
PrintEnhanced:1 Service Template Version 1.01

For UPnP™ Version 1.0

Status: Standardized DCP

Date: October 28, 2006

Document Revision: 1.01

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

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

© 2006 Contributing Members of the UPnP Forum. All Rights Reserved.

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

24 **Contents**

25 **1. OVERVIEW AND SCOPE.....6**

26 **2. SERVICE MODELING DEFINITIONS.....6**

27 2.1. SERVICE TYPE6

28 2.2. TERMINOLOGY AND NOTATIONS6

29 2.2.1. *Conformance Terminology*.....6

30 2.2.2. *Other Terminology*7

31 2.2.3. *Notation: Use of Quotation Marks*9

32 2.2.4. *Notation: Use of Asterisks in Action Names*.....9

33 2.3. REFERENCES9

34 2.4. DERIVED DATA TYPES.....10

35 2.4.1. *Comma Separated Value (CSV) Lists*10

36 2.4.2. *XML Content in UPnP Arguments and State Variables*.....11

37 2.5. NAMING CONVENTIONS12

38 2.6. STATE VARIABLES12

39 2.6.1. *The Printer’s Supported and Default Values*.....12

40 2.6.2. *The Distinguished Value*.....12

41 2.6.3. *PrintEnhanced:1 Service Variables*13

42 2.6.3.1. A_ARG_TYPE_CriticalAttribList16

43 2.6.3.2. A_ARG_TYPE_MediaList17

44 2.6.3.3. A_ARG_TYPE_PrinterAbortReason20

45 2.6.3.4. CharRepSupported20

46 2.6.3.5. ColorSupported21

47 2.6.3.6. ContentCompleteList.....21

48 2.6.3.7. Copies.....22

49 2.6.3.8. CriticalAttributesSupported.....22

50 2.6.3.9. DataSink25

51 2.6.3.10. DeviceId25

52 2.6.3.11. DocumentFormat26

53 2.6.3.12. DocumentUTF16Supported.....27

54 2.6.3.13. FullBleedSupported.....28

55 2.6.3.14. InternetConnectState29

56 2.6.3.15. JobAbortState29

57 2.6.3.16. JobEndState30

58 2.6.3.17. JobId31

59 2.6.3.18. JobIdList.....31

60 2.6.3.19. JobMediaSheetsCompleted31

61 2.6.3.20. JobName.....31

62 2.6.3.21. JobOriginatingUserName32

63 2.6.3.22. MediaSize.....32

64 2.6.3.23. MediaType33

65 2.6.3.24. NumberUp.....34

66 2.6.3.25. OrientationRequested35

67 2.6.3.26. PageMargins.....36

68 2.6.3.27. PrinterLocation.....36

69 2.6.3.28. PrinterName.....36

70 2.6.3.29. PrintQuality36

71 2.6.3.30. PrinterState37

72 2.6.3.31. PrinterStateReasons.....37

73 2.6.3.32. Sides39

74 2.6.3.33. SourceURI40

75 2.6.3.34. XHTMLImageSupported.....40

76 2.7. EVENTING AND MODERATION41

77 2.7.1. *Event Model*.....41

78	2.7.2. Synchronization of Evented Variables.....	42
79	2.8. ACTIONS.....	46
80	2.8.1. CancelJob.....	47
81	2.8.1.1. Arguments.....	47
82	2.8.1.2. Errors.....	47
83	2.8.1.3. Effect on State.....	47
84	2.8.2. CreateJob (deprecated).....	47
85	2.8.2.1. Arguments.....	48
86	2.8.2.2. Errors.....	48
87	2.8.3. CreateJobV2.....	49
88	2.8.3.1. Arguments.....	50
89	2.8.3.2. Errors.....	50
90	2.8.4. CreateURIJob.....	51
91	2.8.4.1. Arguments.....	52
92	2.8.4.2. Errors.....	53
93	2.8.5. GetJobAttributes.....	54
94	2.8.5.1. Arguments.....	54
95	2.8.5.2. Errors.....	54
96	2.8.6. GetMargins.....	55
97	2.8.6.1. Arguments.....	56
98	2.8.6.2. Errors.....	57
99	2.8.6.3. Effect of Action on State.....	57
100	2.8.7. GetMediaList.....	58
101	2.8.7.1. Arguments.....	58
102	2.8.7.2. Errors.....	58
103	2.8.7.3. Effect of Action on State.....	59
104	2.8.8. GetPrinterAttributes (deprecated).....	59
105	2.8.8.1. Arguments.....	59
106	2.8.8.2. Errors.....	60
107	2.8.9. GetPrinterAttributesV2.....	60
108	2.8.9.1. Arguments.....	60
109	2.8.9.2. Errors.....	60
110	2.8.9.3. Effect of Action on State.....	61
111	2.8.10. HTTP POST.....	61
112	2.8.11. HTTP GET.....	61
113	2.8.12. Error Codes.....	62
114	2.9. THEORY OF OPERATION.....	63
115	2.9.1. The Print Model.....	63
116	2.9.2. Jobs.....	63
117	2.9.3. Job Processing.....	64
118	2.9.3.1. Intent of a Print Job.....	64
119	2.9.3.2. Critical Attributes and the Intent of a Print Job.....	65
120	2.9.4. Side-by-side Images.....	67
121	2.9.5. Actions.....	68
122	2.9.6. Events.....	68
123	2.9.7. Security.....	68
124	2.9.8. Localization.....	68
125	2.9.9. IPP Data Type mapping to UPnP Data Types.....	69
126	2.9.10. Improving Output Consistency for XHTML-Print.....	70
127	3. XML SERVICE DESCRIPTION.....	71

