
***PrintBasic:1* Service Template Version 1.01**

For UPnP™ Version 1.0

Status: *Approved Standard*

Date: August 8, 2002

6

7 This Standardized DCP has been adopted as a Standardized DCP by the Steering Committee of
 8 the UPnP™ Forum, pursuant to Section 2.1(c)(ii) of the UPnP™ Forum Membership
 9 Agreement. UPnP™ Forum Members have rights and licenses defined by Section 3 of the
 10 UPnP™ Forum Membership Agreement to use and reproduce the Standardized DCP in UPnP™
 11 Compliant Devices. All such use is subject to all of the provisions of the UPnP™ Forum
 12 Membership Agreement.

13 THE UPNP™ FORUM TAKES NO POSITION AS TO WHETHER ANY INTELLECTUAL
 14 PROPERTY RIGHTS EXIST IN THE STANDARDIZED DCPS. THE STANDARDIZED
 15 DCPS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". THE UPNP™ FORUM
 16 MAKES NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH
 17 RESPECT TO THE STANDARDIZED DCPS, INCLUDING BUT NOT LIMITED TO ALL
 18 IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND
 19 FITNESS FOR A PARTICULAR PURPOSE, OF REASONABLE CARE OR
 20 WORKMANLIKE EFFORT, OR RESULTS OR OF LACK OF NEGLIGENCE.

21 © 2002 Contributing Members of the UPnP™ Forum. All Rights Reserved.

22

Authors	Company
Shivaun Albright	Hewlett-Packard
Tom Hastings	Xerox
Harry Lewis	IBM
Paul Moore	Netreon
Peter Zehler	Xerox
Gerrie Shults	Hewlett-Packard

23

24 **Contents**

25	1. OVERVIEW AND SCOPE.....	5
26	2. SERVICE MODELING DEFINITIONS	5
27	2.1. SERVICE TYPE	5
28	2.2. TERMINOLOGY	5
29	2.2.1. Conformance Terminology	5
30	2.2.2. Other terminology.....	6
31	2.2.3. Notation: use of quotation marks	6
32	2.3. REFERENCES.....	6
33	2.4. INTENT OF A PRINT JOB	7
34	2.4.1. Production vs. Layout Job Attributes	7
35	2.4.2. Precedence of Production vs. Layout Job Attributes.....	8
36	2.5. STATE VARIABLES.....	8
37	2.5.1. Derived data types.....	8
38	2.6. SERVICE STATE TABLE.....	10
39	2.6.1. The Printer's supported and default values.....	10
40	2.6.2. The Distinguished Value used to avoid action override of PDL	10
41	2.6.3. Purposes of the SST State Variables.....	11
42	2.6.4. PrinterName	12
43	2.6.5. PrinterLocation	13
44	2.6.6. DeviceId.....	13
45	2.6.7. PrinterState.....	13
46	2.6.8. PrinterStateReasons	14
47	2.6.9. XHTMLImageSupported.....	15
48	2.6.10. ColorSupported	16
49	2.6.11. JobIdList.....	16
50	2.6.12. JobId.....	16
51	2.6.13. JobEndState	17
52	2.6.14. JobName	17
53	2.6.15. JobOriginatingUserName.....	17
54	2.6.16. DocumentFormat.....	17
55	2.6.17. Copies	18
56	2.6.18. Sides.....	18
57	2.6.19. NumberUp	19
58	2.6.20. OrientationRequested	19
59	2.6.21. MediaSize.....	20
60	2.6.22. MediaType	21
61	2.6.23. PrintQuality.....	22
62	2.6.24. DataSink	23
63	2.6.25. JobMediaSheetsCompleted.....	23
64	2.7. EVENTING AND MODERATION	23
65	2.7.1. Event Model.....	23
66	2.7.2. Synchronization of Evented Variables.....	24
67	2.8. ACTIONS.....	27
68	2.8.1. CreateJob.....	27
69	2.8.2. CancelJob	29
70	2.8.3. GetPrinterAttributes	29
71	2.8.4. GetJobAttributes.....	30
72	2.8.5. HTTP Post	31
73	2.8.6. Non-Standard Actions Implemented by a UPnP Vendor.....	32
74	2.8.7. Common Error Codes.....	32

75 2.9. THEORY OF OPERATION.....32
 76 2.9.1. Jobs.....33
 77 2.9.2. Actions33
 78 2.9.3. Events.....34
 79 2.9.4. Security34
 80 2.9.5. Localization34
 81 2.9.6. IPP Data Type mapping to UPnP Data Types34
 82 3. XML SERVICE DESCRIPTION36
 83

84 **List of Tables**

85 Table 1: Precedence of Production and Layout Job Attributes8
 86 Table 2: State Variables.....11
 87 Table 2.1: allowedValueList for *PrinterState*14
 88 Table 2.2: allowedValueList for *PrinterStateReasons*15
 89 Table 2.3: allowedValueList for *XHTMLImageSupported*16
 90 Table 2.4: allowedValueList for *ColorSupported*16
 91 Table 2.5: allowedValueList for *DocumentFormat*18
 92 Table 2.6: allowedValueList for *Sides*19
 93 Table 2.7: allowedValueList for *NumberUp*19
 94 Table 2.8: allowedValueList for *OrientationRequested*.....20
 95 Table 2.9: allowedValueList for *MediaSize*21
 96 Table 2.10: allowedValueList for *MediaType*.....22
 97 Table 2.11: allowedValueList for *PrintQuality*.....22
 98 Table 3: Event Moderation23
 99 Table 4: Synchronization of Evented Variables25
 100 Table 5.: Transition Actions Used in Table 4.....26
 101 Table 6: Actions27
 102 Table 7: Arguments for *CreateJob*28
 103 Table 8: Arguments for *CancelJob*29
 104 Table 9: Arguments for *GetPrinterAttributes*.....30
 105 Table 10: Arguments for *GetJobAttributes*30
 106 Table 11: Common Error Codes.....32
 107 Table 12: Basic IPP data type mappings34

108	Table 13: Derived data type mappings	35
109	Table 14: Structured Data Type mapping.....	35
110	List of Figures	
111	Figure 1 - Printer Device and Services	33
112		

113 1. Overview and Scope

114 This service definition is compliant with the UPnP Device Architecture version *1.0*.

115 This service-type enables the following functions:

- 116 • *Printing*

117 This service template does not address:

- 118 • *Faxing*

119 2. Service Modeling Definitions

120 2.1. ServiceType

121 A service that is compliant with this template is identified with the following service type: **urn:schemas-upnp-**
122 **org:service:***PrintBasic:1*.

123 2.2. Terminology

124 This section defines terms that are used throughout this specification. These terms are always capitalized in order to
125 indicate that they have the meaning defined in this section.

126 2.2.1. Conformance Terminology

127 The following terms have special meaning relating to conformance and so are always indicated in all capital letters:

- 128 a) **MUST** - This word, or the term "**REQUIRED**", mean that the definition is an absolute requirement of the
129 specification.
- 130 b) **MUST NOT** - This phrase means that the definition is an absolute prohibition of the specification.
- 131 c) **SHOULD** - This word, or the adjective "**RECOMMENDED**", mean that there may exist valid reasons in
132 particular circumstances to ignore a particular item, but the full implications must be understood and
133 carefully weighed before choosing a different course.
- 134 d) **SHOULD NOT** - This phrase, or the phrase "**NOT RECOMMENDED**" mean that there may exist valid
135 reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full
136 implications should be understood and the case carefully weighed before implementing any behavior
137 described with this label.
- 138 e) **MAY** - This word, or the adjective "**OPTIONAL**", mean that an item is truly optional. One vendor may
139 choose to include the item because a particular marketplace requires it or because the vendor feels that it
140 enhances the product while another vendor may omit the same item. An implementation which does not
141 include a particular option **MUST** be prepared to interoperate with another implementation which does
142 include the option, though perhaps with reduced functionality. An implementation which does include a
143 particular option **MUST** be prepared to interoperate with another implementation which does not include
144 the option

145 **2.2.2. Other terminology**

146 This document uses the terminology defined in the UPnP Architecture document, such as: action, SST variable, and
147 action parameter. This sub-section defines the following additional terms which are capitalized in order to indicate
148 their specific meaning as defined in this section.

- 149 a) Print Service (or Printer) - the UPnP entity that accepts actions from UCP (clients), returns responses, and
150 sends events.
- 151 b) PDL Data Stream - the stream of data to be printed as represented in a specified document format.
- 152 c) Production Job Attributes - job attributes that are not inherent to the PDL Data Stream and so the UCP
153 MAY override the PDL Data Stream instructions, if any, by supplying corresponding IN parameters when
154 submitting the job (see section 2.4).
- 155 d) Layout Job Attributes - job attributes that are inherent to the PDL Data Stream and *cannot* be overridden
156 by supplying corresponding IN parameters when submitting the job (see section 2.4).
- 157 e) Comma Separated Value (CSV) - a variable that contains multiple string values separated by the US-ASCII
158 COMMA (',') character (see section 2.5.1.1).
- 159 f) Distinguished Value – a special value defined by this specification for some action IN parameters. Use of
160 Distinguished Value IN parameter allows a PDL Data Stream corresponding value to take effect when it
161 would normally be overridden by the IN parameter. In the case where the Distinguished Value is absent in
162 the PDL data stream and the IN parameter value is specified as 'device-setting', the Service uses its
163 <defaultValue> value for the IN parameter. See section 2.6.2.
- 164 g) Tracked Job - a UPnP or non-UPnP job that is visible to a UPnP control point, i.e., has a JobId and appears
165 in the JobIdList, and on which the control point can perform any of the Job operations defined in this
166 document.
- 167 h) Untracked Job - a non-UPnP job that is not visible to a UPnP control point, i.e., does not have a JobId and
168 does not appear in the JobIdList, and on which the control point cannot perform any of the Job operations
169 defined in this document.

170 **2.2.3. Notation: use of quotation marks**

171 Throughout this document, single quotes (') are used around literal string and integer values in running text, but not
172 in Tables. The single quotes are not part of the values. Double quotes (") are used around words in running text to
173 indicate special English meanings. Variable names, parameters names, and action names are not quoted.

174 **2.3. References**

175 This section lists the references that this document refers to and the tag inside square brackets that is used for each
176 such reference:

177 [DEVICE] - UPnP Device Architecture, version 1.0.

178 [HTTP] - RFC 2616 "Hypertext Transfer Protocol -- HTTP/1.1", R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L.
179 Masinter, P. Leach, T. Berners-Lee. June 1999. (Format: TXT=422317, PS=5529857, PDF=550558 bytes)
180 (Obsoletes RFC2068) (Updated by RFC2817) (Status: DRAFT STANDARD)

181 [MODEL] - RFC 2566 "Internet Printing Protocol/1.0 Model and Semantics", March 1999 and RFC 2911 "Internet
182 Printing Protocol/1.1 Model and Semantics", September 2000, standards. Available at: <http://www.ietf.org>

- 183 [PWG5101.1] *IEEE-ISTO 5101.1-2001 Media Standardized Names <work in progress>*,
184 *ftp://ftp.pwg.org/pub/pwg/standards/pwg5101.1.pdf, .doc, .rtf for standardized names*
- 185 [UPnP-ENHANCED] - Albright, S., Hastings, T., Zehler, P., and G Shults, "PrintEnhancedLayout:0.10 Service
186 Template For UPnP Version 1.0", work in progress, TBD, 2001.
- 187 [XHTML-PRINT] - "XHTML (tm) - Print", version 0.60, May 11, 2001, <work in progress>, Available at:
188 <ftp://ftp.lexmark.com/pub/standards/xhtmll-print.pdf>
- 189 [MULTIPLEXED] - R. Herriot, "The MIME Application/Multiplexed Content-type", June 26, 2001, available at:
190 <http://search.ietf.org/internet-drafts/draft-herriot-application-multiplexed-04.txt> (Subsequent versions, if any, will be
191 available from the same location with the "04" incremented, and eventually as an information RFC.)
192

193 2.4. Intent of a Print Job

194 The intent of a Print job is indicated by the job attributes as represented by either:

- 195 - the IN parameters of the CreateJob action and/or
- 196 - the print instructions in the PDL Data stream.

197 Many job attributes MAY be specified by either or both methods. This section defines the precedence between
198 these two representations of the intent of a print job.

