October 9, 2017
IPP Registration

The Printer Working Group

IPP Presets
(PRESET)

Status: Interim

Abstract: This document is a whitepaper that describes IPP Presets, a mechanism that enables a set of Job Template attribute values to be applied as a set, to provide IPP print solutions with a way to support a variety of user experience optimizations.

This document is a White Paper. For a definition of a "White Paper", see: http://ftp.pwg.org/pub/pwg/general/pwg-process30.pdf

This document is available electronically at:

https://ftp.pwg.org/pub/pwg/ipp/whitepaper/tb-ipp-preset-20171009.odt
https://ftp.pwg.org/pub/pwg/ipp/whitepaper/tb-ipp-preset-20171009.pdf

Title: IPP Presets *(PRESET)*

Table of Contents

68 **List of Figures**

69 **List of Tables**

1 Introduction

This whitepaper defines a system of new IPP attributes that allow a Printer to describe a set of one or more “presets”, which are a set of job template attributes and attribute values that are applied together as a group. Each preset set has a named label and may also have an associated “trigger”, allowing the preset to be applied implicitly in response to the User making a particular settings selection.

2 Terminology

2.1 Protocol Roles Terminology

This document defines the following protocol roles in order to specify unambiguous conformance requirements:

Client : Initiator of outgoing IPP session requests and sender of outgoing IPP operation requests (Hypertext Transfer Protocol -- HTTP/1.1 [RFC7230] User Agent).

Printer : Listener for incoming IPP session requests and receiver of incoming IPP operation requests (Hypertext Transfer Protocol -- HTTP/1.1 [RFC7230] Server) that represents one or more Physical Devices or a Logical Device.

2.2 Printing Terminology

All the printing terminology defined in IPP/1.1 Model and Semantics [RFC8011] is applicable here:

Client : Initiator of outgoing IPP session requests and sender of outgoing IPP operation requests (Hypertext Transfer Protocol (HTTP/1.1) user agent, as defined in [RFC7230]).

Document : An object created and managed by a Printer that contains description, processing, and status information. A Document object can have attached data and is bound to a single Job [PWG5100.5].

’ipp’ URI : An IPP URI as defined in [RFC3510].

’ipps’ URI : An IPP URI as defined in [RFC7472].

Job : An object created and managed by a Printer that contains description, processing, and status information. The Job also contains zero or more Document objects.

Logical Device : A print server, software service, or gateway that processes Jobs and either forwards or stores the processed Job or uses one or more Physical Devices to render output.

Output Device : A single Logical or Physical Device.

Physical Device : A hardware implementation of an endpoint device, e.g., a marking engine, a fax modem, etc.

Printer : Listener for incoming IPP session requests and receiver of incoming IPP operation requests (HTTP/1.1 server, as defined in [RFC7230]) that represents one or more Physical Devices or a Logical Device.

2.3 Other Terms Used in This Document

User : A person or automata using a Client to communicate with a Printer.

Preset : A set of attributes and attribute values that are applied all at once as job settings.

Trigger : An attribute and value whose selection causes a Preset to be selected.

2.4 Acronyms and Organizations

IANA: Internet Assigned Numbers Authority, http://www.iana.org/

IETF: Internet Engineering Task Force, http://www.ietf.org/

ISO: International Organization for Standardization, http://www.iso.org/

PWG: Printer Working Group, http://www.pwg.org/

3 Requirements for IPP Presets

3.1 Rationale for IPP Presets

There are circumstances where a number of settings are chosen as a set to achieve some common printing objective or workflow scenario. For example, the act of selecting a 4”x6” media size implies the desire to print photos. If doing so could trigger the automatic selection of an associated group of settings (change media type to glossy photo, setting the print quality to 'best'), that could have a positive user experience benefit. Sometimes these groups of settings are referred to as “presets”.

Most vendor / model-specific drivers and driver system implement support for such associations, but they do this by including logic in the driver itself. For driverless / omni-driver systems such as IPP Everywhere, some settings collections could be constructed on the Client system, but some could originate from the Printer. IPP needs to be extended to provide attributes to convey these from the Printer to a Client to support Printer-originated “presets”, to support the use cases below.

There is currently no way for the Printer to supply explicit preset information to the Client. Preset information can be configured by admin, operator, or vendor. A crude facility could be provided using Validate-Job and the “preferred-attributes” in the response, but that requires additional Client / Printer operations that are undesirable. This should be manageable locally to the Client once the settings bundles have been provided to it by the Printer.

After the application of a preset, the Client should allow a User to change individual settings. For example, if a preset includes “print-quality” of 'high' (5) and “print-color-mode” of 'color', the Client should allow the User to change the “print-quality” to 'normal' (4).