128

129 **List of Tables**

130	Table 1: State Variables.....	13
-----	-------------------------------	----

131 Table 2: Values for *CriticalAttributesList*16

132 Table 3: allowedValueList for *A_ARG_TYPE_PrinterAbortReason*20

133 Table 4: allowedValueList for *CharRepSupported*.....21

134 Table 5: allowedValueList for *ColorSupported*.....21

135 Table 6: allowedValueList for *CriticalAttributesSupported*24

136 Table 7: allowedValueList for *DocumentFormat*27

137 Table 8: allowedValueList for *DocumentUTF16Supported*28

138 Table 9: allowedValueList for *FullBleedSupported*.....28

139 Table 10: allowedValueList for *InternetConnectState*29

140 Table 11: allowedValueList for *job-abort-reason*.....29

141 Table 12: allowedValueList for *MediaSize*33

142 Table 13: allowedValueList for *MediaType*.....34

143 Table 14: allowedValueList for *NumberUp*.....35

144 Table 15: allowedValueList for *OrientationRequested*.....36

145 Table 16: allowedValueList for *PrintQuality*.....37

146 Table 17: allowedValueList for *PrinterState*37

147 Table 18: allowedValueList for *PrinterStateReasons*39

148 Table 19: allowedValueList for *Sides*40

149 Table 20: allowedValueList for *XHTMLImageSupported*40

150 Table 21: Event Moderation.....41

151 Table 22: Synchronization of Evented Variables43

152 Table 23: Transition Actions Used in Table 19.....45

153 Table 24: Actions46

154 Table 25: Arguments for *CancelJob*47

155 Table 26: Arguments for *CreateJob*.....48

156 Table 27: Arguments for *CreateJobV2*50

157 Table 28: Arguments for *CreateURIJob*52

158 Table 29: Arguments for *GetJobAttributes*54

159 Table 30: Arguments for *GetMargins*56

160 Table 31: Arguments for *GetMediaList*.....58

161 Table 32: Arguments for *GetPrinterAttributes*.....59

162	Table 33: Arguments for <i>GetPrinterAttributesV2</i>	60
163	Table 34: Error Codes	62
164	Table 35: Precedence of Production and Layout Job Attributes	65
165	Table 36: Basic IPP data type mappings	69
166	Table 37: Derived data type mappings	69
167	Table 38: Structured Data Type mapping.....	69
168		

169

170 1. Overview and Scope

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

172 This service type has been defined as a superset of PrintBasic:1.

173 This service-type enables the following functions:

- 174 • *Printing using both “push” and “pull” models:*
 - 175 • Control Point MAY push the print document using HTTP POST.
 - 176 • Control Point MAY provide a URI and request the print service to pull the print document from that
 - 177 location using HTTP GET.
- 178 • *Enhanced Layout Printing: Allows precise positioning and size capability, box properties, EXIF file format,*
- 179 *etc. (for more details, see Enhanced Layout Extension Conformance, section 2.4 of XHTML-Print [XHTML-*
- 180 *PRINT] and section 2.1 of CSS Print Profile [CSSPP]).*
- 181 • *Flexible Job Control with respect to User Intents: CreateJobV2 and CreateURIJob allow the Control Point to*
- 182 *request a job be printed either in a ‘best effort’ manner or if and only if all “critical” aspects of the job request*
- 183 *can be honored by the Printer.*

184

185 1.1. Change Log

Spec Version - Date	Changes from Previous Version
v1-050504	Initial version
v1-20061028	Fixed ambiguity in the description of the A_ARG_TYP_MediaList state variable.

186

187

188 2. Service Modeling Definitions

189 2.1. Service Type

190 A service that is compliant with this specification is identified with the following service type: **urn:schemas-upnp-**

191 **org:service:PrintEnhanced:1.**

192 2.2. Terminology and Notations

193 This section defines terms that are used throughout this specification. These terms are always capitalized in order to

194 indicate that they have the meaning defined in this section.

195 2.2.1. Conformance Terminology

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

- 197 a) **MUST** - This word, or the term "REQUIRED", mean that the definition is an absolute requirement of the
- 198 specification.

- 199 b) **MUST NOT** - This phrase means that the definition is an absolute prohibition of the specification.
- 200 c) **SHOULD** - This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in
201 particular circumstances to ignore a particular item, but the full implications **MUST** be understood and
202 carefully weighed before choosing a different course.
- 203 d) **SHOULD NOT** - This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid
204 reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full
205 implications should be understood and the case carefully weighed before implementing any behavior
206 described with this label.
- 207 e) **MAY** - This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may
208 choose to include the item because a particular marketplace requires it or because the vendor feels that it
209 enhances the product while another vendor may omit the same item. An implementation which does not
210 include a particular option **MUST** be prepared to interoperate with another implementation which does
211 include the option, though perhaps with reduced functionality. An implementation which does include a
212 particular option **MUST** be prepared to interoperate with another implementation which does not include
213 the option