199 2.4.1. Production vs. Layout Job Attributes

200 This specification distinguishes two classes of such job attributes—*Production* and *Layout*. A Layout Job Attribute
201 is one that is inherent to the print output and cannot be overridden by IN parameters when the job is created. A
202 Production Job Attribute is one that can reasonably change at the different times when the job is printed without
203 affecting important job characteristics. Obvious examples of Production Attributes are number of copies, number of
204 sides and number of logical pages per physical sheet of paper, provided that when such Production Attributes are
205 represented in the PDL Data Stream they are represented as print instructions. However, if number of copies or
206 number of logical pages per physical sheet of paper is represented by repetitions of the PDL Data Stream, instead of
207 a print instruction in the PDL Data Stream, such a representation is not considered a Production Job Attribute and so
208 an IN parameter does not override such a representation.

209 Job attributes are partitioned between Production and Layout as follows:

210 Production Job Attributes (Job Attributes takes precedence):

- 211 JobName
- 212 JobOriginatingUserName
- 213 Copies
- 214 Sides
- 215 NumberUp
- 216 PrintQuality

217 Layout Job Attributes (data stream takes precedence):

- 218 OrientationRequested
- 219 MediaSize
- 220 MediaType

2.4.2. Precedence of Production vs. Layout Job Attributes

The UCP MUST supply an allowed value for each of the IN parameters defined for the CreateJob action (see section 2.8.1) or CreateEnhancedJob (see [UPnP-ENHANCED]). The PDL Data Stream MAY also have a value for any Production or Layout attribute represented as a print instruction. The UCP MAY supply the Distinguished Value defined by this document for each IN parameter to request the Printer to use its <defaultValue> value (see section 2.2.2, term f) and section 2.6.2) in case the corresponding print instruction in the PDL Data Stream is absent. The Printer SHOULD take the following action depending on the values supplied by the UCP in the CreateJob IN parameter and supplied in the PDL Data Stream for each given job attribute:

Table 1: Precedence of Production and Layout Job Attributes

Type of job attribute	IN parameter	PDL Data Stream	Printer SHOULD
Production attribute:	<Distinguished Value>	absent	use <defaultValue> in SCPD
	X	absent	use X
	<Distinguished Value>	Y	use Y
	X	X	use X
	X	Y	use X (IN <i>higher</i> than PDL) **
Layout attribute:	<Distinguished Value>	absent	use <defaultValue> in SCPD
	X	absent	use X
	<Distinguished Value>	Y	use Y
	X	X	use X
	X	Y	use Y (PDL <i>higher</i> than IN) **

** Only when both are supplied, does the precedence depend on whether the attribute is a Production Attribute or a Layout Attribute. Production IN parameters take precedence, while Layout PDL print instructions take precedence.

NOTE: Even for Layout Attributes, the IN parameter value supplied in the CreateJob action will be used as long as no overriding value is found in the PDL Data Stream itself.

2.5. State Variables

2.5.1. Derived data types

This section defines some derived data types that are represented as UPnP string data types with special syntax.

2.5.1.1. Comma Separated Value (CSV) Lists

The UPnP printer service uses variables that represent lists, or one-dimensional arrays, of values. Examples include the supported sets of document formats and media stock. The UPnP Device Architecture, Version 1.0 [DEVICE], does not provide for either an array type or a list type, so a list type is defined here. Lists may either be homogeneous (all values are the same type) or heterogeneous (values of different types are allowed). The data type of a homogeneous list is *string (CSV x)*, where *x* is the type of the individual values. The data type of a heterogeneous list is of the form *string (CSV x,y,z)*, where *x*, *y* and *z* are the types of individual element values. If the number of elements in the heterogeneous list is too large to show each type individually, that variable type is represented as *string (CSV heterogeneous)*, and the variable description includes additional information as to the expected sequence of values appearing in the list and their corresponding types.

- 248 ○ A list is represented as a UPnP String type.
- 249 ○ Values within a list are separated by commas.
- 250 ○ Only three value types are used as CSV elements in this specification—string, integer and boolean.
- 251 ○ Integer values are represented in CSVs with the same syntax as the int data type specified in [DEVICE]
- 252 (i.e., optional leading sign, optional leading zeroes)
- 253 ○ Boolean values are represented in CSVs as either ‘0’ for false or ‘1’ for true (which is a subset of the
- 254 defined boolean data type values specified in [DEVICE]: ‘0’, ‘false’, ‘no’, ‘1’, ‘true’, ‘yes’.
- 255 ○ String values are represented in CSVs with the same syntax as the string data types specified in [DEVICE]
- 256 (i.e., any Unicode string), with two exceptions that are represented using a backslash escape character:
- 257 ○ The comma (’,’) is represented as ‘\,’.
- 258 ○ The backslash (‘\’) is represented as ‘\\’.
- 259 ○ Any white space before, after, or interior to a string value is part of that string value. White space before,
- 260 after, or interior to any other data type is not allowed.

261 Examples:

Type refinement of string	Value	Comments
CSV string	text/xml,application/vnd.hp-PCL,application/postscript	List of three document types
CSV int	1,-5,006,0,+7	List of 5 integers.
CSV boolean	0,1,1,0	List of 4 booleans
CSV string	Smith\, Fred,Jones\, Davey	List of 2 user names, “Smith, Fred” and “Jones, Davey”
CSV i4,string,u2	-29837, string with leading blanks,0	Note that the second value is “ string with leading blanks”
CSV i4	3, 4	Illegal CSV. White space is not allowed as part of an integer value.
CSV string	,,	List of 3 empty string values
CSV heterogeneous	Alice,Marketing,5,Susan,R&D,21,David,Finance,7	List of unspecified number of people and associated attributes. Each person is described by 3 elements, a name <i>string</i> , a department <i>string</i> and years-of-service <i>u2</i> .

262

263 2.5.1.2. State variables, actions and action parameters

264 All state variables, actions and action parameters are mixed case with the first letter of each word being capitalized.

265 Most of these variables, actions and parameters are derived directly from IPP by removing the hyphens and up-

266 casing the first letter of each word. Unless specified otherwise, all variable values and action parameter values are

267 all lower case with hyphens, same as in IPP. See Internet Printing Protocol/1.0 Model and Semantics (RFC 2566)

268 and Internet Printing Protocol/1.1 Model and Semantics (RFC 2911), hereafter referred to as [MODEL]. The action

269 and attribute descriptions in these tables are only a brief summary. Implementations MUST conform to the

270 complete semantics specified in these referenced documents for each attribute indicated with [MODEL] in order to

271 achieve the kind of interoperability between client and Printer implementations of different vendors IPP has

272 demonstrated. A full description of their meaning can be found in the indicated sections in [MODEL].

273 2.6. Service State Table

274 A conforming UPnP Print Service implementation MUST support all of the Printer Service State Variables in the
275 Service State Table (SST). The first part of the Service State Table contains variables that represent Printer
276 attributes and the second part contains variables that represent Job attributes.

277 2.6.1. The Printer's supported and default values

278 The table below defines "Allowed Values" for each SST variable. The values in a Service Description's
279 <allowedValueList> element are the actual values supported by the Print Service instance (Printer).

280 Each SST variable definition in this document specifies whether or not vendors in their Service Description MAY
281 subset and/or extend the <allowedValueList> element in their Service Description from those "Allowed Values"
282 values given in this document. The Printer's "current" <allowedValueList> and <defaultValue> values may or may
283 not be the same as the factory supported and default values, respectively, for that parameter, i.e., someone may have
284 changed the settings from the factory-supplied values. Any <allowedValueList> and <defaultValue> element value
285 MAY be changed at any time after Service Discovery. Furthermore, the current <allowedValueList> and
286 <defaultValue> values for a job parameter could also possibly change between invocations of the action that uses it,
287 for example, if someone reconfigures the Printer's "current" device setting for that parameter. However, the UPnP
288 Device Architecture, version 1.0 [DEVICE], states that any change to the <allowedValueList> or <defaultValue>
289 element requires the printer to issue an "ssdp:byebye" and then re-advertise itself. Each of the values in the
290 <defaultValue> elements is implementation specific, but MUST be one of the values from the Service Description's
291 associated <allowedValueList> element, if present.

292 2.6.2. The Distinguished Value used to avoid action override of PDL

293 Some Print Service actions have IN parameters that will always override any corresponding value that might be
294 provided in the PDL data stream (see section 2.4.2). For those situations where the UCP prefers to let the PDL data
295 stream value override the IN parameter, the PrintBasic Service has added the Distinguished Value 'device-setting'
296 to the <allowedValueList> of the associated state variable. In the case where the Distinguished Value is absent in
297 the PDL data stream and the IN parameter value is specified as 'device-setting', the Service uses its <defaultValue>
298 value for the IN parameter. For example, see CreateJob action, section 2.8.1. When the UCP supplies the
299 Distinguished Value for such an IN parameter, the Print Service MUST process the action following the
300 corresponding print instruction in the PDL Data Stream, if present. If absent, the Print Service MUST process the
301 action as if the Service's then current <defaultValue> for that IN parameter value had been supplied by the UCP. In
302 other words, the Service's then current <defaultValue> value has lower precedence than the PDL Data Stream. All
303 implementations MUST support all Distinguished Value parameters defined herein. The two preceding
304 requirements also mean that the Distinguished Value for a variable MUST be included in the variable's allowed
305 value set, even if the vendor is subsetting the allowed value set. However, the Distinguished Value itself MUST
306 NOT be used for the actual value of the <defaultValue> element in the SCPD. Note: the Distinguished Values
307 defined herein for a variable/parameter are not otherwise valid values for the variable/parameter.

308 The value used as the Distinguished Value for a parameter, is specified in the definition of the parameter's
309 associated state variable. This guarantees uniqueness of the Distinguished Value across all actions that might use it.
310 Any vendor extensions to the set of Print Service actions that use IN parameters with an associated variable that has
311 a defined Distinguished Value SHOULD also support the use of Distinguished Values in their action invocations.
312 Any vendor extension that does support such Distinguished Values in their actions MUST use the same
313 Distinguished Value that is defined in this document. While vendors may use the Distinguished Value concept in
314 their Print Service extensions, this specification provides no mechanism for indicating either that Distinguished
315 Values are supported or the actual Distinguished Value used for a specific variable/parameter.

316 The Distinguished Value for all string variables defined herein is the string 'device-setting'. For any vendor
317 extensions, the Distinguished Value for all string variables MUST be 'device-setting'. The Distinguished Value for

318 all integer variables defined herein is the value '0'. For any vendor extensions, the Distinguished Value for integer
 319 variables SHOULD be '0' (or '-1' if '0' is otherwise a useful value).

320 **2.6.3. Purposes of the SST State Variables**

321 The first part of the SST defines the Printer attributes. The second part of the SST defines the Job attributes. Many
 322 of the Job attributes in the SST are present solely for the purpose of meeting the UPnP Device Architecture
 323 [DEVICE] requirement that all action parameters MUST have a related SST variable. The full specification for
 324 such action parameters is given with the variable in the SST. Some of the Printer attributes can be queried with the
 325 GetPrinterAttributes action (see section 2.5.3) and some of the Job attributes can be queried for a specified job with
 326 the GetJobAttributes action (see section 2.5.4)

327 **Table 2: State Variables**

Variable Name	Req. or Opt. ¹	Data Type	Allowed Value	Default Value	Eng. Units
<i>Printer Attributes</i>					
<i>VariableName</i>	<i>R</i>	<i>string</i>	<i>TBD</i>	<i>DEFAULT</i>	<i>TBD</i>
PrinterName	<i>R</i>	<i>string</i>		<i><implementation specific></i>	<i>N/A</i>
PrinterLocation	<i>R</i>	<i>string</i>		<i><implementation specific></i>	<i>N/A</i>
Deviceld	<i>R</i>	<i>string</i>		<i><implementation specific></i>	<i>N/A</i>
PrinterState	<i>R</i>	<i>string</i>	<i>See section 2.6.7</i>	<i>idle</i>	<i>N/A</i>
PrinterStateReasons	<i>R</i>	<i>string</i>	<i>See section 2.6.8</i>	<i>none</i>	<i>N/A</i>
XHTMLImageSupported	<i>R</i>	<i>string</i>	<i>See section 2.6.9</i>	<i>image/jpeg</i>	<i>N/A</i>
ColorSupported	<i>R</i>	<i>boolean</i>	<i>See section 2.6.10</i>	<i><implementation specific></i>	<i>N/A</i>
JobIdList	<i>R</i>	<i>string (CSV i4)</i>	<i>See section 2.6.11</i>	<i><empty String></i>	<i>N/A</i>
JobId	<i>R</i>	<i>i4</i>	<i>Range: 0 to 2³¹-1</i>	<i>0</i>	<i>N/A</i>
JobEndState	<i>R</i>	<i>string (CSV i4,string, string,i4, string)</i>	<i>See section 2.6.13</i>	<i><empty String></i>	<i>N/A</i>