The PWG Semantic Model [PWG5105.1] defined the concept of a “job ticket template”. Saved job ticket resources are similar but not exactly the same. In particular they lack the notion of a “trigger”.

3.2 Use Cases

3.2.1 Explicit Preset Selection

Bert has found a good recipe for gazpacho on the Web, and wants to print the recipe to put it into his recipe binder. He clicks on the “Print” button in the web page. When the print dialog is presented, he selects the Preset labeled “Recipe for binder”. The “Recipe for binder” Preset specifies “2 pages per sheet” page layout, one-sided printing, trimming and punching. The Client applies the Preset to the settings in the print dialog. Bert clicks on “Print”; the Client prints the Job. Bert puts it into his recipe binder.

149 ### 3.2.2 Implicit Preset Selection

150 Kelli is in the process of printing a photo. In the print dialog, she switches the selected
151 media size from A4 to 4”x6”. Her Client has a Trigger for 4”x6” media size that names a
152 Preset named “Photos”; the “Photos” Preset includes glossy photo media type, single-
153 sided printing, and 'high' print quality. The Client acts on the Trigger by applying the
154 settings in the “Photos” Preset. Kelli is pleased that these choices were made
155 automatically by her system, saving her time and effort.

156 ### 3.2.3 Client Storing a Preset to Printer

157 Ernie has constructed his own Preset named “Better Binder Recipe”, and he would like to
158 share it with Bert. Ernie selects that Preset and taps on the “Store Preset on Printer”
159 button. The preset is uploaded to the Printer. When Bert next goes to print, he sees the
160 “Better Binder Recipe” preset that Ernie added to the Printer, and uses that for his next
161 recipe printing tasks.

162 ## 3.3 Exceptions

163 ### 3.3.1 Overriding Preset Selection

164 Bert selects the Preset labeled “Recipe for binder” in his print dialog, that selects “2 pages
165 per sheet” page layout, one-sided printing, trimming and punching. Bert decides he wants
166 to re-enable two-sided printing, and does so using the controls in the print dialog. He prints
167 the recipe and puts it into his recipe binder, pleased that he can take advantage of the
168 power of Presets but still maintain full control over a Job's settings.

169 ## 3.4 Out of Scope

170 The following are considered out of scope for this document:

171 1. The user interface for Presets
172 2. Changes to the core IPP specifications

173 ## 3.5 Design Requirements

174 The design requirements for this document are:

175 1. Define new IPP attributes that describe a Preset as a set of attributes and
176 attribute values that will be applied all at once. Each Preset is to have a unique
177 name.
178 2. Define new IPP attributes that describe a Trigger as an attribute and value and a
179 corresponding Preset name, that operates according to the principle “if Trigger
180 attribute value is chosen, then apply Preset”, to support implicit Preset selection.
181 3. Define sections to register all attributes, values, operations, and service types
182 with IANA.

4 IPP Presets Definitions

This specification defines IPP attributes and operations used for Presets and Triggers.

4.1 Printer Description Attributes

4.1.1 job-presets-supported (1setOf collection)

This REQUIRED Printer Description attribute lists named Presets that are stored on the Printer. Each collection value contains a REQUIRED “preset-name (keyword | name(MAX))” attribute and one or more Job Template attributes that are part of the Preset. The attribute names and values MUST be supported by the Printer and be listed in its Printer Description attributes. The set of attribute values MUST NOT be in conflict with one another as described by a constraint in “job-constraints-supported”.

4.1.1.1 preset-name (keyword | name(MAX))

This attribute provides a unique name for the Preset. Values can be localized using the message catalog provided at the URL specified by the “printer-strings-uri” Printer Description attribute [PWG5100.13].

4.1.1.2 Examples

Below is an example “job-presets-supported” attribute, which includes 2 collections, described using PAPI notation [PAPI]:

```
job-presets-supported={
     preset-name="draft"
     print-quality=3
},{
     preset-name="photo"
     print-content-optimize='graphics'
     print-quality=5
}
```

4.1.2 job-triggers-supported (1setOf collection)

This RECOMMENDED Printer Description attribute lists Triggers that are stored on the Printer. Each collection value contains a REQUIRED "preset-name (keyword | name(MAX))" member attribute (section 4.1.1.1) and one or more Job Template attributes that are part of the Trigger.

