INTERNET-DRAFT There is 1 issue highlighted like this
<draft-ietf-ipp-collection-00.txt>

Roger deBry
Utah Valley State College~~IBM Printing Company~~
T. Hastings
Xerox Corporation
R. Herriot
Xerox Corporation

February 22~~1~~, 2000~~December 8, 1999~~

Internet Printing Protocol~~/1.1~~: The 'collection' attribute syntax

Status of this Memo:

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of [RFC2026]. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress".

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed as http://www.ietf.org/shadow.html.

Abstract

This document specifies an OPTIONAL attribute syntax called 'collection' for use with the Internet Printing Protocol/1.0 (IPP) [RFC2565, RFC2566], ~~and~~ IPP/1.1 [ipp-mod, ipp-pro], and subsequent versions. A 'collection' is a container holding one or more named values, which are called "member" attributes. A collection allows data to be grouped like a ~~C struct~~PostScript dictionary or a Java Map.

The full set of IPP documents includes:

- Design Goals for an Internet Printing Protocol [RFC2567]
- Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [RFC2568]
- Internet Printing Protocol/1.1: Model and Semantics (this document)
- Internet Printing Protocol/1.1: Encoding and Transport [IPP-PRO]
- Internet Printing Protocol/1.1: Implementer's Guide [IPP-IIG]
- Mapping between LPD and IPP Protocols [RFC2569]

The "Design Goals for an Internet Printing Protocol" document takes a broad look at distributed printing functionality, and it enumerates real-life scenarios that help to clarify the features that need to be included in a printing protocol for the Internet. It identifies requirements for three types of users: end users, operators, and administrators. It calls out a subset of end user requirements that are satisfied in IPP/1.0. A few OPTIONAL operator operations have been added to IPP/1.1.

The "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol" document describes IPP from a high level view, defines a roadmap for the various documents that form the suite of IPP specification documents, and gives background and rationale for the IETF working group's major decisions.

The "Internet Printing Protocol/1.1: Encoding and Transport" document is a formal mapping of the abstract operations and attributes defined in the model document onto HTTP/1.1 [RFC2616]. It defines the encoding rules for a new Internet MIME media type called "application/ipp". This document also defines the rules for transporting over HTTP a message body whose Content-Type is "application/ipp". This document defines a new scheme named 'ipp' for identifying IPP printers and jobs.

The "Internet Printing Protocol/1.1: Implementer's Guide" document gives insight and advice to implementers of IPP clients and IPP objects. It is intended to help them understand IPP/1.1 and some of the considerations that may assist them in the design of their client and/or IPP object implementations. For example, a typical order of processing requests is given, including error checking. Motivation for some of the specification decisions is also included.

The "Mapping between LPD and IPP Protocols" document gives some advice to implementers of gateways between IPP and LPD (Line Printer Daemon) implementations.

Table of Contents

1 Problem Statement

IPP supports most of the common data structures that are available in programming languages. It lacks a mechanism for grouping several values of different types. The ~~C~~ Java language uses the ~~struct~~ Map to solve this problem and PostScript has a dictionary.

2 Solution

The IPP 'collection' is a container holding one or more named values (i.e. attributes), which are called member attributes. A collection also has a type name, which identifies the ~~allowed~~ expected member attributes, as does ~~the name of a C struct~~a subclass of a ~~or~~ Java ~~class~~Map. A collection value is similar to a group, such as an operation group. They both consist of a ~~series~~ set of attributes.

The name of each member attribute MUST be unique within a collection, but MAY be the same as the name of a member attribute in another collection type and/or MAY be the same as the name of an attribute that is not a member of a collection.

A client or Printer is said to "recognize" collections as a single attribute value if it can determine the beginning and end of a collection value and if it can distinguish attributes within the collection from attributes outside of the collection. In order to support legacy IPP implementations, a client MUST indicate that it "recognizes" collections by including the operation attribute "collection-syntax-recognized" with the value of 'true' in each request. A printer MUST indicate that it "recognizes" collections by supporting the attribute "collection-syntax-recognized" with the value of 'true'. ~~the name of a member attribute MUST be different from any attribute in an operation or object unless its semantics are identical to those in the operation or object.~~

The fact that a Printer recognizes collections does not require the printer to support collection values of attributes that are defined to have values of collections and other attribute syntaxes. For example, if an attribute is defined to have the attribute syntax: (type3 keyword | name | collection), a Printer that recognizes collections MAY support only keyword values of such an attribute. ~~Note: it is not possible to distinguish between two Printers that both recognize collections but one ignores all collection values and the other ignores the collection a client sends it even though it processes some collection values.~~

Each member attribute can have any attribute syntax type, including 'collection', and can be either single-valued or multi-valued. The length of a collection value is not limited. However, the length of each member attribute MUST NOT exceed the limit of its attribute syntax.