Variable Name	Req. or Opt. ¹	Data Type	Allowed Value	Default Value	Eng. Units
<i>Job Attributes</i>					
JobName	R	<u>string</u>		<empty string>	N/A
JobOriginatingUserName	R	<u>string</u>		<empty string>	N/A
DocumentFormat	R	<u>string</u>	See section 2.6.16	<implementation specific> RECOMMENDED value: application/vnd.pwg-xhtml-print+xml See Note below.	N/A
Copies	R	<u>i4</u>	Range: 0 to 2 ³¹ -1	<implementation specific> RECOMMENDED value: 1	N/A
Sides	R	<u>string</u>	See section 2.6.18	<implementation specific> RECOMMENDED value: one-sided	N/A
NumberUp	R	<u>string</u>	See section 2.6.19	<implementation specific> RECOMMENDED value: 1	N/A
OrientationRequested	R	<u>string</u>	See section 2.6.20	<implementation specific> RECOMMENDED value: portrait	N/A
MediaSize	R	<u>string</u>	See section 2.6.21	<implementation specific>	N/A
MediaType	R	<u>string</u>	See section 2.6.22	<implementation specific> RECOMMENDED value: Stationery (if supported)	N/A
PrintQuality	R	<u>string</u>	See section 2.6.23	<implementation specific> RECOMMENDED value: normal	N/A
DataSink	R	<u>uri</u>		<empty string>	N/A
JobMediaSheetsCompleted	R	<u>i4</u>	Range: -1 to 2 ³¹ -1	0	N/A
<i>Non-standard state variables implemented by a UPnP vendor go here.</i>	X	TBD	TBD	TBD	TBD

328 ¹ R = REQUIRED, O = Optional, X = Non-standard.

329 NOTE: The value “application/vnd.pwg-xhtml-print+xml” MUST be shortened to 31 characters for
 330 interoperability reasons. This value MUST be: “application/vnd.pwg-xhtml-print”. Any additional values that are
 331 used by a vendor MUST also be 31 characters or less for interoperability.

332 2.6.4. PrinterName

333 The administratively assigned user-friendly name of the Printer. How the Printer's Service Description
 334 <defaultValue> element is configured with this value is implementation-specific, e.g., local console, Presentation
 335 Service (web access). If the Device Service has only one device, then the Device's <friendlyName> and
 336 PrinterName are recommended to have the same value. However, if the Device contains several devices, the
 337 PrinterName identifies the Printer.

338 *(See [MODEL] section 4.4.4)*

339 **2.6.5. PrinterLocation**

340 *Indicates the location of the device. For example, "Bobby's room". How the Printer's Service Description*
 341 *<defaultValue> element is configured with this value is implementation-specific, e.g., local console, Presentation*
 342 *Service (web access).*

343 *(See [MODEL] section 4.4.4)*

344 **2.6.6. DeviceId**

345 *The value of this variable MUST exactly match the IEEE 1284-2000 Device ID string, except the length field MUST*
 346 *not be specified.. The value is assigned by the Printer vendor and MUST NOT be localized by the Print Service.*

347 *The IEEE 1284-2000 Device ID is a length field followed by a case-sensitive string of ASCII characters defining*
 348 *peripheral characteristics and/or capabilities. For the purposes of this specification, the length bytes MUST NOT*
 349 *be included. The Device ID sequence is composed of a series of keys and values of the form:*

350 *key: value {,value} repeated for each key*

351 *As indicated, each key will have one value, and MAY have more than one value. The minimum necessary keys (case-*
 352 *sensitive) are MANUFACTURER, COMMAND SET, and MODEL. (These keys MAY be abbreviated as MFG,*
 353 *CMD, and MDL respectively.) Each implementation MUST supply these three keys and possibly additional ones as*
 354 *well. Each key (and each value) is a string of characters. Any characters except colon (:), comma (,), and semi-*
 355 *colon (;) MAY be included as part of the key (or value) string. Any leading or trailing white space (SPACE[x'20'],*
 356 *TAB[x'09'], VTAB[x'0B'], CR[x'0D'], NL[x'0A'], or FF[x'0C']) in the string is ignored by the parsing program (but*
 357 *is still counted as part of the overall length of the sequence).*

358 *An example ID String, showing optional comment and active command set keys and their associated values (the text*
 359 *is actually all on one line):*

360

361 `MANUFACTURER:ACME Manufacturing;`

362 `COMMAND SET:PCL,PJL,PS,XHTML-Print+xml;`

363 `MODEL:LaserBeam 9;`

364 `COMMENT:Anything you like;`

365 `ACTIVE COMMAND SET:PCL;`

366 *(See IEEE 1284-2000 clause 7.6)*

367 *Note: One of the purposes of the DeviceId variable is to select a printer driver for those UCPs that need a printer*
 368 *driver. The values of the COMMAND SET key are interpreted by the printer driver provided by the vendor and so*
 369 *are vendor-defined, rather than being standardized.*

370 **2.6.7. PrinterState**

371 *Identifies the current state of the service. Values:*

372 ***idle** - new jobs can start processing immediately without waiting.*

373 ***processing** - jobs (Tracked or Untracked) are processing; new jobs will wait before processing, i.e., are*
 374 *said to be pending.*

375 ***stopped** - no jobs can be processed and intervention is needed.*

376 *(See [MODEL] section 4.4.11)*

377 *Vendors MUST NOT subset or extend allowed values.*

378 **Table 2.1:** allowedValueList for **PrinterState**

Value	Req. or Opt.
<i>idle</i>	<u>R</u>
<i>processing</i>	<u>R</u>
<i>stopped</i>	<u>R</u>

379

380 **2.6.8. PrinterStateReasons**

381 *Indicates additional information about why the Printer is in its current state. Multiple conditions MAY exist. The*
 382 *vendor chooses the single value for PrinterStateReasons variable to indicate the most important condition.*

383 *Note: some of these reasons describe state of the printer that cannot be entered on the basis of the currently defined*
 384 *UPnP actions set. For example the printer can be 'paused'; there is no PausePrinter action. The reason these states*
 385 *are presented is because some other protocol (or console action) can have caused the printer to enter that state.*

386 *Reason values:*

387 **none** - Indicates that there are no current state reasons

388 **attention-required** - The device has stopped for a reason other than the PrinterStateReasons listed here and
 389 requires human intervention before it can continue.

390 **media-jam** - The device has a media jam.

391 **paused** - Someone has paused the printer and the PrinterState is 'stopped'. In this state, a Printer will not produce
 392 printed output.

393 **door-open** - One or more covers on the device are open.

394 **media-low** - At least one input tray is low on media.

395 **media-empty** - At least one input tray is empty.

396 **output-area-almost-full** - One or more output area is almost full (e.g. tray, stacker, collator).

397 **output-area-full** - One or more output area is full, e.g., tray, stacker, collator.

398 **marker-supply-low** - The device is low on at least one marker supply, e.g., toner, ink, ribbon.

399 **marker-supply-empty** - The device is out of at least one marker supply, e.g., toner, ink, ribbon.

400 **marker-failure** - The device has at least one marking device which has failed and requires service or replacement.

401 **media-change-request** - A job has been submitted that is requesting media that is currently not loaded. The job
 402 has specified a particular MediaSize and MediaType parameter value combination that is not loaded, although the
 403 Printer supports that combination.

404 (See [MODEL] section 4.4.12. The IPP severity suffix **MUST NOT** be included and, unlike IPP, only one value
 405 **MUST** occur at a time.)

406 *Vendors **MUST** support the values that represent conditions that are detectable in their implementation. Therefore,*
 407 *vendors **MAY** subset allowed values if specific PrinterStateReasons are undetectable in their implementation.*

408 *Vendors **MAY** extend allowed values. However, Printer vendors need to understand the implications of extending*
 409 *this list on a UCP. The UCP usually localizes the PrinterStateReasons value (as with other string variable values)*
 410 *to the human language of the user. However, such a Printer vendor extension value will not be recognized by the*
 411 *UCP. As a Fallback presentation, the UCP **MAY** display the value received as is, which should be in English and*
 412 *therefore, might not be understandable by the user. Alternatively, the vendor might use the general*
 413 *PrinterStateReasons value: 'attention-required' and then explain the problem on the Printer console which the user*
 414 *would see when they are by the Printer.*

415 **Table 2.2:** allowedValueList for *PrinterStateReasons*

Value	Req. or Opt. ³
<i>none</i>	<u>R</u>
<i>attention-required</i>	<u>O</u>
<i>media-jam</i>	<u>O</u>
<i>paused</i>	<u>O</u>
<i>door-open</i>	<u>O</u>
<i>media-low</i>	<u>O</u>
<i>media-empty</i>	<u>O</u>
<i>output-area-almost-full</i>	<u>O</u>
<i>output-area-full</i>	<u>O</u>
<i>marker-supply-low</i>	<u>O</u>
<i>marker-supply-empty</i>	<u>O</u>
<i>media-change-request</i>	<u>O</u>
<i>Vendor-defined</i>	<u>O</u>

416 ³ *Vendors MUST support the values that represent conditions that are detectable in their implementation*

417

418 **2.6.9. XHTMLImageSupported**

419 *Identifies the Image formats supported by the Printer. The image MUST be sent as part of an XHTML-Print*
 420 *document[XHTML-PRINT], either interleaved within XHTML-Print using the MIME Application/Multiplexed*
 421 *Content Type [MULTIPLEXED] or as a referenced object. The Printer MUST support both the inline and*
 422 *referenced object forms as defined in XHTML-Print [XHTML-PRINT]. A printer device vendor MAY choose to*
 423 *support other XHTMLImageSupported formats, however, there is no requirement to support the MIME*
 424 *Application/Multiplexed Content Type [MULTIPLEXED] for these other image formats.*

425 *All UPnP printers MUST support at least the 'image/jpeg' image format.*

426 *Vendors MAY extend the allowed values for this attribute.*

427 *Note: 'image/jpeg' is registered as a MIME Media Type with IANA.*

428 **Table 2.3: allowedValueList for *XHTMLImageSupported***

Value	Req. or Opt.
<i>image/jpeg</i>	<u>R</u>
<i>Vendor-defined</i>	<u>O</u>

429

430 **2.6.10. ColorSupported**

431 *Identifies whether or not the device is capable of multi-hued color printing. A printer that is capable of full color*
 432 *output has a value of '1' (TRUE). A grayscale capable or business graphics capable printer has the value of '0'*
 433 *(FALSE), as would a highlight printer.*

434 *(Note: though this variable is named the same as the corresponding IPP "color-supported" (boolean) Printer*
 435 *attribute, the semantics differ: A UPnP Printer must be capable of full color output in order to have a '1' (TRUE)*
 436 *value. See [MODEL] section 4.4.26)*

437 *All UPnP printers MUST support either the '0' or the '1' value.*

438 *Vendors MUST NOT extend the allowed values for this attribute.*

439 **Table 2.4: allowedValueList for *ColorSupported***

Value	Req. or Opt.
<i>0</i>	<u>O</u>
<i>1</i>	<u>O</u>

440

441 **2.6.11. JobIdList**

442 *The list of JobId values for all tracked jobs known by the Print Service, i.e. all active and queued jobs, but NOT jobs*
 443 *that have completed, been aborted by the print service, or canceled. It is RECOMMENDED that jobs submitted to*
 444 *the Printer by protocols other than UPnP be represented in JobIdList.*

445 *The list is a sequence of Comma Separated i4 Values (CSV i4 - see section 2.5.1.1). Each value is a JobId of a job*
 446 *on the printer. The values range from 1 to 2³¹-1. The list is in the order that the jobs are expected to be completed.*

447 *The first job in the list is either currently printing, attempting to print (but the Printer is stopped), or is the next job*
 448 *to print (if no jobs are currently printing or all jobs are in the 'pending-held' state). The last job in the list will be*
 449 *printed last. The first JobId in the list is removed when the job completes or is aborted. The corresponding JobId*
 450 *in the list is removed when a job is canceled (see section 2.8.2).*

451 *When all jobs are completed, cancelled or aborted, the JobIdList variable is an empty string.*

452 *The Print Service, on receipt of a new job, generates a JobId which identifies the new Job on that Print Service.*