214 2.2.2. Other Terminology

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

- 218 a) **Comma Separated Value (CSV)** - a variable that contains multiple string values separated by the US-ASCII
219 **COMMA** (',') character (see section 2.4.1).
- 220 b) **Content Complete** — A job is said to be Content Complete when the Printer holds all information
221 necessary to finish printing the job—it will not need to access any more external data. For example, an
222 XHTML-Print job **MUST** satisfy two conditions to be content complete. First, the Printer will have
223 fetched the complete source document object and all objects that are referenced either directly by URIs in
224 the source or indirectly by URIs in previously referenced objects. Second, all remaining unprinted content
225 from these objects is locally buffered by the Printer and will not be released until it has been printed or
226 canceled.
- 227 c) **Critical Attribute** – a print job attribute whose value the Printer can determine at print time and that a
228 Control Point is allowed to declare as critical to the successful completion of a print job. Specifically,
229 when a Control Point indicates that a particular attribute is Critical and the Printer is unable to satisfy the
230 requested value for that attribute at print time, the Printer **MUST** abort the job. The value of the Critical
231 Attribute can either be directly detectable by the Printer or it **MAY** be supplied by implementation-defined
232 means that are outside the scope of this specification, such as a user-controlled front panel setting. See
233 section 2.9.3.2
- 234 d) **Deprecated** – A construct which is deprecated is targeted for obsolescence from the PrintBasic:1 service
235 specification. It **SHOULD NOT** be used by Control Points for new applications or extended functionality.
236 Since PrintEnhanced:1 is a superset of PrintBasic:1, the deprecated construct **MUST** be supported by
237 Printers conforming to the PrintEnhanced:1 service.
- 238 e) **Distinguished Value** – a special value defined by this specification for some action IN parameters. Use of
239 Distinguished Value IN parameter allows a Page Description Language (PDL) Data Stream corresponding
240 value to take effect when it would normally be overridden by the IN parameter. In the case where the
241 Distinguished Value is absent in the PDL data stream and the IN parameter value is specified as 'device-
242 setting', the Service uses its <defaultValue> value for the IN parameter. See section 2.6.2.

- 243 f) Full Bleed – A method of printing allowing the entire surface of the medium to be marked. *I.e.*, there is no
244 white (or, more accurately, media-colored) edge around the printed content. Within the context of this
245 Service, its meaning is restricted to include only image content and simple superimposed annotation. That
246 is, print content containing arbitrary text and objects other than images is not considered full bleed, even
247 though that print content might not have an edge.
248 NOTE: In general, media registration and skew tolerances imply that a printer will need to do some special
249 processing to achieve full-bleed output. One common technique is to scale the image up to a size slightly
250 larger than the medium, implying some of the edge pixels will be lost.
- 251 g) Impression – The print content affixed to one surface of a sheet of print medium. When printing only
252 single-sided, there is one impression per physical page, regardless of whether the *n* in n-up is 1 or greater
253 than 1. When duplex printing (printing on both sides of the medium), there are two impressions per
254 physical page, regardless of whether the *n* in n-up is 1 or greater than 1.
- 255 h) Layout Job Attributes - job attributes that are inherent to the integrity of the print content and are not
256 overridden by supplying corresponding IN parameters when submitting the job (see section 2.9.3.1.1).
257 (*E.g.*, page orientation.)
- 258 i) N-up – A method of printing where, when *n* is greater than 1, multiple logical pages are reduced in size
259 and printed on a single medium surface. For example, a 4-up printout has 4 logical pages imprinted on one
260 side of a single page at approximately ¼ of their usual size.
- 261 j) Non-printable Area – As defined by the CSS3 Paged Media Module [CSS3_PM], the area around the edge
262 of the physical medium that the printer is not capable of marking. In this specification it identifies the area
263 around the edge of the physical medium where individual pixels cannot be reliably positioned. For
264 example, a Printer may print in this area when using special techniques such as full-bleed processing, but
265 be unable to reliably place text in this area.
- 266 k) PDL – the Page Description Language. Any of numerous mechanisms to define document content and
267 formatting. Examples include XHTML and CSS, PostScript, PCL, etc.
- 268 l) PDL Data Stream - the stream of data to be printed as represented in a specified document format.
- 269 m) Print Service (or Printer) - the UPnP entity that accepts actions from Control Point (clients), returns
270 responses, sends events, and generates printed output.
- 271 n) Production Job Attributes - job attributes that are not inherent to the integrity of the print content, and so
272 the Control Point MAY override the PDL Data Stream instructions, if any, by supplying corresponding IN
273 parameters when submitting the job (see section 2.9.3.1.1). (*E.g.*, number of copies.)
- 274 o) Tracked Job - a UPnP or non-UPnP job that is visible to a UPnP Control Point; *i.e.*, a print job which has a
275 JobId and appears in the JobIdList, and on which the Control Point can perform any of the Job operations
276 defined in this document.
- 277 p) Untracked Job - a non-UPnP job that is not visible to a UPnP Control Point; *i.e.*, it does not have a JobId
278 and does not appear in the JobIdList, and on which the Control Point cannot perform any of the Job
279 operations defined in this document.
- 280 q) Well-balanced XML - text that is or could be the content of a single element in a valid XML document.
281 Syntactically, well-balanced XML matches the *content* production as defined in [XML] Section 3.1, “Start-
282 tags, End-tags, and Empty Element Tags.” Well-balanced XML could be well-formed XML, though it
283 usually will not be. Also, well-balanced XML will usually contain XML markup, though it is not required
284 to.

285 **2.2.3. Notation: Use of Quotation Marks**

286 Throughout this document, single quotes (‘) are used around literal string and integer values in running text, but not
287 in Tables. The single quotes are not part of the values. Double quotes (“”) are used around words in running text to
288 indicate special English meanings. Variable names, parameter names, and action names are not quoted.

