PWG DRAFT
File: ipp-collection-attr-syntax-990909.doc

Roger deBry
IBM Printing Company
T. Hastings
Xerox Corporation
R. Herriot
Xerox Corporation
September 9, 1999

Internet Printing Protocol/1.1:
attribute syntax for Collection

Abstract

This document specifies an OPTIONAL attribute syntax called 'collection'. 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.

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 language uses the struct to solve this problem.

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 member attributes, as does the name of a C struct or Java class. A collection value is similar to a group, such as an operation group. They both consist of a series 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. In order to support legacy IPP implementations, 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.

Each member attribute can have any 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.

Note: 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 either treat the collection as a bad value or ignore all but one of the identically named members.

3 Definition of a collection type

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

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

2. the following information about each member attribute:

   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, include collection. If the syntax type starts with “1setOf”, the member attribute is multi-valued.

   c) its allowed values, either enumerated explicitly or specified by the values of a referenced attribute.

   d) whether it MUST be or MAY be supplied by a client.

   e) its default value if a client MAY supply it. The default value can be stated explicitly or can come from a specified attribute.

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

g) its semantics

4 Unsupported Values

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

1. If a collection contains unrecognized, unsupported member attributes and/or conflicting value, 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. The unsupported member attributes and conflicting values have their unsupported values.

5 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 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

Octets	**Symbolic Value**	**Protocol field**	**comments**
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	
0x38	sepCollection	value-tag	Separator between collection values
0x0001		name-length	
c		Name	Non-empty for legacy
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	
c		Name	Non-empty for legacy
0x000f		value-length	
job-notify-coll		Value	Matches value of beginCollection

6 Legacy issues

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.

7 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.

8 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.

9 Security Considerations

This attribute syntax causes no more security concerns than any 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.

10 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-v11-03.txt, June, 1999.

[ipp-not]
Isaacson, S., Martin, J., deBry, R., Hastings, T., " Internet Printing Protocol/1.0 & 1.1: IPP Event Notification Specification” draft-ietf-ipp-not-spec-00.doc, work in progress, August 25, 1999.

[ipp-pro]
Herriot, R., Butler, S., Moore, P., Turner, R., "Internet Printing Protocol/1.1: Encoding and Transport", draft-ietf-ipp-protocol-v11-02.txt, June, 1999.

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

11 APPENDIX A: Example of collection usage

This section describes one collection Job Template example.

11.1 "job-notify" Operation attribute

The following example illustrates the definition of a collection attribute for the “job-notify” operation attribute. 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”

2. 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 the client MAY supply it.