4.1.2.1 Examples

Here is an example “job-triggers-supported” attribute, which includes 2 collections, described using PAPI notation [PAPI]:


```
job-triggers-supported={
     preset-name="draft"
     media-col={media-type='stationery-recycled'}
},{
     preset-name="photo"
     media-col={media-type='photographic','photographic-
glossy','photographic-matte'}
}
```

In this example, if the user selects the 'stationery-recycled' media type, that will trigger the selection of the “draft” preset from “job-presets-supported”.

4.1.3 “job-presets-storage-available” (boolean)

This CONDITIONALLY REQUIRED Printer Description attribute specifies whether the Printer has resources available to store an additional Preset provided to it by a Client via a Set-Printer-Attributes operation. This attribute is REQUIRED if a Printer supports accepting Presets via a Set-Printer-Attributes operation. A Client SHOULD check this Printer Description attribute before performing a Set-Printer-Attributes operation to ensure that it will be more likely to succeed.

4.2 Storing Presets and Triggers

Presets and Triggers may be constructed by a User and stored locally on the Client. In some cases (as described in the use case in section 3.2.3), the Client may want to store those Presets and Triggers on the Printer. A Client can store a Preset or a Trigger on the Printer using the Set-Printer-Attributes operation [RFC3380].

If a Printer supports accepting new Presets and Triggers via a Set-Printer-Attributes operation, it advertises this by listing “Set-Printer-Attributes” in its “operations-supported” Printer Description attribute [RFC8011], by listing “job-presets-supported” and “job-triggers-supported” in its “printer-settable-attributes-supported” Printer Description attribute [RFC3380], and by supporting the “job-presets-storage-available” attribute (section . The Printer indicates whether it can accept and store an additional Preset by providing the value “true” for “job-presets-storage-available”.

Additionally, the Printer MUST implement the Get-Printer-Supported-Values operation [RFC3380] to allow a Client to request the values that the Printer allows in the Set-Printer-Attributes operation for the “job-presets- supported" and “job-triggers-supported” attributes.

As per section 4.1.3, a Client SHOULD check the state of the “job-presets-storage-available” attribute to determine whether the Printer will accept a new Preset. If the Printer rejects the Preset because it lacks the resources to store it or because it has unsupported attributes or values, then the Printer MUST include the “client-error-bad-request” status code in its Set-Printer-Attributes operation response.

5 Client Implementation Recommendations

5.1 Presets

A Client should list available Presets by name in some manner in its UI presenting printing choices. The Presets may come from the Printer or they may be created by the Client and persisted in some way. When a User selects a Preset, the print settings in that Preset should be applied. Implementors of Clients may want to consider what to do when a setting has been changed by the user and then a Preset has been selected that might change that setting. The Client might notify the User that the setting will be changed, or alternately might apply the Preset but not change the setting changed by the User.

5.2 Triggers

The semantic expectation of a Trigger is “IF setting value is chosen, THEN apply Preset”. Upon detecting that a Trigger's setting value has been chosen by the User, the Client applies the Preset. Client implementors may want to consider cases where Triggers are disabled, such as following manual selection by a user, or perhaps only allowing one Trigger per “print dialog session” to be used.

A Trigger should only be applied in response to user input, and not in response to a value being set by another Preset, a constraint, or some other automatic selection implemented by the Client.

6 Conformance Requirements

6.1 Conformance Requirements for Clients

In order for a Client to claim conformance to this specification, a Client MUST support:

1. The IPP Printer attributes defined in section 4.1;
2. The internationalization considerations in section 7;
3. The security considerations in section 8.

6.2 Conformance Requirements for Printers

In order for a Printer to claim conformance to this specification, a Printer MUST support:

1. The IPP Printer attributes defined in section 4.1;
2. The internationalization considerations in section 7;
3. The security considerations in section 8.

7 Internationalization Considerations

For interoperability and basic support for multiple languages, conforming implementations MUST support the Universal Character Set (UCS) Transformation Format -- 8 bit (UTF-8) [RFC3629] encoding of Unicode [UNICODE] [ISO10646] and the Unicode Format for Network Interchange [RFC5198].

Implementations of this specification SHOULD conform to the following standards on processing of human-readable Unicode text strings, see:

- Unicode Bidirectional Algorithm [UAX9] – left-to-right, right-to-left, and vertical
- Unicode Line Breaking Algorithm [UAX14] – character classes and wrapping
- Unicode Normalization Forms [UAX15] – especially NFC for [RFC5198]
- Unicode Text Segmentation [UAX29] – grapheme clusters, words, sentences
- Unicode Identifier and Pattern Syntax [UAX31] – identifier use and normalization
- Unicode Collation Algorithm [UTS10] – sorting
- Unicode Locale Data Markup Language [UTS35] – locale databases

Implementations of this specification are advised to also review the following informational documents on processing of human-readable Unicode text strings:

- Unicode Character Encoding Model [UTR17] – multi-layer character model
- Unicode in XML and other Markup Languages [UTR20] – XML usage
- Unicode Character Property Model [UTR23] – character properties
- Unicode Conformance Model [UTR33] – Unicode conformance basis

8 Security Considerations

The IPP extensions defined in this document require the same security considerations as defined in the IPP/1.1: Model and Semantics [RFC8011] plus additional security considerations below .