289 **2.2.4. Notation: Use of Asterisks in Action Names**

290 PrintEnhanced:1 defines three separate actions for creating a print job—CreateJob, CreateJobV2 and CreateURIJob.
291 Some job processing behaviors depend on which action created the job, but many behaviors are common to two or
292 all three Create actions. To avoid many name repetitions, we will use the following shorthand notations when
293 referring collectively to two or more Create actions:

294 Create* —all three actions
295 CreateJob* — CreateJob and CreateJobV2
296

297 **2.3. References**

298 This section lists the references that this document refers to and the tag inside square brackets that is used to refer to
299 each such reference:

- 300 [DEVICE] - UPnP Device Architecture, version 1.0 and UPnP Vendor’s Implementation Guide. Available at:
301 <http://www.upnp.org/standardizeddcp/docs/documents/upnpresource20040907.zip>
- 302 [HTTP] - RFC 2616 “Hypertext Transfer Protocol -- HTTP/1.1”, R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L.
303 Masinter, P. Leach, T. Berners-Lee. June 1999. (Format: TXT=422317, PS=5529857, PDF=550558 bytes)
304 (Obsoletes RFC2068) (Updated by RFC2817) (Status: DRAFT STANDARD) Available at: [ftp://ftp.rfc-](ftp://ftp.rfc-editor.org/in-notes/rfc2616.txt)
305 [editor.org/in-notes/rfc2616.txt](ftp://ftp.rfc-editor.org/in-notes/rfc2616.txt)
- 306 [MODEL] - RFC 2566 “Internet Printing Protocol/1.0 Model and Semantics”, March 1999 and RFC 2911 “Internet
307 Printing Protocol/1.1 Model and Semantics”, September 2000, standards. Available at: <http://www.ietf.org>
- 308 [PWG5101.1] *IEEE-ISTO 5101.1-2001 The Printer Working Group Standard for Media Standardized Names 26*
309 *February 2002.* Available at: <ftp://ftp.pwg.org/pub/pwg/standards/pwg5101.1.pdf>, [.doc](ftp://ftp.pwg.org/pub/pwg/standards/pwg5101.1.doc), [.rtf](ftp://ftp.pwg.org/pub/pwg/standards/pwg5101.1.rtf)
- 310 [PWG5101.2] *IEEE-ISTO 5101.2-2004 The Printer Working Group Standard for Repertoires Supported Element 1*
311 *February 2004.* Available at: <ftp://ftp.pwg.org/pub/pwg/candidates/cs-crrepsup10-20040201-5101.2.pdf>
- 312 [XHTML-PRINT] – XHTML-Print, W3C Candidate Recommendation, 20 January 2004. Available at:
313 <http://www.w3.org/TR/2004/CR-xhtml-print-20040120>
- 314 [CSSPP] – CSS Print Profile, W3C Candidate Recommendation, 25 January 2004. Available at:
315 <http://www.w3.org/TR/2004/CR-css-print-20040225>
- 316 [CSS3_PM] - CSS3 Paged Media Module, W3C Candidate Recommendation, 25 February 2004. Available at:
317 <http://www.w3.org/TR/2004/CR-css3-page-20040225>
- 318 [MULTIPLEXED] – RFC 3391 "The MIME Application/Vnd.pwg-multiplexed Content-type", R. Herriot.
319 December 2002. (Status: INFORMATIONAL) Available at: <ftp://ftp.rfc-editor.org/in-notes/rfc3391.txt>
320
- 321 [XML] – [Extensible Markup Language \(XML\) 1.0 \(Second Edition\)](http://www.w3.org/TR/2000/REC-xml-20001006/), T. Bray, J.Paoli, C. M. Sperberg-McQueen, E
322 Maler, eds. W3C Recommendations, 6 October 2000.
- 323 [XPCSSGUIDE] – XHTML-PRINT/CSS-Print Profile Guidelines for PrintEnhanced:1, January 2005. Available at:
324 http://www.upnp.org/standardizeddcp/docs/documents/PrintEnhanced1_guidelines_v1_050504.pdf

325 2.4. Derived Data Types

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

327 2.4.1. Comma Separated Value (CSV) Lists

328 The UPnP PrintEnhanced:1 Service uses variables that represent lists, or one-dimensional arrays, of values.
329 Examples include the supported sets of document formats and media stock. The UPnP Device Architecture,
330 Version 1.0 [DEVICE], does not provide for either an array type or a list type, so a list type is defined here. Lists
331 MAY either be homogeneous (all values are the same type) or heterogeneous (values of different types are
332 allowed). The data type of a homogeneous list is *string (CSV x)*, where *x* is the type of the individual values. The
333 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
334 element values. If the number of elements in the heterogeneous list is too large to show each type individually, that
335 variable type is represented as *string (CSV heterogeneous)*, and the variable description includes additional
336 information as to the expected sequence of values appearing in the list and their corresponding types.

- 337 ○ A list is represented as a UPnP String type.
- 338 ○ Values within a list are separated by commas.
- 339 ○ Only three value types are used as CSV elements in this specification—string, integer and boolean.
- 340 ○ Integer values are represented in CSVs with the same syntax as the int data type specified in [DEVICE]
341 (i.e., optional leading sign, optional leading zeroes).
- 342 ○ Boolean values are represented in CSVs as either ‘0’ for false or ‘1’ for true (which is a subset of the
343 defined boolean data type values specified in [DEVICE]: ‘0’, ‘false’, ‘no’, ‘1’, ‘true’, ‘yes’).
- 344 ○ String values are represented in CSVs with the same syntax as the string data types specified in [DEVICE]
345 (i.e., any Unicode string), with two exceptions that are represented using a backslash escape character:
 - 346 ○ The comma (‘,’) is represented as ‘\,’.
 - 347 ○ The backslash (‘\’) is represented as ‘\\’.
- 348 ○ Any white space before, after, or interior to a string value is part of that string value. White space before,
349 after, or interior to any other data type is not allowed.

