printers would return all printer attributes even when only one unsupported attribute was requested. The Printers recognized this was an implementation problem.

BO3-IMP-5: Some Clients had problems accepting IPP responses that did not include the HTTP status message. Is the HTTP status message required in the IPP response?

Proposed response: IPP Client software should not parse the reason phrase per rfc2068 section 6.1.1

“The Status-Code element is a 3-digit integer result code of the attempt to understand and satisfy the request. These codes are fully defined in section 10. The Reason-Phrase is intended to give a short textual description of the Status-Code. The Status-Code is intended for use by automata and the Reason-Phrase is intended for the human user.

...

The reason phrases listed here are only recommended -- they may be replaced by local equivalents without affecting the protocol.”

Also, note that the Printer should be sending a reason phrase per rfc2068 section 6.1

“The first line of a Response message is the Status-Line, consisting of the protocol version followed by a numeric status code and its associated textual phrase, with each element separated by SP characters. No CR or LF is allowed except in the final CRLF sequence.

Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF”

3 Basic Interoperability

Key: P=Print-Job,
G=GetJobAttributes,
F=failure

v=failure due to unsupported version

V=failure corrected by client changing version

3.1 Mixed version Interoperability

	C1	C2	C3	C4	C5	C6	C7	C8	C9
P1	PG	PG	PG	PG	PG	PG	PG	PG	PG
P2	PG	PG	PG	PG	PG	PG	PG	PG	PG
P3	PG	PG	PG	PG	PG	PG	PG	PG	PG
P4	PG	PG	PG	PG	PG	PG	PG	PG	PG
P5	PG	PG	PG	PG	PG	PG	PG	PG	PG
P6	PG	PG	PG	PG	PG	PG	PG	PG	PG
P7	FFV	FFV	FFv	FF	FF	PG	PG	FF	FFv
P8	PG	PG	PG	PG	PG	PG	PG	PG	FF
P9	PG	PG	PG	PG	PG	PG	PG	FF	FF
P10	PG	PG	PG	PG	PG	PG	PG	PG	PG
P11	PG	PG	PG	PG	PG	PG	PG	PG	PG
P12	PG	PG	PG	PG	PG	PG	PG	PG	PG
P13	PG	PG	PG	PG	PG	PG	PG	PG	PG
P14	PG	PG	PG	PG	PG	PG	PG	PG	PG
P16	FFV	FFV	FFv	FFV	PG	PG	PG	FFV	
P17	PG	PG	PG	PG	PG	PG	PG	G	FG

141 out of 151 Print - 93%

145 out of 152 GetPrinterAttributes - 95%

3.2 IPP v1.1 Only

	C1	C2	C3	C4	C8	C9
P1	PG	PG	PG	PG	PG	PG
P2	PG	PG	PG	PG	PG	PG
P3	PG	PG	PG	PG	PG	PG
P4	PG	PG	PG	PG	PG	PG
P5	PG	PG	PG	PG	PG	PG
P6	PG	PG	PG	PG	PG	PG
P9	PG	PG	PG	PG	FF	FF
P10	PG	PG	PG	PG	PG	PG
P11	PG	PG	PG	PG	PG	PG
P12	PG	PG	PG	PG	PG	PG
P13	PG	PG	PG	PG	PG	PG
P14	PG	PG	PG	PG	PG	PG
P17	PG	PG	PG	PG	G	FG

74 Of 77 Print - 96%

76 of 78GetPrinterAttributes 97%

3.3 IPP v1.0 Only

	C5	C6	C7
P7	FF	PG	PG
P8	PG	PG	PG
P16	PG	PG	PG

8 Of 9 Print - 89%

8 Of 9 GetPrinterAttributes - 89%

3.4 IPP 1.0 Client/1.1 Printer

	C5	C6	C7
P1	PG	PG	PG
P2	PG	PG	PG
P3	PG	PG	PG
P4	PG	PG	PG
P5	PG	PG	PG
P6	PG	PG	PG
P9	PG	PG	PG
P10	PG	PG	PG
P11	PG	PG	PG
P12	PG	PG	PG
P13	PG	PG	PG
P14	PG	PG	PG
P16	PG	PG	PG

13 of 13 Print - 100%

13 of 13 GetPrinterAttributes - 100%

3.5 IPP 1.1 Client/1.0 Printer

	C1	C2	C3	C4	C8	C9
P7	FFV	FFV	FFv	FF	FF	FFv
P8	PG	PG	PG	PG	PG	
P16	FFV	FFV	FFv	FFV	FFV	

5 Of 16 Printer 31%, negotiated to 11 of 16 - 69%

5 Of 16 GetPrinterAttribute 31%, negotiated to 11 of 16 - 69%

4 Security

Simple print jobs were submitted between Client and Printer with the Printer requiring security.