453 *The JobId is placed in the appropriate place in the JobIdList. The Print Service returns the value of the JobId*
 454 *parameter as part of the response to a CreateJob action.*

455 **2.6.12. JobId**

456 *The JobId of the current job, i.e., the job that has caused the PrinterState variable to be 'processing' or 'stopped'.*
 457 *The JobId MUST be the first JobId in the JobIdList or 0. If there is no current job, i.e., the PrinterState is 'idle'*
 458 *(there are no jobs, or all jobs are pending or held), then JobId contains a 0 which is an invalid JobId for a job). If*
 459 *JobId is 0, the printer is either idle OR a non-UPnP job is printing (and the Printer implementation has chosen*
 460 *NOT to display non-UPnP jobs, i.e., the job is an Untracked Job).*

461 *(See [MODEL] section 4.3.2)*

462 **2.6.13. JobEndState**

463 *This variable holds the “terminating” state of the job most recently removed from the JobIdList. It is evented; it is*
 464 *triggered when any JobId is removed from the JobIdList. However, the JobEndState is not an OUT parameter of*
 465 *any action, so it is not available to a client via polling.*

466 *JobEndState is a heterogeneous CSV list of five items: JobId, JobName, JobOriginatingUserName,*
 467 *JobMediaSheetsCompleted, job-completion-state (same order as the GetJobAttributes OUT parameters, plus the*
 468 *job-completion-state).*

469 ***JobId:** the JobId of the job being removed (see section 2.6.12)*

470 ***JobName:** The name of the job. See section 2.6.14.*

471 ***JobOriginatingUserName:** The name of the user that submitted the job. See section 2.6.15.*

472 ***JobMediaSheetsCompleted:** If JobId was the “active” job, i.e., the first job in JobIdList, this is the final*
 473 *value of JobMediaSheetsCompleted for the job. Otherwise, this value is ‘0’. See section 2.6.25.*

474 *job-completion-state: One of ‘aborted’, ‘canceled’ or ‘successful’ as defined below:*

475 ***aborted:** The job did not complete successfully, for one of two reasons—either (1) the printer*
 476 *encountered a non-recoverable error while processing the job or attempting to receive the data,*
 477 *or (2) the job was created by the CreateEnhancedJob action (see [UPnP-ENHANCED]) and the*
 478 *printer detected during processing that the job requirements could not be met.*

479 ***successful:** The job printed successfully all of the pages of the job and the sheets have been*
 480 *stacked in the output bin.*

481 ***canceled:** The job was canceled either by a CancelJob action or the equivalent in another*
 482 *protocol.*

483 **2.6.14. JobName**

484 *The user-friendly name of the job. It is RECOMMENDED that the client (UCP) supply a value to help a user easily*
 485 *distinguish between the jobs that he/she has submitted.*

486 **2.6.15. JobOriginatingUserName**

487 *The name of the user that submitted the job. Either supplied by the client (UCP) or by the security infrastructure, if*
 488 *any. It is RECOMMENDED that the client (UCP) supply a value to help a user easily distinguish between the jobs*
 489 *that he/she has submitted and jobs that others have submitted.*

490 **2.6.16. DocumentFormat**

491 *Identifies the DocumentFormat of the job as a mime media type. One special value is ‘application/octet-stream’. If*
 492 *the Printer service supports this value, the Printer service MUST be capable of auto-sensing the format of the*
 493 *document data.*

494 *Another special value is ‘unknown’. This value is intended for the UCP to supply that does not know the document*
 495 *format of the document data. The behavior of the Printer when receiving the ‘unknown’ value is*
 496 *IMPLEMENTATION DEFINED. However, if the Printer can perform auto sensing of the data, (the*
 497 *‘application/octet-stream’ behavior), it is RECOMMENDED that it do so.*

498 *If the UCP (client) does not know the document format, it SHOULD supply the ‘application/octet-stream’ value and*
 499 *let the Printer determine the format, unless the Printer doesn’t support the ‘application/octet-stream’ value, in*
 500 *which case the UCP’s only recourse is to supply the special ‘unknown’ value. All UPnP printers MUST support at*
 501 *least the ‘application/vnd.pwg-xhtml-print’ document format[XHTML-PRINT] and the ‘unknown’ value.*

502 *(See [MODEL] section 4.1.9)*

503 *The vendors MAY extend the allowed values for this attribute, but MUST NOT support the ‘device-setting’*
 504 *Distinguished Value. The vendor MAY subset the allowed values as long as ‘application/vnd.pwg-xhtml-print’ and*
 505 *‘unknown’ remain as supported values.*

506 **Table 2.5:** allowedValueList for *DocumentFormat*

Value	Req. or Opt.
<i>unknown</i>	<u>R</u>
<i>application/vnd.pwg-xhtml-print+xml</i> <i>See NOTE below.</i>	<u>R</u>
<i>text/plain</i>	<u>O</u>
<i>text/plain;charset=utf-8</i>	<u>O</u>
<i>application/octet-stream</i>	<u>O</u>
<i>application/postscript</i>	<u>O</u>
<i>application/vnd.hp-PCL</i>	<u>O</u>
<i>Vendor-defined</i>	<u>O</u>

507 NOTE: The value “application/vnd.pwg-xhtml-print+xml” MUST be shortened to 31 characters for
508 interoperability reasons. This value MUST be: “application/vnd.pwg-xhtml-print”. Any additional values that are
509 used by a vendor MUST also be 31 characters or less for interoperability.

510 **2.6.17. Copies**

511 *Contains the number of copies of the document to be printed for the job. See [MODEL] section 4.2.5.*

512 *The ‘0’ Distinguished Value indicates that the control point wants the Printer to use its <defaultValue> value for*
513 *Copies, which MUST be greater than 0, but to allow that value to be overridden if a corresponding value is*
514 *encountered in the PDL Data Stream.*

515 *Vendors MAY subset the allowed values, but MUST support the ‘0’ Distinguished Value.*

516 *Vendors MUST NOT extend the allowed values.*

517 **2.6.18. Sides**

518 *Specifies how pages are to be imposed upon the sides of a selected medium for the job. Values:*

519 *one-sided*

520 *two-sided-long-edge*

521 *two-sided-short-edge*

522 *device-setting*

523

524 *(See [MODEL] section 4.2.8)*

525 *The ‘device-setting’ Distinguished Value indicates that the control point wants the Printer to use its <defaultValue>*
526 *value for Side, but to allow that value to be overridden if a corresponding value is encountered in the PDL Data*
527 *Stream.*

528 *Vendors MAY subset allowed values, but MUST support the ‘device-setting’ Distinguished Value.*

529 *Vendors MUST NOT extend allowed values.*

530 **Table 2.6: allowedValueList for Sides**

Value	Req. or Opt.
<i>device-setting</i>	<u>R</u>
<i>one-sided</i>	<u>R</u>
<i>two-sided-long-edge</i>	<u>Q</u>
<i>two-sided-short-edge</i>	<u>Q</u>

531

532 **2.6.19. NumberUp**

533 *Description: Indicates the number of PDL Data Stream pages to impose upon a single side of an instance of a*
 534 *selected medium for the job. Examples:*

535 *1 - One page per side.*

536 *2 - Two pages per side.*

537 *4 - Four pages per side.*

538 ***device-setting***

539 *The value is represented as ASCII decimal digits without leading zeros, so that the Allowed Values can be*
 540 *represented as individual integer (string) values in the range 1 to 2**31-1.*

541 *(See [MODEL] section 4.2.9)*

542 *The 'device-setting' Distinguished Value indicates that the control point wants the Printer to use its <defaultValue>*
 543 *value for NumberUp, but to allow that value to be overridden if a corresponding value is encountered in the PDL*
 544 *Data Stream.*

545 *Vendors MAY subset or extend allowed values, but MUST support the 'device-setting' Distinguished Value.*

546 **Table 2.7: allowedValueList for NumberUp**

Value	Req. or Opt.
<i>device-setting</i>	<u>R</u>
<u>1</u>	<u>R</u>
<u>2</u>	<u>Q</u>
<u>4</u>	<u>Q</u>
<i>Vendor-defined</i>	<u>Q</u>

547

548 **2.6.20. OrientationRequested**

549 *Indicates the desired orientation for printed pages for any MIME type format of the job. Which MIME type formats*
 550 *a Printer is able to change the orientation depends on implementation and MAY depend on the actual document*
 551 *content. Values:*

552 *portrait*

553 *landscape*

554 *reverse-landscape*

555 *reverse-portrait*

556 *device-setting*

557 (See [MODEL] section 4.2.10 which intends the “orientation-requested” attribute to apply to ‘text’ MIME types.)
 558 The ‘device-setting’ Distinguished Value indicates that the control point wants the Printer to use its <defaultValue>
 559 value for OrientationRequested, but to allow that value to be overridden if a corresponding value is encountered in
 560 the PDL Data Stream.

561 Vendors MAY subset allowed values, but MUST support the ‘device-setting’ Distinguished Value.

562 Vendors MUST NOT extend allowed values.

563 **Table 2.8: allowedValueList for OrientationRequested**

Value	Req. or Opt.
<i>device-setting</i>	<u>R</u>
<i>portrait</i>	<u>R</u>
<i>landscape</i>	<u>O</u>
<i>reverse-landscape</i>	<u>O</u>
<i>reverse-portrait</i>	<u>O</u>

564

565 **2.6.21. MediaSize**

566 Identifies the medium size name and dimensions that the Printer Service uses for all sheets of the job. Each value
 567 MUST include the name of the size followed by the dimensions in inches or millimeters followed by the “in” or
 568 “mm” suffix to indicate the units. Both the Inch and Millimeter dimension MAY include a non-zero decimal
 569 fraction set off by a period (.). The name of the size consists of a class part and a name part separated by an
 570 underscore (_). The class part MUST be “na”, “asme”, or “oe” for inch units and “iso”, “jis”, “jpn”, “prc”,
 571 “roc”, or “om” for metric units (see [PWG5101.1] for additional class names). The name part is set off by a
 572 second underscore (_) and the dimensions are separated by the lower case letter x. The shorter dimension MUST
 573 come first. See the Allowed Values for examples.

574 For sizes that do not have standard names, a UCP or a Print Service can create a customized name using the
 575 ‘custom_xxx’ class and name, where xxx indicates the custom name of the medium, followed by the dimensions in
 576 inches or millimeters as for standard names. For example, a custom 3.5 by 5.0 inch medium that, say, represents
 577 an index card, could be indicated by the string value:

578 `custom_index-card_3.5x5in`

579 The customized values configured for the Printer MUST be added to the Printer’s <allowedValueList>.

580 If a Printer supports the control point supplying custom names that are not one of the values in the Printer’s
 581 <allowedValueList> element, the Printer’s <allowedValueList> element MUST include both the
 582 ‘custom_max_IIIxJJmm’ and ‘custom_min_IIIxJJmm’ (and/or ‘custom_max_IIIxJJin’ and
 583 ‘custom_min_IIIxJJin’) Allowed Values to indicate the minimum and maximum custom sizes that the Printer will
 584 allow the control point to supply.

585 (See [PWG5101.1] for suggested media size names and their dimensions. These names SHOULD NOT use the
 586 “custom” class name.)

587 The ‘device-setting’ Distinguished Value indicates that the control point wants the Printer to use its <defaultValue>
 588 value for MediaSize, but to allow that value to be overridden if a corresponding value is encountered in the PDL
 589 Data Stream.

590 Vendors MAY subset and extend allowed values, but MUST support the ‘device-setting’ Distinguished Value.

591 How the Printer’s Service Description <defaultValue> and <allowedValueList> elements are configured with these
 592 values is implementation-specific, e.g., local console, Presentation Service (web access).

593 **Table 2.9:** allowedValueList for **MediaSize**

Value ³	Req. or Opt.
<i>device-setting</i>	<u>R</u>
<i>na_letter_8.5x11in</i>	<u>Q</u>
<i>na_legal_8.5x14in</i>	<u>Q</u>
<i>iso_a4_210x297mm</i>	<u>Q</u>
<i>iso_c5_162x229mm</i>	<u>Q</u>
<i>iso_dl_110x220mm</i>	<u>Q</u>
<i>jis_b4_257x364mm</i>	<u>Q</u>
<i>custom_xxx_IIIxJJmm</i>	<u>Q</u>
<i>custom_xxx_IIIxJJin</i>	<u>Q</u>
<i>custom_min_IIIxJJmm</i>	<u>Q</u>
<i>custom_max_IIIxJJin</i>	<u>Q</u>
<i>Vendor-defined (see [PWG5101.1])</i>	<u>Q</u>

594 ³ These values represent examples and are not intended to be exhaustive (see [PWG5101.1]).