8.1 Human-readable Strings

Implementations of this specification SHOULD conform to the following standard on processing of human-readable Unicode text strings, see:

- Unicode Security Mechanisms [UTS39] – detecting and avoiding security attacks

Implementations of this specification are advised to also review the following informational document on processing of human-readable Unicode text strings:

- Unicode Security FAQ [UNISECFAQ] – common Unicode security issues

9 IANA and PWG Considerations

9.1 Attribute Registrations

The attributes defined in this document will be published by IANA according to the procedures in IPP Model and Semantics [RFC8011] section 6.2 in the following file:

http://www.iana.org/assignments/ipp-registrations

The registry entries will contain the following information:

Printer Description attributes:	Reference
job-presets-supported (1setOf collection)	[5100.PRESET]
preset-name (keyword | name(MAX))	[5100.PRESET]
job-triggers-supported (1setOf collection)	[5100.PRESET]
preset-name (keyword | name(MAX))	[5100.PRESET]
job-presets-storage-available (boolean)	[5100.PRESET]

10 References

10.1 Normative References

[ISO10646] "Information technology -- Universal Coded Character Set (UCS)", ISO/IEC 10646:2011

[PWG5100.5] D. Carney, T. Hastings, P. Zehler. “Internet Printing Protocol (IPP): Document Object”, PWG 5100.5-2003, October 2003, http://ftp.pwg.org/pub/pwg/candidates/cs-ippdocobject10-20031031-5100.5.pdf

[PWG5100.12] R. Bergman, H. Lewis, I. McDonald, M. Sweet, "IPP Version 2.0, 2.1, and 2.2", PWG 5100.12-2015, October 2015, http://ftp.pwg.org/pub/pwg/standards/std-ipp20-20151030-5100.12.pdf

[PWG5100.13] M. Sweet, I. McDonald, P. Zehler, "IPP: Job and Printer Extensions - Set 3 (JPS3)", PWG 5100.13-2012, July 2012,

http://ftp.pwg.org/pub/pwg/candidates/cs-ippjobprinterext3v10-20120727-5100.13.pdf

[PWG5100.19] S. Kennedy, "IPP Implementor's Guide v2.0", PWG 5100.19-2015, August 2015, http://ftp.pwg.org/pub/pwg/candidates/cs-ippig20-20150821-5100.19.pdf

[RFC2817] R. Khare, S. Lawrence, “Upgrading to TLS Within HTTP/1.1”, RFC 2817, May 2000, https://www.ietf.org/rfc/rfc2817.txt

[RFC3510] R. Herriot, I. McDonald, “Internet Printing Protocol/1.1: IPP URL Scheme”, RFC 3510, April 2003, https://tools.ietf.org/html/rfc3510

[RFC3629] F. Yergeau, “UTF-8, a transformation format of ISO 10646”, RFC 3629, November 2003, https://www.ietf.org/rfc/rfc3629.txt

[RFC3380] T. Hastings, R. Herriot, C. Kugler, H. Lewis, "Internet Printing Protocol (IPP): Job and Printer Set Operations", RFC 3380, September 2002, http://tools.ietf.org/html/rfc3380

[RFC5198] J. Klensin, M. Padlipsky, “Unicode Format for Network Interchange”, RFC 5198, March 2008, https://www.ietf.org/rfc/rfc5198.txt

[RFC7230] R. Fielding, J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, June 2014, http://www.ietf.org/rfc/rfc7230.txt

[RFC7472] I. McDonald, M. Sweet, “Internet Printing Protocol (IPP) over HTTPS Transport Binding and the 'ipps' URI Scheme”, RFC 7472, March 2015, https://tools.ietf.org/html/rfc7472

[RFC8010] M. Sweet, I. McDonald, “Internet Printing Protocol/1.1: Encoding and Transport”, RFC 8010, January 2017, https://www.ietf.org/rfc/rfc8010.txt

[RFC8011] M. Sweet, I. McDonald, “Internet Printing Protocol/1.1: Model and Semantics”, RFC 8011, January 2017, https://www.ietf.org/rfc/rfc8011.txt