350 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> .

351

352 2.4.2. XML Content in UPnP Arguments and State Variables

353 The UPnP V1.0 architecture [DEVICE] specifies that all UPnP action argument values are transmitted inside a
 354 SOAP XML body. All argument values are passed as character data. When an argument value contains any XML
 355 markup, or any character that could be construed as XML markup, that argument value must be properly escaped
 356 according to the rules of XML ([XML] Section 2.4 Character Data and Markup). For example, the out argument
 357 MediaList of the action GetMediaList contains XML markup, and therefore MUST be properly escaped (see section
 358 2.6.3.2). The same would be true of the value of any evented state variable in a GENA message, but this document
 359 does not define any evented state variables with XML content.

360 The XML escaping rules are summarized from the [XML] reference mentioned above:

- 361 • The character ‘<’ MUST be encoded as one of:
 362 ‘<’
 363 ‘<’
 364 ‘<’
- 365 • The character ‘&’ MUST be encoded as one of:
 366 ‘&’
 367 ‘&’
 368 ‘&’
- 369 • When the character ‘>’ appears in the sequence ‘-->’ (‘>’ preceded by two hyphens or two minus signs) it
 370 MUST be encoded as one of:
 371 ‘>’
 372 ‘>’
 373 ‘>’
 374 All other occurrences of ‘>’ MAY be encoded. Therefore, to avoid special testing, ‘>’ SHOULD always be
 375 encoded.

376 2.5. Naming Conventions

377 All state variables, actions and action parameters are mixed case with the first letter of each word being capitalized.
378 Most of these variables, actions and parameters are derived directly from IPP by removing the hyphens and up-
379 casing the first letter of each word. Unless specified otherwise, all variable values and action parameter values are
380 all lower case with hyphens, as in IPP. See Internet Printing Protocol/1.0 Model and Semantics (RFC 2566) and
381 Internet Printing Protocol/1.1 Model and Semantics (RFC 2911), hereafter referred to as [MODEL]. The action and
382 attribute descriptions in these tables are only a brief summary. Implementations SHOULD conform to the complete
383 semantics specified in these referenced documents for each attribute indicated with [MODEL] in order to achieve
384 the kind of interoperability between client and Printer implementations of different vendors IPP has demonstrated.
385 A full description of their meaning can be found in the indicated sections in [MODEL].

386 2.6. State Variables

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

390 2.6.1. The Printer's Supported and Default Values

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

393 Each SST variable definition in this document specifies whether or not vendors in their Service Description MAY
394 subset and/or extend the <allowedValueList> element in their Service Description from those "Allowed Values"
395 values given in this document. The Printer's "current" <allowedValueList> and <defaultValue> values MAY or
396 MAY NOT be the same as the factory supported and default values, respectively, for that parameter; i.e., someone
397 may have changed the settings from the factory-supplied values. Any <allowedValueList> and <defaultValue>
398 element value MAY be changed at any time after Service Discovery. Furthermore, the current <allowedValueList>
399 and <defaultValue> values for a job parameter could also possibly change between invocations of the action that
400 uses it; for example, someone MAY reconfigure the Printer's "current" device setting for a particular parameter.
401 However, the UPnP Device Architecture, version 1.0 [DEVICE], states that any change to the <allowedValueList>
402 or <defaultValue> element requires the Printer to issue an "ssdp:byebye" and then re-advertise itself. Each of the
403 values in the <defaultValue> elements is implementation specific, but MUST be one of the values from the Service
404 Description's associated <allowedValueList> element, if present.

405 2.6.2. The Distinguished Value

406 Some Print Service actions have IN parameters that will always override any corresponding value that might be
407 provided in the PDL data stream (see section 2.9.3.1.1). For those situations where the Control Point prefers to let
408 the PDL data stream value override the IN parameter, the PrintEnhanced:1 Service has added the Distinguished
409 Value 'device-setting' to the <allowedValueList> of the associated state variable. In the case where the attribute is
410 absent in the PDL data stream and the IN parameter value is specified as the Distinguished Value 'device-setting',
411 the Service uses its <defaultValue> value for the IN parameter. For example, see CreateJob action, section 2.8.2.
412 When the Control Point supplies the Distinguished Value for such an IN parameter, the Print Service MUST
413 process the action following the corresponding print instruction in the PDL Data Stream, if present. If absent, the
414 Print Service MUST process the action as if the Service's current <defaultValue> for that IN parameter value had
415 been supplied by the Control Point. In other words, the Service's then current <defaultValue> value has lower
416 precedence than the PDL Data Stream. All implementations MUST support all Distinguished Value parameters
417 defined herein. The two preceding requirements also mean that the Distinguished Value for a variable MUST be
418 included in the variable's allowed value set, even if the vendor is sub-setting the allowed value set. However, the
419 Distinguished Value itself MUST NOT be used for the actual value of the <defaultValue> element in the SCPD.

420 Note: the Distinguished Values defined herein for a variable/parameter are not otherwise valid values for the
 421 variable/parameter.

422 The value used as the Distinguished Value for a parameter, is specified in the definition of the parameter’s
 423 associated state variable. This guarantees uniqueness of the Distinguished Value across all actions that might use it.
 424 Any vendor extensions to the set of Print Service actions that use IN parameters with an associated variable that has
 425 a defined Distinguished Value SHOULD also support the use of Distinguished Values in their action invocations.
 426 Any vendor extension that does support such Distinguished Values in their actions MUST use the same
 427 Distinguished Value that is defined in this document. While vendors MAY use the Distinguished Value concept in
 428 their Print Service extensions, this specification provides no mechanism for indicating either that Distinguished
 429 Values are supported or the actual Distinguished Value used for a specific variable/parameter.