4.1 Basic Security

	C2	C3	C4	C5	C6	C8	C9
P1							
P2	Success	Success		Success	Success	Success	Success
P4	Success	Success	Success	Success	Success	Success	Success
P5	Success	Success		Success	Fail	Success	Success
P7	Success	Success		Fail	Success	Fail	
P9	Success	Success	Success	Success	Success	Fail	
P10	Success	Success		Success	Success	Success	Success
P11	Success	Success		Success	Success	Success	Success
P12	Success	Success		Success	Success	Success	Success
P13	Success	Success		Success		Success	Success
P14	Success	Success		Success	Success	Success	Success

55 out of 59 – 93%

4.2 Digest Security

	C2	C3	C4	C8	C9
P2		Success		Fail	Success
P4		Fail	Success	Fail	Success
P5		Success	Fail	Success	
P6		Success	Success	Success	Success
P7				Fail	
P9		Success	Success	Fail	Fail
P10				Fail	Success
P11		Success	Success	Success	Success
P12	Success	Success	Success	Fail	Success
P14				Fail	

21 out of 31 – 68%

4.3 SSLv3

	C2	C6
P1	Success	Success
P2	Success	Success
P7	Success	Success
P11		Success
P13	Success	

8 out of 8 – 100%

4.4 TLS

	C2	C6
P2	Success	Success
P11	Success	Success
P13	Success	

7 out of 7 – 100%

5 Notification

Notification tested was not monitored vigorously. The notification specifications are still quite new. The notification implementations have had little chance for pair wise interoperability testing. There was limited testing time at the Bake-Off itself. As a result, the notification testing provided early implementers an opportunity to experiment with each others implementations. In the following table, empty spaces cannot be assumed untested combinations and not all failures may be listed.

5.1 Notification results matrix

Key: S=Subscription
 L=ListSubscription
 M=Mailto notification
 I=INDP notification
 FAIL=unable to interoperate

	C2	C3	C4	C7	C9
P1	SLI		SLM		SLI
P2			FAIL		SL
P3	SL				
P5	SLM		SLM		SLM
P8			FAIL		
P11	SLM	SLM	SLM	SLM	SLM
P12	SLIM	SLM	SLM	SLM	SLIM
P13	SLM	SLM	SLM	SLM	SLM

Subscription successes: 23

ListSubscription successes: 23

INDP successes: 4

Mailto successes: 19

Failures: 2

5.2 Sample Mailto Notification

This notification arrived as a multipart MIME

(Content-type: multipart/alternative;boundary=" -----_NextPart_1.1")

5.2.1 Raw text

```
-----_NextPart_1.1
Content-type: text/plain; charset="iso-2022-jp"
Content-Transfer-Encoding: 7bit
```

```

Printer: IPP Printer
Notify printer uri: http://207.16.151.112:631/printer
Notify subscribed event: job-state-changed
Job name: foobar
Job id: 2
Job state message: job-completed-successfully
Job state: completed
Job state reasons: job-completed-successfully

```

```
-----=_NextPart_1.1
```

```
Content-type: text/html;charset="iso-2022-jp"
```

```

<html><body><table>
<td>Printer<td>IPP Printer<tr>
<td>Notify printer uri<td>http://207.16.151.112:631/printer<tr>
<td>Notify subscribed event<td>job-state-changed<tr>
<td>Job name<td>foobar<tr>
<td>Job id<td>2<tr>
<td>Job state message<td>job-completed-successfully<tr>
<td>Job state<td>completed<tr>
<td>Job state reasons<td>job-completed-successfully<tr>
</table></body></html>

```

```
-----=_NextPart_1.1--
```

5.2.2 HTML View

The html appears in the mail reader like the table below

Printer	IPP Printer
Notify printer uri	http://207.16.151.112:631/printer
Notify subscribed event	job-state-changed
Job name	foobar
Job id	2
Job state message	job-completed-successfully
Job state	completed
Job state reasons	job-completed-successfully