The member attributes in a collection can be in any order. When a client sends the Printer a collection, the order that the Printer stores the value and the order returned in a response MAY be different from the order sent by the client.

~~Note: I~~if a collection contains two or more member attributes with the same attribute name, the collection is not well formed. The receiver of such a collection ~~can~~ MAY either treat the collection as a bad value or ignore all but one of the identically named member attributes.

~~ISSUE 01: if an attribute is multi-valued, it may be useful for multiple occurrences of an attribute in a collection or a group to be equivalent to a single occurrence with the set of all values of the multiple occurrences. This rule might be useful for machine generated attributes where no state is kept.~~

3 Definition of a Collection Type

When a specification defines an attribute "xxx" whose syntax type is 'collection' or '1setOf collection', it must define following aspects of the ~~collection~~attribute.

1. The name of the attribute "xxx"
2. Its syntax type, which includes a collection syntax-type
3. Its default-value is specified by
   a) the attribute's definition
   b) an attribute, such as "xxx-default", which may have a collection value
4. Its supported values, which may be specified by one of:
   a) the attribute's definition
   b) a boolean attribute, such as "xxx-supported", which is true if the attribute is supported. The supported values are specified by the attribute's definition which specifies the supported values for each member of a collection or the "yyy-supported" that specifies the value supported for the "yyy" member attribute.
   c) an attribute, such as "xxx-supported", which contains the explicit collection values and other values supported.

~~1.~~5. the name of the collection type, whose characters are the same as those for a keyword.

~~2.~~6. the following information about each "yyy" member attribute ~~"yyy"~~:

a) its name, which is a keyword like all attributes. It must be unique within the collection type. ~~It must also be unique with respect to operation and object attributes unless its semantics are identical to those in the operation or object.~~

b) its syntax type, which may be any IPP syntax type, includinge 'collection'. If the attribute syntax type starts with "1setOf", the member attribute is multi-valued.

c) its ~~allowed~~ supported values, either enumerated explicitly or specified by the values of a referenced attribute which may be specified by either:~~.~~

– the attribute's definition

– an attribute, such as "yyy-supported", which contains the explicit values supported. The "yyy-supported" attribute is a Printer attribute and not in a collection. For example, if a collection contains the attribute "media" attribute and its supported values are specified by the attribute "media-supported" attribute, the "media-supported" attribute is the same Printer attribute that the "media" attribute uses.

d) whether "yyy"it MUST be or MAY be supplied by a client in a request.

e) its the default value of "yyy" if it is OPTIONAL for a client MAY to supply the "yyy" attribute in a requestit. The default value is specified by can be stated explicitly or can come from a specified attribute either:.

– the attribute's definition

– an attribute, such as "yyy-default", which may have a collection value

f) whether "yyy" it MUST be or MAY be supported by the printer.

g) its Tthe semantics of "yyy".

4 Order of Member Attributes

The member attributes of a collection value are unordered. A Printer and a client MUST accept member attributes of a collection in any order. NoteTherefore, a Printer and a client mayMAY send the member attributes of a collection value in any order. A Printer NEED NOT return member attributes to a client in the order received from a client.

5 New Operation Attribute

5.1 collection-syntax-recognized (boolean)

A client MUST include this operation attribute with a value of 'true' in each request if it recognizes the collection-syntax. If a client does not include this operation attribute or its value is not 'true' in a request, then a Printer MUST NOT sendreturn a collection in a response.

ISSUE 01: If a Printer creates a notification subscription [ipp-ntfy] with a request that does not include "collection-syntax-recognized" (boolean) operation attribute with a value of 'true', then a Printer MUST NOT send a collection in a Notification to a Notification Recipient?

56 New Printer Attribute

5.16.1 collection-syntax-recognized (boolean)

A Printer MUST support this attribute with a value of 'true' if it recognizes the collection-syntax. If a Printer does not support this attribute or its value is not 'true', then a client MUST NOT send a collection in a request.

67 New Out-of-band value

6.17.1 'none'

'none'	The specified Job Template attribute in the request MUST NOT be applied to the job. Specifically, this value overrides the Printer's "xxx-default" attribute value for the Job Template attribute, if one exists.

This "out-of-band" value allows a client to specify "turn-off" a feature that is specified by an attribute whose value is a collection. Because a client specifies a value, the Printer uses the client-specified value and not the Printer's default value.