430 The Distinguished Value for all string variables defined herein is the string ‘device-setting’. For any vendor
 431 extensions, the Distinguished Value for all string variables MUST be ‘device-setting’. The Distinguished Value for
 432 all integer variables defined herein is the value ‘0’. For any vendor extensions, the Distinguished Value for integer
 433 variables SHOULD be ‘0’ (or ‘-1’ if ‘0’ is otherwise a useful value).

434 **2.6.3. PrintEnhanced:1 Service Variables**

435 The first part of the SST defines the Printer attributes. The second part of the SST defines the Job attributes. Many
 436 of the Job attributes in the SST are present solely for the purpose of meeting the UPnP Device Architecture
 437 [DEVICE] requirement that all action parameters MUST have a related SST variable. The full specification for
 438 such action parameters is given with the variable in the SST. Some of the Printer attributes can be queried with the
 439 GetPrinterAttributes action (see section 2.8.8) or the GetPrinterAttributesV2 action (see section 2.8.9); and some of
 440 the Job attributes can be queried for a specified job with the GetJobAttributes action (see section 2.8.5).

441 **Table 1: State Variables**

Variable Name	Req. or Opt. ¹	Data Type	Allowed Value	Default Value (Mandatory except where indicated below)	Eng. Units
<i>Printer & Job Attributes (in alphabetical order)</i>					
A_ARG_TYPE_CriticalAttribList	R	<u>string</u> (<u>CSV</u> <u>string</u>)	See section 2.6.3.1	N/A	N/A
A_ARG_TYPE_MediaList	R	<u>string</u> (<u>well-balanced</u> <u>XML</u>)	See section 2.6.3.2	N/A	N/A
A_ARG_TYPE_PrinterAbortReason	R	<u>string</u>	See section 2.6.3.3	N/A	N/A
CharRepSupported	R	<u>string</u>	See section 2.6.3.4	<implementation specific>	N/A

Variable Name	Req. or Opt. ¹	Data Type	Allowed Value	Default Value (Mandatory except where indicated below)	Eng. Units
ColorSupported	R	<u>boolean</u>	See section 2.6.3.5	<implementation specific>	N/A
ContentCompleteList	R	<u>string</u> (<u>CSV i4</u>)	See section 2.6.3.6	<empty string>	N/A
Copies	R	<u>i4</u>	Range: 0 to 2 ³¹ -1	<implementation specific> RECOMMENDED value: 1	N/A
CriticalAttributesSupported	R	<u>string</u> (<u>CSV string</u>)	See section 2.6.3.8	<implementation specific>	N/A
DataSink	R	<u>uri</u>	See Section 2.6.3.9	<empty string>	N/A
DeviceId	R	<u>string</u> – <u>MUST be limited to 512 bytes.</u>		<implementation specific>	N/A
DocumentFormat	R	<u>string</u>	See section 2.6.3.11	<implementation specific> RECOMMENDED value: application/xhtml-print-e	N/A
DocumentUTF16Supported	R	<u>string</u>	See section 2.6.3.12	<implementation specific>	N/A
FullBleedSupported	R	<u>boolean</u>	See section 2.6.3.13	<implementation specific>	
InternetConnectState	R	<u>string</u>	See section 2.6.3.14	<implementation specific>	N/A
JobAbortState	R	<u>string</u> (<u>CSV i4, string, string, i4, string, string</u>)	See section 2.6.3.15	<empty string>	N/A
JobEndState	R	<u>string</u> (<u>CSV i4, string, string, i4, string</u>)	See section 2.6.3.16	<empty String>	N/A

Variable Name	Req. or Opt. ¹	Data Type	Allowed Value	Default Value (Mandatory except where indicated below)	Eng. Units
JobId	R	<u>i4</u>	Range: 0 to 2 ³¹ -1	0	N/A
JobIdList	R	<u>string</u> (CSV i4)	See section 2.6.3.18	<empty String>	N/A
JobMediaSheetsCompleted	R	<u>i4</u>	Range: -1 to 2 ³¹ -1	0 or -1	N/A
JobName	R	<u>string</u>		<empty string>	N/A
JobOriginatingUserName	R	<u>string</u>		<empty string>	N/A
MediaSize	R	<u>string</u>	See section 2.6.3.22	<implementation specific>	N/A
MediaType	R	<u>string</u>	See section 2.6.3.23	<implementation specific> RECOMMENDED value: Stationery (if supported)	N/A
NumberUp	R	<u>string</u>	See section 2.6.3.24	<implementation specific> RECOMMENDED value: 1	N/A
OrientationRequested	R	<u>string</u>	See section 2.6.3.25	<implementation specific> RECOMMENDED value: portrait	N/A
PageMargins	R	<u>string</u> (CSV <u>string</u> , <u>string</u> , <u>string</u> , <u>string</u>)	See section 2.6.3.26 The following represents an example: 1.0in, 1.0in, 2.0in, 2.0in, ...	<implementation specific>	N/A