595 **2.6.22. MediaType**

596 Identifies the medium type that the Printer Service uses for all impressions of the job. Example values:

597	stationery	Separately cut sheets of an opaque material
598	transparency	Separately cut sheets of a transparent material
599	envelope	Envelopes that can be used for conventional mailing purposes
600	labels	Label stock [For example, a sheet of peel-off labels].
601	photographic	Separately cut sheets of an opaque material to produce photographic quality images
602	cardstock	Separately cut sheets of an opaque material that is heavier and stiffer than stationery.
603	device-setting	Indicates that the control point wants the Printer to use its <defaultValue> value for
604		MediaType.

605 The values are a subset of and the descriptions are a taken verbatim from the Media Type Names in [PWG5101.1].
 606 The 'device-setting' Distinguished Value indicates that the control point wants the Printer to use its <defaultValue>
 607 value for MediaType, but to allow that value to be overridden if a corresponding value is encountered in the PDL
 608 Data Stream.

609 Vendors MAY subset or extend allowed values, but MUST support the 'device-setting' Distinguished Value. See
 610 [PWG5101.1] for additional example values.

611 How the Printer's Service Description <defaultValue> and <allowedValueList> elements are configured with these
 612 values is implementation-specific, e.g., local console, Presentation Service (web access).

613 **Table 2.10:** allowedValueList for *MediaType*

Value ³	Req. or Opt.
<i>device-setting</i>	<u>R</u>
<i>stationery</i>	<u>O</u>
<i>stationery-inkjet</i>	<u>O</u>
<i>transparency</i>	<u>O</u>
<i>envelope</i>	<u>O</u>
<i>labels</i>	<u>O</u>
<i>photographic</i>	<u>O</u>
<i>cardstock</i>	<u>O</u>
<i>Vendor-defined (see [PWG5101.1])</i>	<u>O</u>

614 ³ *These values represent examples and are not intended to be exhaustive (see [PWG5101.1]).*615 **2.6.23. PrintQuality**616 *Specifies the print quality requested for the job. Values:*617 *draft*618 *normal*619 *high*620 *device-setting*621 *(See [MODEL] section 4.2.13)*622 *The ‘device-setting’ Distinguished Value indicates that the control point wants the Printer to use its <defaultValue>*
623 *value for PrintQuality, but to allow that value to be overridden if a corresponding value is encountered in the PDL*
624 *Data Stream.*625 *Vendors MAY subset allowed values, but MUST support the ‘device-setting’ Distinguished Value.*626 *Vendors MUST NOT extend allowed values.*627 **Table 2.11:** allowedValueList for *PrintQuality*

Value	Req. or Opt.
<i>device-setting</i>	<u>R</u>
<i>draft</i>	<u>O</u>
<i>normal</i>	<u>R</u>
<i>high</i>	<u>O</u>

628

629 **2.6.24. DataSink**

630 Contains the URL to which the UCP is to send the HTTP Post operation (see section 2.8.5) for the job. This value
 631 is returned by the Printer in the CreateJob action response, rather than being supplied by the UCP in the CreateJob
 632 action request.

633 **2.6.25. JobMediaSheetsCompleted**

634 The number of media sheets completed for the job so far. The **JobMediaSheetsCompleted** value includes
 635 completion of stacking the output. If a Printer implementation does not know the number of media sheets completed,
 636 then it **MUST** return a -1 value to indicate “unknown”. If JobId is 0, then **JobMediaSheetsCompleted** **MUST** be 0
 637 (or -1, if the media sheets are unknown).

638 It is possible in some implementations that the final value of JobMediaSheetsCompleted is known, but that
 639 intermediate values are not known. In this case the Printer **SHOULD** return 0 for a job that is not active, -1 for an
 640 active job and the proper final value for completed jobs. The Printer **MUST** still return -1 for
 641 JobMediaSheetsCompleted when it does not know the value, even in situations that it normally would know the
 642 value. A UCP **MUST NOT** conclude that receipt of a value of -1 for JobMediaSheetsCompleted means that the
 643 Printer will always return -1. Even implementations that can never successfully count media sheets completed
 644 might still know that a canceled or aborted job never marked any paper, so it could properly return a value of ‘0’
 645 for JobMediaSheetsCompleted in the JobEndState variable.

646 **2.7. Eventing and Moderation**647 **Table 3: Event Moderation**

Variable Name	Evented	Moderated Event	Max Event Rate ¹ (sec)	Logical Combination	Min Delta per Event ²
<i>PrinterState</i>	<i>Yes</i>	<i>No</i>	<i>N/A</i>		<i>N/A</i>
<i>PrinterStateReasons</i>	<i>Yes</i>	<i>No</i>	<i>N/A</i>		<i>N/A</i>
<i>JobIdList</i>	<i>Yes</i>	<i>No</i>	<i>N/A</i>		<i>N/A</i>
<i>JobEndState</i>	<i>Yes</i>	<i>No</i>	<i>N/A</i>		<i>N/A</i>
<i>JobMediaSheetsCompleted</i>	<i>Yes</i>	<i>Yes</i>	<i>5</i>		<i>N/A</i>

648 ¹ Events containing this variable value **SHOULD** occur no more often than once every MaxEventRate seconds.

649 ² See 4.4, Eventing: Augmenting the UPnP Template Language in [DEVICE].

650 **2.7.1. Event Model**

651 The eventing model for the print service has three main purposes. First is to inform the UCP when there is a change
 652 in condition of the print device. Examples: the printer becomes idle, a paper jam occurs or the printer is low on
 653 paper. The PrinterState and PrinterStateReasons variables provide this information. Second is for job tracking.
 654 Events inform a UCP when a job is submitted, completed or removed from the job queue. The JobIdList and
 655 JobEndState provide this information. JobEndState indicates the final status of each job. It lets control points know
 656 whether it completed successfully or was canceled or aborted. Third is to inform a UCP of the progress of the
 657 current job. JobMediaSheetsCompleted is a moderated evented variable that updates an interested UCP on the
 658 number of impressions printed for the current job.

659 2.7.2. Synchronization of Evented Variables

660 Table 4 below describes how internal printer state changes affect the values of the five evented state variables, plus
661 the non-evented variable, JobId. These state changes can be forced by any of: a UCP invoking one of the print
662 service actions documented herein, a non-UPnP external action or printer internal events and conditions. The effect
663 of some non-UPnP external actions is indirect, i.e., they affect internal printer state immediately, but, if they result
664 in any UPnP-visible effect, the affect appears later. All of these indirect effects have to do with management of
665 untracked jobs. They are included in this table because their ultimate effect can be visible at some later time. A
666 UCP should be aware of this to fully understand observed behavior. For print service implementers, the complete
667 table is a guideline to the information that must be kept and how it is synchronized to guarantee that the externally
668 visible state variables are always correct.

669 In Table 4, column 1 contains the current value of the variable PrinterState. Column 2 lists the events that can
670 trigger an internal printer state change. Column 3 gives the new printer state and the complete set of actions taken
671 by the printer on the transition that is triggered by the column 2 event. In several cases, the actions taken depend on
672 other printer conditions in addition to the triggering event. Those situations are identified in the table by dividing
673 the lower right portion of the corresponding event “cell” into multiple subcells, one for each condition or set of
674 conditions that requires a different set of transition actions. The upper portion of the event cell is extended into
675 column 3, signifying that no transition action(s) can be specified for this event except when the conditions in the
676 event’s subcells are also considered. The word *invisible* in column 3 means there is no state change that could be
677 observed by a UPnP UCP. All of the actions listed in column 3 MUST be completed atomically relative to all
678 external UPnP observations.

679 For purposes of this document, atomically means:

- 680 1. From the viewpoint of any UCP observer external to the Print Service, all of the values change at the same
681 time. To achieve this, it is RECOMMENDED that all evented variables changed by this collected set of
682 actions appear in a single event message.
- 683 2. It is not possible through any query action for a UCP to detect that any single state variable has changed
684 unless it detects that all have changed and been properly updated to their new values.

685 To help understand the actions, let’s follow one transition through the tables. Find the entry in column 2 “Terminate
686 active job that was tracked. Its termination condition, T , is one of ‘successful’, ‘canceled’ or ‘aborted’.” Since it has subcell
687 entries, there is no direct entry in column 3. Assume the normal situation of a busy printer with more jobs queued
688 and that all of them are tracked. The relevant added condition is “Next job is tracked.” That takes us to column 3
689 with actions of “J3, M0, E1(T)”. Looking in Table 5 we see that J3 tells us to remove the first element of the JobIdList and set
690 the new value of JobId to the new first element of JobIdList. M0 tells us to reset JobMediaSheetsCompleted to ‘0’ if we track it,
691 or leave it at ‘-1’ if we don’t. E1(T) tells us to set JobEndState with all the corresponding values for the job just completed,
692 including the indicator whether it was ‘successful’, ‘canceled’ or ‘aborted’. Also note that the M2 value inside JobEndState is
693 set according to the actual final value of the sheets printed, if known.

694

695 **Table 4: Synchronization of Evented Variables**

State	Transition events (and conditions)	Transition actions
?	Initialize PrintBasic service	I, R0, J0, M0, E0
idle	CreateJob or CreateEnhancedJob or create non-UPnP tracked job	P, J1
	Create untracked job — action invoked by non-UPnP entity	P
	<printer error>	S, R1
processing	CreateJob or CreateEnhancedJob or create non-UPnP tracked job	P, J2
	Create untracked job — action invoked by non-UPnP entity	<i>invisible</i>
	Terminate active job that was tracked. Its termination condition, <i>T</i> , is one of ‘successful’, ‘canceled’ or ‘aborted’.	
	No more jobs.	I, J0, M0, E1(<i>T</i>)
	Next job is tracked.	J3, M0, E1(<i>T</i>)
	Next job is untracked, and there are no more tracked jobs.	J0, M0, E1(<i>T</i>)
	Next job is untracked, but there are still tracked jobs in the queue.	J4, M0, E1(<i>T</i>)
	Terminate tracked job that was not active. Its termination condition, <i>T</i> , is one of ‘canceled’ or ‘aborted’.	
	Terminate active job that was untracked.	
	No more jobs.	I, M0
	Next job is tracked.	J6, M0
	Next job is untracked.	<i>invisible</i>
	Terminate inactive job that was untracked.	
	Drop a sheet into the output tray that is not the last sheet of the job.	
	Job is tracked.	M1
	Job is untracked.	<i>invisible</i>
	<printer error>	
	No part of any job was lost.	S, R1
	The active job was lost. It was tracked; the next job is tracked.	S, R1, J3, M0, E1(aborted)
	The active job was lost. It was tracked; the next job is untracked.	S, R1, J4, M0, E1(aborted)
The active job was lost. It was untracked; the next job is tracked.	S, R1, J6, M0	
The active job was lost. It was untracked; the next job is untracked.	S, R1	
stopped	All problems corrected.	
	No jobs are queued.	I, R0
	Jobs are queued.	P, R0
	The reported problem is fixed, but another problem still exists.	
	CreateJob or CreateEnhancedJob or create non-UPnP tracked job	
	JobIdList is empty.	J1, M0
	JobIdList is not empty.	J2
Create untracked job.		
		<i>invisible</i>

696 **Table 5.: Transition Actions Used in Table 4**

	Variable(s) affected		
	Label	New variable value(s)	Action Descriptions
Printer State	I	idle	Printer enters idle state.
	P	processing	Printer enters processing state.
	S	stopped	Printer enters stopped state.
PrinterState Reasons	R0	none	Printer is operating normally, there are no problems to report.
	R1	<reason>	Old value was 'none'. New value is the reason the printer is in the current PrinterState (§ 2.6.7)
	R2	<new reason>	Old value was something other than 'none'. New value is still not 'none', but is different from old value.
JobIdList, JobId	J0	JobIdList ← {} JobId ← 0	New list value is empty.
	J1	JobIdList ← {id ₁ } JobId ← id ₁	New list contains single job
	J2	JobIdList ← {...,id _n ,id _{n+1} } OR {...,id _i ,id _{n+1} ,id _{i+1} ,..., id _n } <no change to JobId>	Old list may or may not have been empty. New list has same contents as old list <i>plus</i> one new job added. This job will normally be added at the end, but implementations are not required to do so.
	J3	JobIdList ← {id ₂ ,...} JobId ← id ₂	Old list had at least two jobs. New list has same content <i>except</i> first job was removed. JobId is set to the new first element in JobIdList.
	J4	JobIdList ← {id ₂ ,...} JobId ← 0	Old list had at least two jobs. New list has same content <i>except</i> first job was removed. JobId is set to '0' since the new first element in JobIdList is not the active job.
	J5	JobIdList ← {...,id _{i-1} ,id _{i+1} ,...} <no change to JobId>	Old list had at least two jobs. New list has same contents as old <i>except</i> the i th job, where i > 1, has been removed.
	J6	<no change to JobIdList> JobId ← id ₁	JobIdList is unchanged. JobId is set to the first element in JobIdList.
JobMediaSheets Completed	M0	'-1' or '0'	The value is '-1' if the printer never tracks this sheet count or if the current value is unknown. Otherwise, it is set to '0'.
	M1	'-1' or new Value=old Value+1	If the printer tracks sheet count for the active job, the value is incremented. Otherwise, the value is '-1', signifying unknown.
	M2	'-1' or known final value for job	'-1' if the printer does not know final sheet count. Actual sheet count if it is known. <i>Specifically, it could be '0' if the printer knows it never produced a sheet of paper for this job, even if the printer does not normally count sheets.</i>
JobEndState	E0	{}	JobEndState is initialized to the empty list.
	E1(T)	{ id ₁ , JobName_of_id ₁ , JobOriginatingUserName_of_id ₁ , M2, T}	The active job (first element in JobIdList) was terminated. T indicates the termination condition: one of 'successful', 'canceled' or 'aborted'.
	E2(T)	{ id _i , JobName_of_id _i , JobOriginatingUserName_of_id _i , M2, T}	The job in i th position (i > 1)of JobIdList was terminated. T indicates the termination condition: either 'canceled' or 'aborted'.