If a Printer supports the use of the 'collection' attribute syntax for an attribute, a Printer MUST support the use of the "out-of-band" value 'none'.

A Printer MUST support the "out-of-band" value 'none' as the value for an attribute "xxx" if:

- the definition of the attribute specifies 'none' MUST be supported AND
- the definition of the attribute specifies 'none' MAY be supported and it is a value of the attribute "xxx-supported".

48 Unsupported Values

The rules for returning an unsupported collection attribute are an extension to the current rules.

If a collection contains unrecognized, unsupported member attributes and/or conflicting values, the attribute returned in the Unsupported Group is a collection containing the unrecognized, unsupported member attributes, and/or conflicting values. The unrecognized member attributes have an out-of-band value of 'unsupported' (see the beginning of [ipp-mod] section 4.1). The unsupported member attributes and conflicting values have their unsupported values.

59 Encoding

This section defines the encoding of a collection syntax type. A collection is encoded by using three new tags:

Tag name	Tag value	Meaning
beginCollection	0x34	Begin the named collection.
endCollection	0x37	End the named collection.
~~sepCollection~~	~~0x38~~	~~Separate two collections of a multi-valued attribute~~

A collection value is encoded as a sequence of attribute values preceded by a beginCollection value and followed by an endCollection value. The value field of a beginCollection and an endCollection both contain the name of the collection type, which is a string of ASCII characters. These values allow a receiver to optionally match an endCollection value with a beginCollection. A 1setOf collection is encoded using the rules for 1setOf and collection~~, except that adjacent endCollection and beginCollection values MUST be combined into a single sepCollection value. Its value field contains the collection type. In a 1setOf collection, the endCollection value marks the end of last collection in the 1setOf collection. For legacy reasons, the~~ The name field for the endCollection ~~and sepCollection~~ must be ~~non-~~empty. ~~The name is arbitrarily assigned to be "c".~~

The following example is written in the style of the IPP/1.1 "Encoding and Transport" document [ipp-pro]. The following example is for a job-notify attribute containing a set of 2 collections.

Octets	**Symbolic Value**	**Protocol field**	**comments**
0x34	beginCollection	value-tag	Beginning of the collection
0x000a		name-length	
job-notify	job-notify	Name	
0x000f		Value-length	
job-notify-coll	job-notify-coll	Value	Collection type
0x45	uri type	value-tag	"notify-recipients" attribute
0x0010		name-length	
notify-recipient	notify-recipient	Name	
0x0013		value-length	
ipp-notify:port=700		Value	
0x44	keyword type	value-tag	"notify-event-groups" attribute
0x000d		name-length	
notify-events		Name	
0x0d		value-length	
job-completed		Value	
0x44	keyword type	value-tag	2nd "notify-event-groups" attribute
0x0000		name-length	0 length means next multiple value
0x0011		value-length	
job-state-changed	job-completion	Value	
0x37	endCollection	value-tag	
0x0000		name-length	
0x000f		value-length	
job-notify-coll		Value	Matches value of beginCollection

Octets	Symbolic Value	Protocol field	comments
0x34	beginCollection	value-tag	Separator between collection values
0x0001		name-length	
0x000f		value-length	
job-notify-coll		Value	Matches value of beginCollection
0x45	uri type	value-tag	"notify-recipients" attribute
0x0010		name-length	
notify-recipient		Name	
0x0014		value-length	
mailto:smith@foo.com		Value	
0x44	keyword type	value-tag	"notify-event-groups" attribute
0x000d		name-length	
notify-events		Name	
0x0d		value-length	
job-completed		Value	
0x37	endCollection	value-tag	End of last collection
0x0001		name-length	
0x000f		value-length	
job-notify-coll		Value	Matches value of beginCollection

610 Legacy issues

If a client supportsrecognizes collections in responses, it MUST include the operation attribute "collection-syntax-recognized" operation attribute with the value of 'true' in each operation whether or not the request contains a collection.

If a Printer supportsrecognizes collections in requests, it MUST support the attribute "collection-syntax-recognized" Printer Description attribute with the value of 'true'.

A client that supports collections MUST NOT send collections in a request to a Printer that does not supportrecognize collections.

A Printer that supports collections MUST NOT sendreturn collections in a response to a client that does not supportrecognize collections.