Variable Name	Req. or Opt. ¹	Data Type	Allowed Value	Default Value (Mandatory except where indicated below)	Eng. Units
PrinterLocation	R	<u>string</u>		<implementation specific>	N/A
PrinterName	R	<u>string</u>		<implementation specific>	N/A
PrintQuality	R	<u>string</u>	See section 2.6.3.29	<implementation specific> RECOMMENDED value: normal	N/A
PrinterState	R	<u>string</u>	See section 2.6.3.30	idle	N/A
PrinterStateReasons	R	<u>string</u>	See section 2.6.3.31	none	N/A
Sides	R	<u>string</u>	See section 2.6.3.32	<implementation specific> RECOMMENDED value: one-sided	N/A
SourceURI	R	<u>uri</u>	See Section 2.6.3.33	<empty string>	
XHTMLImageSupported	R	<u>string</u>	See section 2.6.3.34	image/jpeg	N/A
Non-standard state variables implemented by a UPnP vendor go here.	X	TBD	TBD	TBD	TBD

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

443 2.6.3.1. A_ARG_TYPE_CriticalAttribList

444 *A_ARG_TYPE_CriticalAttribList is used as the related state variable for CriticalAttributesList which is used as the*
 445 *IN argument to CreateJobV2 or CreateURIJob. CriticalAttributesList is a CSV list of attributes from the*
 446 *allowedValueList of CriticalAttributesSupported (the exception to this is the value ‘none’). When the Control Point*
 447 *specifies the value “none” in the CriticalAttributesList, this means that the Control Point is not declaring any*
 448 *particular attribute as critical to the successful completion of the print job OR the Printer does not support any*
 449 *Critical Attributes.*

450 *When the Control Point provides the pdl-fidelity value in the A_ARG_TYPE_CriticalAttribList, it SHOULD NOT*
 451 *also provide other Critical Attributes that are controlled by the PDL. For example, when pdl-fidelity is contained*
 452 *in the A_ARG_TYPE_CriticalAttribList, the list SHOULD NOT also contain font-size.*

453 *If the A_ARG_TYPE_CriticalAttribList contains pdl-fidelity and other attributes also controlled by the PDL, and*
 454 *one or more of those attributes cannot be satisfied by the Printer, the Printer MAY provide either pdl-fidelity or the*
 455 *other attribute as the job-abort-reason.*

456 *When the Control Point specifies the “none” value in the CriticalAttributesList, it SHOULD NOT also provide*
 457 *other Critical Attributes. Printers MUST support “none” and MAY support any of the other values listed in the*
 458 *table below.*

459 Table 2: Values for CriticalAttributesList

Value	Req. or Opt.
<i>None</i>	<u>R</u>
<i>The value for CriticalAttributesList MUST be “none” or a CSV list of the following values (dependent on the printer’s implementation of CriticalAttributesSupported):</i>	
<i>copies</i>	<u>O</u>
<i>sides</i>	<u>O</u>
<i>number-up</i>	<u>O</u>
<i>orientation-requested</i>	<u>O</u>
<i>media-size</i>	<u>O</u>
<i>media-type</i>	<u>O</u>
<i>print-quality</i>	<u>O</u>
<i>text-layout</i>	<u>O</u>
<i>image-layout</i>	<u>O</u>
<i>image-orientation</i>	<u>O</u>
<i>pdl-fidelity</i>	<u>O</u>
<i>font-family</i>	<u>O</u>
<i>font-size</i>	<u>O</u>
<i>vendor-defined</i>	<u>O</u>

460

461 **2.6.3.2. A_ARG_TYPE_MediaList**462 *This variable is used as the related state variable for the OUT argument MediaList for the action GetMediaList.*463 *A value of type A_ARG_TYPE_MediaList is a possibly empty sequence of either MediaType or MediaSize*
464 *elements. More precisely, it can take one of the following three allowed forms:*

- 465 1. *An empty string.*
- 466 2. *A sequence of one or more <MediaType ...> elements.*
- 467 3. *A sequence of one or more <MediaSize ...> elements.*

468 *NOTE: This definition represents well-balanced XML, but **not** well-formed XML—there is no root (document)*
469 *element. Since the value is not well-formed, Control Points cannot pass it directly to a standard XML parser.*
470 *Control Points need to implement a workaround as described below in section 2.6.3.2.1. Device implementations*
471 *MUST NOT add a root element to this value in an attempt to make it well-formed XML, because it will produce*
472 *unexpected results with Control Points that are already implementing a workaround.*

473 *Also, in the following Examples 1 through 6, the value of the OUT argument MediaList is shown in its unescaped*
474 *XML form. Only Example 7 shows the MediaList argument that is fully escaped according to the requirements*
475 *described above in section 2.4.2.*

476

477 *Example 1: MediaType as a function of MediaSize*
478 IN: MediaSize="om_small-photo_100x150mm"
479 IN: MediaType="none"
480 OUT:
481 <MediaType MediaSize="om_small-photo_100x150mm">
482 photographic-glossy
483 photographic-matte
484 cardstock
485 </MediaType>
486

487 *Example 2: MediaSize as a function of MediaType*
488 IN: MediaSize="none"
489 IN: MediaType="photographic-glossy"
490 OUT:
491 <MediaSize MediaType="photographic-glossy">
492 na_index-4x6_4x6in
493 na_5x5_5x7in
494 na-8x10
495 na_letter_8.5x11in
496 </MediaSize>
497

498 *Example 3: All types for all sizes*
499 IN: MediaSize="none"
500 IN: MediaType="none"
501 OUT:
502 <MediaType MediaSize="om_small-photo_100x150mm">
503 photographic-glossy
504 photographic-matte
505 cardstock
506 </MediaType>
507 <MediaType MediaSize="jpn_hagaki_100x148mm">
508 photographic-glossy
509 photographic-matte
510 cardstock
511 </MediaType>
512 ...
513
514