697

698 **2.8. Actions**

699 Immediately following this table is detailed information about these actions, including short descriptions of the
700 actions, the effects of the actions on state variables, and error codes defined by the actions.

701 **Table 6: Actions**

Name	Req. or Opt. ¹
<i>CreateJob</i>	<i>R</i>
<i>CancelJob</i>	<i>R</i>
<i>GetPrinterAttributes</i>	<i>R</i>
<i>GetJobAttributes</i>	<i>R</i>
<i>Non-standard actions implemented by a UPnP vendor go here.</i>	<i>X</i>

702 ¹ R = REQUIRED, O = Optional, X = Non-standard.

703 Note: the error codes are derived from IPP status codes as follows (see [MODEL] for the detailed definition of each
704 error code):

705 (Client Error minus 400₁₆) convert to decimal + 10 + 700
706 (Server Error minus 400₁₆) convert to decimal + 60 + 700

707

708 Error codes are returned in the <SOAP:Fault> element. A vendor MAY subset or extend these error codes, first by
709 supporting additional IPP error codes defined [MODEL] in the UPnP 700 range, and then by supporting private
710 error codes in the UPnP 800 range, if no suitable IPP error code exists.

711 **2.8.1. CreateJob**

712 *This action is the first step in submitting a job to the printer. The Printer returns a unique JobId to identify the job*
713 *for this service. The Printer generates the JobId in an implementation-defined manner. The Printer MUST return*
714 *values in the range 1 to 2³¹-1; 0 and negative values are invalid values to be returned as a result of a CreateJob*
715 *action. Furthermore, the Printer SHOULD NOT re-use values recently assigned, since UCPs would confuse such*
716 *jobs with older jobs.*

717 *The <allowedValueList> element of the Service Description indicates the values of the parameters that the Print*
718 *Service instance (Printer) supports (see section 2.3). The Printer performs the following validation in the indicated*
719 *order:*

- 720 1. *If the DocumentFormat is not supported, the Printer MUST reject the request and return the*
721 *ClientErrorDocumentFormatNotSupported (720) error code.*
- 722 2. *If the client (UCP) supplies input parameters that are unsupported or their values are unsupported (except*
723 *DocumentFormat), the Printer (1) MUST accept the CreateJob request, (2) MUST ignore or substitute*
724 *supported values, respectively, and (3) MUST print the job. This behavior corresponds to the 'false' or*
725 *omitted value of the IPP "ipp-attribute-fidelity" operation attribute. However, unlike IPP, the Printer*
726 *does not return any indication that attributes are being ignored or that values are being substituted.*
- 727 3. *If a client (UCP) supplies a conflicting combination of MediaSize and MediaType (or any other set of IN*
728 *parameters), the Printer MUST accept the CreateJob request, (2) MUST ignore or substitute the*
729 *conflicting values, and (3) MUST print the job. Whether or not a Printer can detect combinations of*
730 *different parameter values that are not supported, such as combinations of MediaType and MediaSize*
731 *values that are not supported, is IMPLEMENTATION-DEPENDENT. If an implementation does detect*
732 *combinations that are not supported, it substitutes values for one or more parameters to give a*
733 *combination that is supported.*

734 *The client (UCP) MUST send print data to the print service via a separate HTTP Post operation to the DataSink*
 735 *URL (see section 2.8.5) returned by the Printer in the CreateJob action response.*

736 **2.8.1.1. Arguments**

737 **Table 7: Arguments for CreateJob**

Argument	Direction	relatedStateVariable
<i>JobName</i>	<i>IN</i>	<i>JobName</i>
<i>JobOriginatingUserName</i>	<i>IN</i>	<i>JobOriginatingUserName</i>
<i>DocumentFormat</i>	<i>IN</i>	<i>DocumentFormat</i>
<i>Copies</i>	<i>IN</i>	<i>Copies</i>
<i>Sides</i>	<i>IN</i>	<i>Sides</i>
<i>NumberUp</i>	<i>IN</i>	<i>NumberUp</i>
<i>OrientationRequested</i>	<i>IN</i>	<i>OrientationRequested</i>
<i>MediaSize</i>	<i>IN</i>	<i>MediaSize</i>
<i>MediaType</i>	<i>IN</i>	<i>MediaType</i>
<i>PrintQuality</i>	<i>IN</i>	<i>PrintQuality</i>
<i>JobId</i>	<i>OUT</i>	<i>JobId</i>
<i>DataSink</i>	<i>OUT</i>	<i>DataSink</i>

738 Section 2.5 describes the CreateJob action IN/OUT argument's related state variables. The State Variable Table
 739 provides a description and data type as well as the allowed and default values.

740 **2.8.1.2. Errors**

errorCode	errorDescription	Description
Codes 401, 402, 403, 501, 600-99 from the table Common Action Error Codes (below)	See the table Common Action Error Codes (below)	See the table Common Action Error Codes (below)
<i>720</i>	<i>ClientErrorFormatNotSupported</i>	<i>The supplied DocumentFormat parameter value is not supported by the Printer object. The Printer object MUST return this status code, even if there are other parameters that are not supported as well, since this error is a bigger problem than with other input parameters.</i>
<i>760</i>	<i>ServerErrorInternalError</i>	<i>The Printer encountered an unexpected condition that prevented it from fulfilling the request. This error differs from "server-error-temporary-error" in that it implies a more permanent type of internal error.</i>
<i>765</i>	<i>ServerErrorTemporaryError</i>	<i>A temporary error that occurs while the printer processes the action. The client MAY try the unmodified request again at some later point in time with an expectation that the temporary internal error condition MAY have been cleared. If there is a more specific 6xx errors defined that applies to a temporary error, such as disk full, that code SHOULD be used.</i>

741 **2.8.2. CancelJob**

742 *This operation allows a client to cancel a print job from the time the job is created up to the time it is completed,*
 743 *cancelled or aborted.*

744 **2.8.2.1. Arguments**

745 **Table 8: Arguments for *CancelJob***

Argument	Direction	relatedStateVariable
<i>JobId</i>	<i>IN</i>	<i>JobId</i>

746

747 **2.8.2.2. Errors**

errorCode	errorDescription	Description
Codes 401, 402, 403, 501, 600-99 from the table Common Action Error Codes (below)	See the table Common Action Error Codes (below)	See the table Common Action Error Codes (below)
<i>716</i>	<i>ClientErrorNotFound</i>	<i>The printer has not found a job matching the JobId parameter (including when the parameter was not in the range: 1 to 2³¹-1).</i>
<i>760</i>	<i>ServerErrorInternalError</i>	<i>The Printer encountered an unexpected condition that prevented it from fulfilling the request. This error differs from "server-error-temporary-error" in that it implies a more permanent type of internal error.</i>
<i>765</i>	<i>ServerErrorTemporaryError</i>	<i>A temporary error that occurs while the printer processes the action. The client MAY try the unmodified request again at some later point in time with an expectation that the temporary internal error condition MAY have been cleared. If there is a more specific 6xx errors defined that applies to a temporary error, such as disk full, that code SHOULD be used.</i>

748 **2.8.2.3. Effect on State**

749 *The specified job with a JobId from 1 to 2³¹-1 is removed from the JobIdList. If the job was the current job (i.e.,*
 750 *JobId specified the current job), then JobId is set according to the transition actions described in Section 2.7.2.*

751 **2.8.3. GetPrinterAttributes**

752 The GetPrinterAttributes action allows a client (UCP) to determine the state of the printer and values of certain state
 753 variables that represent Printer attributes. In particular, the UCP can determine the number of pending jobs. The
 754 UCP can also determine the state of the Print Service, and which job, if any, is the current job.

755 Note: The GetPrinterAttributes action does not allow a client to discover the supported values of standard attributes.
 756 The client can discover what is supported from the <allowedValueList> element in the Service Description (see
 757 section 2.5). Neither does the GetPrinterAttributes action allow a client to discover vendor added attributes.
 758 Vendors MUST define their own private actions to return such additional attributes.

759 **2.8.3.1. Arguments**760 **Table 9: Arguments for *GetPrinterAttributes***

Argument	Direction	relatedStateVariable
<i>PrinterState</i>	<i>OUT</i>	<i>PrinterState</i>
<i>PrinterStateReasons</i>	<i>OUT</i>	<i>PrinterStateReasons</i>
<i>JobIdList</i>	<i>OUT</i>	<i>JobIdList</i>
<i>JobId</i>	<i>OUT</i>	<i>JobId</i>

761

762 **2.8.3.2. Errors**

errorCode	errorDescription	Description
Codes 401, 402, 403, 501, 600-99 from the table Common Action Error Codes (below)	See the table Common Action Error Codes (below)	See the table Common Action Error Codes (below)
<i>760</i>	<i>ServerErrorInternalError</i>	<i>The Printer encountered an unexpected condition that prevented it from fulfilling the request. This error differs from "server-error-temporary-error" in that it implies a more permanent type of internal error.</i>
<i>765</i>	<i>ServerErrorTemporaryError</i>	<i>A temporary error that occurs while the printer processes the action. The client MAY try the unmodified request again at some later point in time with an expectation that the temporary internal error condition MAY have been cleared. If there is a more specific 6xx errors defined that applies to a temporary error, such as disk full, that code SHOULD be used.</i>

763 **2.8.4. GetJobAttributes**

764 *The GetJobAttributes action allows a client (UCP) to determine some of the values of job-related variables of the*
765 *specified job with a JobId from 1 to 2³¹-1. Only active and queued jobs can be queried since only these jobs are*
766 *maintained in the JobIdList variable. These variables allow end users to identify their job (i.e., "JobName",*
767 *"JobOriginatingUserName"). Other information can be derived from the GetJobAttributes action.*

768 *If the specified job is found, its parameters are returned whether the job is active or queued. If the specified job is*
769 *not found, the ClientErrorNotFound (716) is returned. Any job not found either never existed or has reached its*
770 *terminating state (i.e., completed, cancelled, aborted) and is no longer known to the Print Service. If the value of*
771 *JobMediaSheetsCompleted is greater than 0, the referenced job is active and the printer has physically completed*
772 *printing and stacking the number of media sheets indicated. If the value of JobMediaSheetsCompleted is 0 or -1,*
773 *the client can determine whether the referenced job is active according to whether it is the first entry in JobIdList.*
774 *The value of JobIdList can be retrieved either from its most recent event value or from the action*
775 *GetPrinterAttributes.*

776 **2.8.4.1. Arguments**777 **Table 10: Arguments for *GetJobAttributes***

Argument	Direction	relatedStateVariable
<i>JobId</i>	<i>IN</i>	<i>JobId</i>
<i>JobName</i>	<i>OUT</i>	<i>JobName</i>
<i>JobOriginatingUserName</i>	<i>OUT</i>	<i>JobOriginatingUserName</i>
<i>JobMediaSheetsCompleted</i>	<i>OUT</i>	<i>JobMediaSheetsCompleted</i>

778 **2.8.4.2. Errors**

errorCode	errorDescription	Description
Codes 401, 402, 403, 501, 600-99 from the table Common Action Error Codes (below)	See the table Common Action Error Codes (below)	See the table Common Action Error Codes (below)
<i>716</i>	<i>ClientErrorNotFound</i>	<i>The printer has not found a job matching the JobId parameter (including when the parameter was not in the range: 1 to 2³¹-1).</i>
<i>760</i>	<i>ServerErrorInternalError</i>	<i>The Printer encountered an unexpected condition that prevented it from fulfilling the request. This error differs from "server-error-temporary-error" in that it implies a more permanent type of internal error.</i>
<i>765</i>	<i>ServerErrorTemporaryError</i>	<i>A temporary error that occurs while the printer processes the action. The client MAY try the unmodified request again at some later point in time with an expectation that the temporary internal error condition MAY have been cleared. If there is a more specific 6xx errors defined that applies to a temporary error, such as disk full, that code SHOULD be used.</i>

779

780 **2.8.5. HTTP Post**

781 The client (UCP) sends the print data using an HTTP [HTTP] Post operation (with chunking if desired), to the URL
 782 returned as the DataSink output parameter of the CreateJob action. Having received this DataSink URL in the
 783 CreateJob response, the client MUST then open a connection to the device using the URL and send the data.