Although a client or Printer that doesn't supportrecognize collections will skip over the beginCollection and endCollection tags as unrecognized syntax types, the client or Printer will mistakenly assume that the member attributes are outside of the unrecognized collection. Thus it is important that clients and Printers that don't supportrecognize collections not receive them.~~The encoding has been designed to work with IPP/1.0 and IPP/1.1 implementations. An IPP/1.0 or IPP/1.1 receiver will treat the three new syntax types, beginCollection, endCollection and sepCollection as unrecognized syntax types. A legacy implementation is expected to behave as follows.~~

~~A beginCollection value appears to be an attribute with an unsupported value.~~

The member attributes that follow the beginCollection appear to be normal attributes within their group (e.g. normal for the operation attributes group). If an attribute has the same name as an attribute allowed in the group, it as a recognized member of the group (e.g. as a normal operation attribute).

An endCollection value appears to be an attribute with an unsupported value and unrecognized name "c". The same is true for a sepCollection value.

711 IANA Considerations

This attribute syntax will be registered with IANA after the WG approves its specification according to the procedures for extension of the IPP/1.1 Model and Semantics [ipp-mod] and after IPP becomes a proposed IETF standard.

812 Internationalization Considerations

This attribute syntax by itself has no impact on internationalization. However, the member attributes that are subsequently defined for use in a collection may have internationalization considerations, as may any attribute, according to [ipp-mod].

913 Security Considerations

This attribute syntax causes no more security concerns than any other attribute syntax. It is only the attributes that are subsequently defined to use this or any other attribute syntax that may have security concerns, depending on the semantics of the attribute, according to [ipp-mod].

1014 References

[ipp-mod]
Isaacson, S., deBry, R., Hastings, T., Herriot, R., Powell, P., "Internet Printing Protocol/1.1: Model and Semantics" draft-ietf-ipp-model-v11-04.txt, June 23, 1999.

[ipp-ntfyot]
Isaacson, S., Martin, J., deBry, R., Hastings, T., Shepherd, M., Bergman, R. " Internet Printing Protocol/1.0 & 1.1: IPP Event Notification Specification" draft-ietf-ipp-not-spec-021.txtdoc, work in progress, October 10, 1999February 2, 2000.

[ipp-pro]
Herriot, R., Butler, S., Moore, P., Turner, R., "Internet Printing Protocol/1.1: Encoding and Transport", draft-ietf-ipp-protocol-v11-03.txt, June 23, 1999.

[ISO-10175]
ISO/IEC 10175 Document Printing Application (DPA), June 1996.

[RFC2565]
Herriot, R., Butler, S., Moore, P., Tuner, R., "Internet Printing Protocol/1.0: Encoding and Transport", RFC 2565, April 1999.

[RFC2566]
R. deBry, T. Hastings, R. Herriot, S. Isaacson, P. Powell, "Internet Printing Protocol/1.0: Model and Semantics", RFC 2566, April 1999.

[RFC2567]
Wright, D., "Design Goals for an Internet Printing Protocol", RFC 2567, April 1999.

[RFC2568]
Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol", RFC 2568, April 1999.

[RFC2569]
Herriot, R., Hastings, T., Jacobs, N., Martin, J., "Mapping between LPD and IPP Protocols", RFC 2569, April 1999.

[RFC2616]
R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.

15 Author's Addresses

Tom Hastings
Xerox Corporation
737 Hawaii St. ESAE 231
El Segundo, CA 90245

Phone: 310-333-6413
Fax: 310-333-5514
e-mail: hastings@cp10.es.xerox.com

Robert Herriot
Xerox Corp.
3400 Hill View Ave, Building 1
Palo Alto, CA 94304

Phone: 650-813-7696
Fax: 650-813-6860
e-mail: robert.herriot@pahv.xerox.com

Roger deBry
Utah Valley State College
Orem, UT 84058

Phone: (801) 222-8000
EMail: debryro@uvsc.edu

1216 APPENDIX A: Example of collection usage

This section describes one collection Job Template example.

1.116.1 "job-notify" Operation attribute

The following example illustrates the definition of a collection attribute for the "job-notify" operation attribute (see [ipp-ntfy]). Each column of the table corresponds to information that is required for member attributes. Only the semantics have been omitted.

1.collection type: "job-notify-coll"

members of the collection

Member name	Member type	Supported-values	Client supplied/ default	Printer support
notify-recipient	uri	notify-recipient-schemes-supported	MUST	MUST
notify-events	1setOf type2 keyword	notify-events-supported	notify-events-default	MUST
subscriber-user-data	octetString(63)	<any octet string>	<empty octetString>	MUST
notify-attributes-charset	charset	charset-supported	attributes-charset in operation group	MAY
notify-attributes-natural-language	naturalLanguage	generated-natural-language-supported	attributes-natural-language in operation group	MAY

Note: for the "client supplied/default" column, the default is specified if it is OPTIONAL for the client MAY to supply the member attribute in a requestit.

1317 Appendix BA: Full Copyright Statement