6 Attribute Coverage

Attribute	Implemented	Attribute	Implemented
Operation Attributes			
version-number	√		
operation-id	√		
request-id	√		
attributes-charset	√		
attributes-natural-language	√		
printer-uri	√		
requesting-user-name	√		
status code	√		
Status-message	√		
Mandatory Printer Attributes			
printer-uri-supported	√	Mandatory job Attributes	
uri-security-supported	√	Job-uri	√
printer-name	√	Job-id	√
printer-state	√	Job-printer-uri	√
printer-state-reasons	√	Job-name	√
operations-supported	√	Job-originating-user-name	√
charset-configured	√	Job-state	√
charset-supported	√	job-state-reasons	√
natural-language-configured	√	Time-at-creation	√
generated-natural-language-supported	√	Time-at-processing	√
printer-is-accepting-jobs	√	Time-at-completed	√
pdl-override-supported	√	Attributes-charset	√
document-format-default	√	Attributes-natural-language	√
document-format-supported	√		
queued-job-count	√		
printer-up-time	√		
compression-supported	√		
Optional Printer Attributes			
printer-location	√	Optional Job Attributes	
printer-info	√	Job-more-info	√
printer-more-info	√	Job-state-message	√
		Job-detailed-status-message	

Attribute	Implemented	Attribute	Implemented
printer-driver-installer	√	Job-document-access-error	
printer-make-and-model	√	Number-of-documents	√
printer-more-info-manufacturer	√	Output-device-assigned	√
printer-state-message	√	Date-time-at-creation	
printer-message-from-operator	√	Date-time-at-processing	
color-supported	√	Date-time-at-completed	
reference-uri-schemes-supported	√	Number-of-intervening-jobs	√
printer-current-time	√	Job-message-from-operator	√
multiple-operation-time-out	√	Job-k-octets	√
job-k-octets-supported	√	job-impressions	√
job-impressions-supported	√	job-media-sheets	√
job-media-sheets-supported	√	Job-k-octets-processed	√
job-priority-default	√	job-impressions-completed	√
job-priority-supported	√	job-media-sheets-completed	√
job-hold-until-default	√	job-priority	√
job-hold-until-supported	√	job-hold-until	√
job-sheets-default	√	job-sheets	√
job-sheets-supported	√	multiple-document-handling	√
multiple-document-handling-default	√	copies	√
multiple-document-handling-supported	√	finishings	√
copies-default	√	page-ranges	
copies-supported	√	sides	√
finishings-default	√	number-up	√
finishings-supported	√	orientation-requested	√
page-ranges-supported	√	media	
sides-default	√	printer-resolution	
sides-supported	√	print-quality	√
number-up-default	√		
number-up-supported	√		
orientation-requested-default	√		
orientation-requested-supported	√		
media-default			

Attribute	Implemented	Attribute	Implemented
media-supported	√		
media-ready			
printer-resolution-default	√		
printer-resolution-supported	√		
print-quality-default	√		
print-quality-supported	√		

6.1 Attribute Coverage Summary

Operational Attributes: 9 out of 9 – 100%

Mandatory Printer Attributes: 17 out of 17 – 100%

Optional Printer Attributes: 39 out of 41 – 95%

Mandatory Job Attributes: 12 out of 12 – 100%

Optional Job Attributes: 22 out of 30 – 73%

Total Mandatory Attribute Coverage: 38 out of 38 – 100%

Total Optional Attribute Coverage: 61 out of 71 – 86%

Total Attribute Coverage: 99 out of 109 – 91%

7 Firewalls

Firewalls were able to:

- 1) Block all IPP traffic
- 2) Allow all IPP traffic to pass
- 3) Select specific IPP Printers to receive IPP traffic
- 4) Select specific IPP Client and Printer pairs to exchange IPP traffic
- 5) Require IPP Clients to authenticate to firewall before allowing IPP traffic to Printer
- 6) Require IPP Client to authenticate to firewall. The firewall then authenticates to the IPP Printer on the Client's behalf before allowing IPP traffic through.

In short, The Firewalls performed as expected. It is a site-specific policy decision on allowing IPP to pass through the firewall.

8 HTTP Proxies

HTTP Proxies worked when placed in between an IPP Client and Printer. The Proxies continued to allow IPP Clients and Printers to communicate even when security was used.