784 The client MUST open the data connection on the DataSink URL within 30 seconds after receiving the CreateJob
 785 response. Otherwise, the printer MUST time out, discard jobs for which no data has been received, and remove its
 786 JobId from the JobIdList variable. If no data at all is received for a job then the Printer SHOULD delete the job
 787 after a wait of at least 30 seconds and remove its JobId from the JobIdList variable. If data has been received for a
 788 job but a subsequent chunked HTTP POST operation does not arrive for an implementation-defined period of time
 789 (at least 30 seconds) then the data received so far is printed and the job completes as usual. If the Printer receives
 790 an HTTP Post for the DataSink URL after the timeout period, the Printer returns the HTTP 408 (Request Timeout)
 791 status code, if the job still exists, otherwise, the HTTP 404 (Not Found) status code.

792 If the Printer accepts the CreateJob action, but subsequently cannot accept the HTTP Post (because it is too busy or
 793 is accepting another job), the Printer MUST reject the HTTP Post and return the HTTP 503 (Service Unavailable).
 794 The Printer SHOULD reset the timer to 30 seconds or some other implementation-specific value and SHOULD
 795 return that value in the Retry-After HTTP header in the error response. The printer SHOULD ensure that the Retry-
 796 After value is less than the maximum amount of time that the device will timeout. If the Printer does not return

797 Retry-After header, the HTTP spec [HTTP] says that the UCP assumes an HTTP 500 error (internal server error)
798 and no retry is allowed and the printer aborts the job.

799 An event will be sent to the client whenever the JobId is removed from the JobIdList.

800 The URL MUST be a valid HTTP URL [HTTP]. The Printer MUST support HTTP/1.1 chunking [HTTP] for the
801 Post operation. The client MUST send the DocumentFormat MIME Media Type value in the HTTP Content-Type
802 header (or the 'unknown' special value, if the client doesn't know the actual document format - see section 2.6.16).

803 2.8.6. Non-Standard Actions Implemented by a UPnP Vendor

804 To facilitate certification, non-standard actions implemented by UPnP vendors SHOULD be included in this service
805 template. The UPnP Device Architecture lists naming requirements for non-standard actions (see the section on
806 Description).

807 2.8.7. Common Error Codes

808 The following table lists error codes common to actions for this service type. If an action results in multiple errors,
809 the most specific error SHOULD be returned.

810 **Table 11: Common Error Codes**

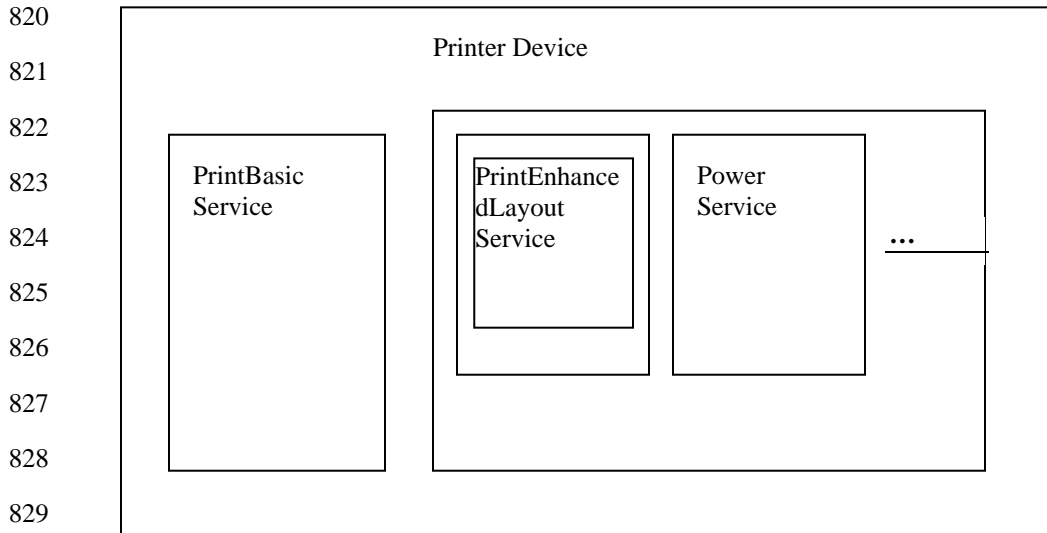
errorCode	errorDescription	Description
401	Invalid Action	See UPnP Device Architecture section on Control.
402	Invalid Args	See UPnP Device Architecture section on Control.
404	Invalid Var	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
600-699	TBD	Common action errors. Defined by UPnP Forum Technical Committee.
701-799		Common action errors defined by the UPnP Forum working committees.
800-899	TBD	<i>(Specified by UPnP vendor.)</i>

811 2.9. Theory of Operation

812 The UPnP Printer device ('printer') has one REQUIRED service called PrintBasic Service. Optional services MAY
813 include the PrintEnhancedLayout Service or basic power functions as illustrated below.

814 The model presented is very simple, it is intended to allow a user to send a job to a printer, be informed when it has
815 started printing and when it has finished printing. In addition, a user can cancel a previously submitted job. Also a
816 client (UCP) can determine which CreateJob action parameter values a Print Service implementation supports using
817 the values returned in the <allowedValueList> element of the Service Description.

818 No optional SST variables or actions are specified. The standard UPnP print service MUST support all the variables
819 and actions defined.



830

831 **Figure 1 - Printer Device and Services**832 **2.9.1. Jobs**

833 The print service's main task is to accept print jobs from clients, queue them up (if the printer is capable of handling
 834 more than one job at a time) and then print them. A job is identified by an integer, the JobId, which is allocated by
 835 the device. The [MODEL] describes the rules for JobId production (1 to 2**31-1). The JobId is returned by the
 836 CreateJob action.

837 The set of jobs that a printer has in its queue is exposed in a very simple way.

- 838
- 839
- 840 ○ The complete list of known jobs is made available as a state variable represented as CSV list (see section
 841 2.5.1.1) called **JobIdList**.
 - 842 ○ All waiting jobs appear in the **JobIdList** variable even those that the device has decided not to print for
 843 some reason (they are in the IPP 'pending' or 'pending-held' job state).
 - 844 ○ The order of jobs in the **JobIdList** variable indicates the order in which the jobs will be initiated.
 - 845 ○ The job that is actually printing at the moment (or for which the Print Service is stopped) is called the
 846 current job. If the current job is "Tracked" (see 2.2.2g), its job identifier is stored in the **JobId** Print
 847 Service state variable and that same JobId value is also the first **JobId** in the **JobIdList**. If there is no
 current job, i.e., there are no jobs, or all jobs are pending or held, or an Untracked Job (2.2.2h) is printing,
 the **JobId** is 0.
 - 848 ○ Once a job has been printed (or cancelled or aborted) it no longer appears in the **JobIdList**, whether or not
 849 the Printer has any other jobs to print.
 - 850 ○ When the Print Service has no tracked jobs to print, the **JobIdList** state variable is an empty string.

851 **2.9.2. Actions**

852 The following four actions are defined and MUST be supported by conforming PrintBasic Service implementations:

- 853
- CreateJob. This action is used to submit a job to the printer. The allocated JobId is returned.

- 854 o CancelJob. This can be used to cancel a job using the JobId.
- 855 o GetPrinterAttributes. This action can be used to query some of the printer attributes.
- 856 o GetJobAttributes. This action can be used to query some of the job attributes of a specified job.

857 **2.9.3. Events**

858 One of the primary goals of this specification is to allow a user to know when their print job has started and when it
 859 has finished. The UPnP eventing mechanism can be used for this purpose. There are five evented state variables,
 860 JobIdList, JobEndState, PrinterState, PrinterStateReasons, and JobMediaSheetsCompleted that MAY change
 861 whenever a job stops or starts. A client implementation SHOULD therefore subscribe to UPnP events from the
 862 print service in order to monitor the progress of a job. A UCP can determine when a particular job that it submitted
 863 has started printing by matching the first entry of the evented JobIdList variable with the JobId value returned to it
 864 by the CreateJob action. Similarly a UCP can determine that a job has completed, whether successful or not, by
 865 matching the JobId for that job with the first element of the evented JobEndState variable.

866 Four of the five evented variables are also available as OUT parameters of either GetPrinterAttributes or
 867 GetJobAttributes, so a UCP can obtain their values by polling. However, the JobEndState is not an OUT parameter
 868 of any action, so it is only available to a client by eventing, not by polling.

869 **2.9.4. Security**

870 In keeping with the lightweight approach to security taken by UPnP no security is defined by this specification.

871 If a vendor decides to include some form of security they are strongly encouraged to adopt the model that IPP uses –
 872 which is in fact fairly simple.

873 **2.9.5. Localization**

874 A UPnP printer is assumed to be operating within the locale of the user. No other localization mechanism is defined
 875 for the Print Service. The UCP (client) is expected to localize the well-known string values (that correspond to IPP
 876 keyword values) to the locale of its user. The UCP (client) is expected to convert the enum integer values to human
 877 readable string values in the locale of the user.

878 **2.9.6. IPP Data Type mapping to UPnP Data Types**

879 Basic IPP data types are transformed as follows.

880 **Table 12: Basic IPP data type mappings**

IPP Type (see [MODEL] for details)	UPnP Variable Type
Text	string
OctetString	bin.base64
Boolean	boolean
Integer	int
integer (0..2**31 -1)	i4 qualified by an <allowedValueRange>



881 The derived types in IPP are mapped onto the following UPnP data types.

882 **Table 13: Derived data type mappings**

IPP Type	UPnP Type	Notes (see [MODEL] for details)
name	string	A Name is a string with limited length. It is intended to have machine-readable meaning (as opposed to a simple text string).
keyword	string	A keyword is a name that has a limited set of allowed values in US-English represented as lowercase letters ("a" - "z"), digits ("0" - "9"), hyphen ("-"), dot ("."), and underscore ("_").
enum	string	An equivalent keyword string is used for each value using the symbol in IPP for each enum value, since the representation is XML.
uri	uri	A URI.
uriScheme	string	A string that specifies a URI scheme (http, ipp, etc.).
naturalLanguage	-	Not supported.
charset	-	Not supported.
mimeMediaType	string	A MIME type ('text/plain' for example).

883

884 **Table 14: Structured Data Type mapping**

IPP Type	UPnP equivalent
resolution	This is represented as a pair of integers <Attribute Name>X and <attribute Name>Y
IsetOf X	See the earlier discussion on arrays in section 2.5.1.1.