[UAX9] Unicode Consortium, “Unicode Bidirectional Algorithm”, UAX#9, May 2016, http://www.unicode.org/reports/tr9

[UAX14] Unicode Consortium, “Unicode Line Breaking Algorithm”, UAX#14, June 2016, http://www.unicode.org/reports/tr14

[UAX15] Unicode Consortium, “Normalization Forms”, UAX#15, February 2016, http://www.unicode.org/reports/tr15

[UAX29] Unicode Consortium, “Unicode Text Segmentation”, UAX#29, June 2016, http://www.unicode.org/reports/tr29

[UAX31] Unicode Consortium, “Unicode Identifier and Pattern Syntax”, UAX#31, May 2016, http://www.unicode.org/reports/tr31

[UNICODE] The Unicode Consortium, “Unicode® 10.0.0”, June 2017, http://unicode.org/versions/Unicode10.0.0/

[UTS10] Unicode Consortium, “Unicode Collation Algorithm”, UTS#10, May 2016, http://www.unicode.org/reports/tr10

[UTS35] Unicode Consortium, “Unicode Locale Data Markup Language”, UTS#35, October 2016, http://www.unicode.org/reports/tr35

[UTS39] Unicode Consortium, “Unicode Security Mechanisms”, UTS#39, June 2016, http://www.unicode.org/reports/tr39

10.2 Informative References

[PAPI] A. Hlava, N. Jacobs, M. Sweet, "Open Standard Print API (PAPI)", July 2005, http://prdownloads.sourceforge.net/openprinting/PAPI-specification.pdf?download

[PWG5105.1] P. Zehler, T. Hastings, S. Albright, “Semantic Model v1.0”, PWG 5105.1-2004, January 2004, http://ftp.pwg.org/pub/pwg/candidates/cs-sm10-20040120-5105.1.pdf

[UNISECFAQ] Unicode Consortium “Unicode Security FAQ”, November2016, http://www.unicode.org/faq/security.html

[UTR17] Unicode Consortium “Unicode Character Encoding Model”, UTR#17, November 2008, http://www.unicode.org/reports/tr17

[UTR20] Unicode Consortium “Unicode in XML and other Markup Languages”, UTR#20, January 2013, http://www.unicode.org/reports/tr20

[UTR23] Unicode Consortium “Unicode Character Property Model”, UTR#23, May 2015, http://www.unicode.org/reports/tr23

[UTR33] Unicode Consortium “Unicode Conformance Model”, UTR#33, November 2008, http://www.unicode.org/reports/tr33

11 Authors' Addresses

Primary authors:

Smith Kennedy
HP Inc.
11311 Chinden Blvd.
Boise, Idaho 83714
smith.kennedy@hp.com

The authors would also like to thank the following individuals for their contributions to this standard:

Ira McDonald – High North
Mike Sweet – Apple Inc.
Eren Rodriguez – HP Inc.

12 Change History

12.1 October 9, 2017

Updated to as per HP feedback:

- Added the “job-presets-storage-available” attribute definition and semantics
- Added additional Client considerations and Printer behavior for error conditions when the submitted Preset contains unsupported values or lacks resources to store the Preset it received

12.2 September 12, 2017

Updated as per feedback from August 2017 PWG vF2F session and subsequent discussion on IPP reflector:

- Extensively updated structure of section 4 “IPP Presets Definitions”
  - Added section 4.2 to discuss storing presets using Set-Printer-Attributes
  - Added and then removed section 4.3 (placeholder) to discuss storing presets as resources, because it was decided in an ipp@pwg.org reflector discussion that this was not the way we wanted to go.
- Added “Client Implementation Recommendations” section
- Added “Conformance Requirements” section
- Added “IANA and PWG Considerations” section

12.3 August 7, 2017

Minor clarifications and editorial changes to section 3.

12.4 July 28, 2017

Updated following IPP WG review and feedback:

- Added Printing Terminology by copy / paste from RFC 8011 section 2.2
- Incorporated Internationalization and Security Considerations content from IPP System

- Added and fixed many references
- Refactored section 4 according to the meeting minutes to include PAPI examples to better illustrate the structure, which is difficult to articulate using conventional IPP syntax (since there isn't a formal “data type” for “any attribute”

Other additions and changes:

- Added a new use case “Client Saving Preset Settings to Printer” to explore how that might be supported in IPP, and if that requires additional definitions.

12.5 June 9, 2017

Updated and refactored following May 11 IPP WG teleconference

- Expanded use case descriptions
- Refactored IPP attribute definitions

12.6 April 18, 2017

Initial revision.