885

3. XML Service Description

```

886
887 <?xml version="1.0"?>
888 <scpd xmlns="urn:schemas-upnp-org:service-1-0">
889   <specVersion>
890     <major>1</major>
891     <minor>0</minor>
892   </specVersion>
893   <actionList>
894     <action>
895       <name>CreateJob</name>
896       <argumentList>
897         <argument>
898           <name>JobName</name>
899           <direction>in</direction>
900           <relatedStateVariable>JobName</relatedStateVariable>
901         </argument>
902         <argument>
903           <name>JobOriginatingUserName</name>
904           <direction>in</direction>
905           <relatedStateVariable>JobOriginatingUserName</relatedStateVariable>
906         </argument>
907         <argument>
908           <name>DocumentFormat</name>
909           <direction>in</direction>
910           <relatedStateVariable>DocumentFormat</relatedStateVariable>
911         </argument>
912         <argument>
913           <name>Copies</name>
914           <direction>in</direction>
915           <relatedStateVariable>Copies</relatedStateVariable>
916         </argument>
917         <argument>
918           <name>Sides</name>
919           <direction>in</direction>
920           <relatedStateVariable>Sides</relatedStateVariable>
921         </argument>
922         <argument>
923           <name>NumberUp</name>
924           <direction>in</direction>
925           <relatedStateVariable>NumberUp</relatedStateVariable>
926         </argument>
927         <argument>
928           <name>OrientationRequested</name>
929           <direction>in</direction>
930           <relatedStateVariable>OrientationRequested</relatedStateVariable>
931         </argument>
932         <argument>
933           <name>MediaSize</name>
934           <direction>in</direction>
935           <relatedStateVariable>MediaSize</relatedStateVariable>
936         </argument>
937         <argument>
938           <name>MediaType</name>

```

```

939     <direction>in</direction>
940     <relatedStateVariable>MediaType</relatedStateVariable>
941 </argument>
942 <argument>
943     <name>PrintQuality</name>
944     <direction>in</direction>
945     <relatedStateVariable>PrintQuality</relatedStateVariable>
946 </argument>
947 <argument>
948     <name>JobId</name>
949     <direction>out</direction>
950     <relatedStateVariable>JobId</relatedStateVariable>
951 </argument>
952 <argument>
953     <name>DataSink</name>
954     <direction>out</direction>
955     <relatedStateVariable>DataSink</relatedStateVariable>
956 </argument>
957 </argumentList>
958 </action>
959 <action>
960 <name>CancelJob</name>
961 <argumentList>
962 <argument>
963     <name>JobId</name>
964     <direction>in</direction>
965     <relatedStateVariable>JobId</relatedStateVariable>
966 </argument>
967 </argumentList>
968 </action>
969 <action>
970 <name>GetPrinterAttributes</name>
971 <argumentList>
972 <argument>
973     <name>PrinterState</name>
974     <direction>out</direction>
975     <relatedStateVariable>PrinterState</relatedStateVariable>
976 </argument>
977 <argument>
978     <name>PrinterStateReasons</name>
979     <direction>out</direction>
980     <relatedStateVariable>PrinterStateReasons</relatedStateVariable>
981 </argument>
982 <argument>
983     <name>JobIdList</name>
984     <direction>out</direction>
985     <relatedStateVariable>JobIdList</relatedStateVariable>
986 </argument>
987 <argument>
988     <name>JobId</name>
989     <direction>out</direction>
990     <relatedStateVariable>JobId</relatedStateVariable>
991 </argument>
992 </argumentList>
993 </action>

```

```

994     <action>
995     <name>GetJobAttributes</name>
996     <argumentList>
997     <argument>
998         <name>JobId</name>
999         <direction>in</direction>
1000        <relatedStateVariable>JobId</relatedStateVariable>
1001    </argument>
1002    <argument>
1003        <name>JobName</name>
1004        <direction>out</direction>
1005        <relatedStateVariable>JobName</relatedStateVariable>
1006    </argument>
1007    <argument>
1008        <name>JobOriginatingUserName</name>
1009        <direction>out</direction>
1010        <relatedStateVariable>JobOriginatingUserName</relatedStateVariable>
1011    </argument>
1012    <argument>
1013        <name>JobMediaSheetsCompleted</name>
1014        <direction>out</direction>
1015        <relatedStateVariable>JobMediaSheetsCompleted</relatedStateVariable>
1016    </argument>
1017    </argumentList>
1018    </action>
1019 </actionList>
1020 <serviceStateTable>
1021     <stateVariable sendEvents="no">
1022         <name>PrinterName</name>
1023         <dataType>string</dataType>
1024         <defaultValue></defaultValue>
1025     </stateVariable>
1026     <stateVariable sendEvents="no">
1027         <name>PrinterLocation</name>
1028         <dataType>string</dataType>
1029         <defaultValue></defaultValue>
1030     </stateVariable>
1031     <stateVariable sendEvents="no">
1032         <name>DeviceId</name>
1033         <dataType>string</dataType>
1034         <defaultValue></defaultValue>
1035     </stateVariable>
1036     <stateVariable sendEvents="yes">
1037         <name>PrinterState</name>
1038         <dataType>string</dataType>
1039         <defaultValue>idle</defaultValue>
1040         <allowedValueList>
1041             <allowedValue>idle</allowedValue>
1042             <allowedValue>processing</allowedValue>
1043             <allowedValue>stopped</allowedValue>
1044         </allowedValueList>
1045     </stateVariable>
1046     <stateVariable sendEvents="yes">
1047         <name>PrinterStateReasons</name>
1048         <dataType>string</dataType>

```

```

1049     <defaultValue>none</defaultValue>
1050     <allowedValueList>
1051         <allowedValue>none</allowedValue>
1052         <allowedValue>attention-required</allowedValue>
1053         <allowedValue>media-jam</allowedValue>
1054         <allowedValue>paused</allowedValue>
1055         <allowedValue>door-open</allowedValue>
1056         <allowedValue>media-low</allowedValue>
1057         <allowedValue>media-empty</allowedValue>
1058         <allowedValue>output-area-almost-full</allowedValue>
1059         <allowedValue>output-area-full</allowedValue>
1060         <allowedValue>marker-supply-low</allowedValue>
1061         <allowedValue>marker-supply-empty</allowedValue>
1062         <allowedValue>marker-failure</allowedValue>
1063         <allowedValue>media-change-request</allowedValue>
1064     </allowedValueList>
1065 </stateVariable>
1066 <stateVariable sendEvents="no">
1067     <name>XHTMLImageSupported</name>
1068     <dataType>string</dataType>
1069     <defaultValue>image/jpeg</defaultValue>
1070     <allowedValueList>
1071         <allowedValue>image/jpeg</allowedValue>
1072     </allowedValueList>
1073 </stateVariable>
1074 <stateVariable sendEvents="no">
1075     <name>ColorSupported</name>
1076     <dataType>boolean</dataType>
1077     <defaultValue></defaultValue>
1078 </stateVariable>
1079 <stateVariable sendEvents="yes">
1080     <name>JobIdList</name>
1081     <dataType>string</dataType>
1082     <defaultValue></defaultValue>
1083 </stateVariable>
1084 <stateVariable sendEvents="no">
1085     <name>JobId</name>
1086     <dataType>i4</dataType>
1087     <defaultValue>0</defaultValue>
1088     <allowedValueRange>
1089         <minimum>0</minimum>
1090         <maximum>2147483647</maximum>
1091         <step>1</step>
1092     </allowedValueRange>
1093 </stateVariable>
1094 <stateVariable sendEvents="yes">
1095     <name>JobEndState</name>
1096     <dataType>string</dataType>
1097     <defaultValue></defaultValue>
1098 </stateVariable>
1099 <stateVariable sendEvents="no">
1100     <name>JobName</name>
1101     <dataType>string</dataType>
1102     <defaultValue></defaultValue>
1103 </stateVariable>

```

```

1104 <stateVariable sendEvents="no">
1105   <name>JobOriginatingUserName</name>
1106   <dataType>string</dataType>
1107   <defaultValue></defaultValue>
1108 </stateVariable>
1109 <stateVariable sendEvents="no">
1110   <name>DocumentFormat</name>
1111   <dataType>string</dataType>
1112   <defaultValue>unknown</defaultValue>
1113   <allowedValueList>
1114     <allowedValue>unknown</allowedValue>
1115     <allowedValue>application/vnd.pwg-xhtml-print</allowedValue>
1116     <allowedValue>application/octet-stream</allowedValue>
1117     <allowedValue>text/plain</allowedValue>
1118     <allowedValue>text/plain;charset=utf-8</allowedValue>
1119     <allowedValue>application/postscript</allowedValue>
1120     <allowedValue>application/vnd.hp-PCL</allowedValue>
1121   </allowedValueList>
1122 </stateVariable>
1123 <stateVariable sendEvents="no">
1124   <name>Copies</name>
1125   <dataType>i4</dataType>
1126   <defaultValue>1</defaultValue>
1127   <allowedValueRange>
1128     <minimum>0</minimum>
1129     <maximum>2147483647</maximum>
1130     <step>1</step>
1131   </allowedValueRange>
1132 </stateVariable>
1133 <stateVariable sendEvents="no">
1134   <name>Sides</name>
1135   <dataType>string</dataType>
1136   <defaultValue>one-sided</defaultValue>
1137   <allowedValueList>
1138     <allowedValue>one-sided</allowedValue>
1139     <allowedValue>two-sided-long-edge</allowedValue>
1140     <allowedValue>two-sided-short-edge</allowedValue>
1141     <allowedValue>device-setting</allowedValue>
1142   </allowedValueList>
1143 </stateVariable>
1144 <stateVariable sendEvents="no">
1145   <name>NumberUp</name>
1146   <dataType>string</dataType>
1147   <defaultValue>1</defaultValue>
1148   <allowedValueList>
1149     <allowedValue>1</allowedValue>
1150     <allowedValue>2</allowedValue>
1151     <allowedValue>4</allowedValue>
1152     <allowedValue>device-setting</allowedValue>
1153   </allowedValueList>
1154 </stateVariable>
1155 <stateVariable sendEvents="no">
1156   <name>OrientationRequested</name>
1157   <dataType>string</dataType>
1158   <defaultValue>portrait</defaultValue>

```



```

1159     <allowedValueList>
1160         <allowedValue>portrait</allowedValue>
1161         <allowedValue>landscape</allowedValue>
1162         <allowedValue>reverse-landscape</allowedValue>
1163         <allowedValue>reverse-portrait</allowedValue>
1164         <allowedValue>device-setting</allowedValue>
1165     </allowedValueList>
1166 </stateVariable>
1167 <stateVariable sendEvents="no">
1168     <name>MediaSize</name>
1169     <dataType>string</dataType>
1170     <defaultValue></defaultValue>
1171     <allowedValueList>
1172         <allowedValue>na_letter_8.5x11in</allowedValue>
1173         <allowedValue>na_legal_8.5x14in</allowedValue>
1174         <allowedValue>iso_a4_210x297mm</allowedValue>
1175         <allowedValue>iso_c5_162x229mm</allowedValue>
1176         <allowedValue>iso_dl_110x220mm</allowedValue>
1177         <allowedValue>jis_b4_257x364mm</allowedValue>
1178         <allowedValue>device-setting</allowedValue>
1179     </allowedValueList>
1180 </stateVariable>
1181 <stateVariable sendEvents="no">
1182     <name>MediaType</name>
1183     <dataType>string</dataType>
1184     <defaultValue></defaultValue>
1185     <allowedValueList>
1186         <allowedValue>stationery</allowedValue>
1187         <allowedValue>transparency</allowedValue>
1188         <allowedValue>envelope</allowedValue>
1189         <allowedValue>labels</allowedValue>
1190         <allowedValue>photographic</allowedValue>
1191         <allowedValue>cardstock</allowedValue>
1192         <allowedValue>device-setting</allowedValue>
1193     </allowedValueList>
1194 </stateVariable>
1195 <stateVariable sendEvents="no">
1196     <name>PrintQuality</name>
1197     <dataType>string</dataType>
1198     <defaultValue>normal</defaultValue>
1199     <allowedValueList>
1200         <allowedValue>draft</allowedValue>
1201         <allowedValue>normal</allowedValue>
1202         <allowedValue>high</allowedValue>
1203         <allowedValue>device-setting</allowedValue>
1204     </allowedValueList>
1205 </stateVariable>
1206 <stateVariable sendEvents="no">
1207     <name>DataSink</name>
1208     <dataType>uri</dataType>
1209     <defaultValue></defaultValue>
1210 </stateVariable>
1211 <stateVariable sendEvents="yes">
1212     <name>JobMediaSheetsCompleted</name>
1213     <dataType>i4</dataType>

```

```
1214     <defaultValue>0</defaultValue>
1215     <allowedValueRange>
1216         <minimum>-1</minimum>
1217         <maximum>2147483647</maximum>
1218         <step>1</step>
1219     </allowedValueRange>
1220 </stateVariable>
1221 </serviceStateTable>
1222 </scpd>
```