515 *Example 4: MediaType as a function of the default value of MediaSize*
516 SCPD: <defaultValue>om_small-photo_100x150mm</defaultValue>
517 IN: MediaSize="device-setting"
518 IN: MediaType="none"
519 OUT:
520 <MediaType MediaSize="om_small-photo_100x150mm">
521 photographic-glossy
522 photographic-matte
523 cardstock
524 </MediaType>
525

526 *Example 5: MediaSize as a function of the default value of MediaType*
527 SCPD: <defaultValue>photographic-glossy</defaultValue>
528 IN: MediaSize="none"
529 IN: MediaType="device-setting"
530 OUT:
531 <MediaSize MediaType="photographic-glossy">
532 na_index-4x6_4x6in

```

533         na_5x5_5x7in
534         na-8x10
535         na_letter_8.5x11in
536     </MediaSize>
537

```

538 The final two examples illustrate the well-balanced XML before and after escape conversion by the Printer (refer to
539 section 2.4.2 for details on XML content in UPnP arguments):

540 *Example 6: All types for all sizes (before escape conversion)*

```

541 IN:  MediaSize="none"
542 IN:  MediaType="none"
543 OUT:
544     <MediaType MediaSize="om_small-photo_100x150mm">
545         photographic-glossy
546         photographic-matte
547         cardstock
548     </MediaType>
549     <MediaType MediaSize="jpn_hagaki_100x148mm">
550         photographic-glossy
551         photographic-matte
552         cardstock
553     </MediaType>
554     ...
555

```

556 *Example 7: All types for all sizes (after escape conversion)*

```

557 IN:  MediaSize="none"
558 IN:  MediaType="none"
559 OUT:
560     &lt;MediaType MediaSize="om_small-photo_100x150mm"&gt;
561         photographic-glossy
562         photographic-matte
563         cardstock
564     &lt;/MediaType&gt;
565     &lt;MediaType MediaSize="jpn_hagaki_100x148mm"&gt;
566         photographic-glossy
567         photographic-matte
568         cardstock
569     &lt;/MediaType&gt;
570     ...
571

```

572 2.6.3.2.1. Parsing the MediaList value

573 *When the Control Point receives the MediaList value, it first needs to unescape (reverse the escape conversion of)*
574 *the argument value text. Since the argument value is not well-formed XML, it cannot be passed directly to a normal*
575 *XML parser. The recommended approach to parsing is as follows:*

576 **Make the value well-formed by inserting a <MediaList> start-tag before the value and a**
577 **</MediaList> end-tag after the value, then apply normal parsing to the resulting well-formed value.**
578 **After this start-tag/end-tag insertion, the three forms of MediaList described above would look like this:**

```

579     1. <MediaList></MediaList>
580     2. <MediaList>
581         <MediaType ...>
582         ...
583     </MediaList>

```

584 3. <MediaList>
 585 <MediaSize ...>
 586 ...
 587 </MediaList>
 588

589 **2.6.3.3. A_ARG_TYPE_PrinterAbortReason**

590 *Used for one of the positional values of the evented state variable JobAbortState—see description in section*
 591 *2.6.3.15. Multiple conditions MAY exist. The vendor chooses the single value for the*
 592 *A_ARG_TYPE_PrinterAbortReason variable to indicate the most important condition.*

593 **Table 3: allowedValueList for A_ARG_TYPE_PrinterAbortReason**

Value	Req. or Opt.
<i>hardware-error</i>	<u><i>O</i></u>
<i>external-access-uri-not-found</i>	<u><i>O</i></u>
<i>external-access-object-failure</i>	<u><i>O</i></u>
<i>external-access-doc-format-err</i>	<u><i>O</i></u>
<i>external-access-http-error</i>	<u><i>O</i></u>
<i>vendor-defined</i>	<u><i>O</i></u>

594

595 **2.6.3.4. CharRepSupported**

596 *CharRepSupported is provided to enable the Control Point to determine which characters or glyphs a Printer*
 597 *supports for XHTML-Print. Support for glyphs that are included in CharRepSupported does not guarantee support*
 598 *in other PDL’s, e.g. PCL, Postscript, etc. Supported values are discoverable via the SCPD.*

599

600 *CharRepSupported SHALL use the naming conventions specified in [PWG5101.2] the Printer Working Group*
 601 *(PWG) Repertoire Supported Element. The capability to print 7-bit US-ASCII characters is not included in*
 602 *CharRepSupported; however, that capability is mandatory.*

603 *Based on that convention, the names of several common character repertoires would be:*

- 604 · “iana_iso_8859-1” commonly known as ISO 8859-1
- 605 · “iana_Shift_JIS” commonly known as Shift-JIS
- 606 · “unicode_katakana” from the Unicode Code Charts
- 607 · “vendor_lexmark_specials” a vendor specific character set

608

609 *IANA registered character set names are available from <http://www.iana.org/assignments/character-sets>. The*
 610 *Unicode names are available from <http://www.unicode.org/charts/index.html>.*

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

612 **Table 4: allowedValueList for CharRepSupported**

Value ³	Req. or Opt.
<i>iana_iso_8859-1</i>	<u>Q</u>
<i>iana_Shift_JIS</i>	<u>Q</u>
<i>unicode_katakana</i>	<u>Q</u>
<Other values defined for the Printer Working Group (PWG) Repertoire Supported Element by [PWG5101.2] >	<u>Q</u>
<i>Vendor-defined (see [PWG5101.2])</i>	<u>Q</u>

613

614

615 **2.6.3.5. ColorSupported**

616 *Identifies whether or not the device is capable of multi-hued color printing. A Printer that is capable of full color*
 617 *output has a value of '1' (TRUE). A grayscale capable or business graphics capable Printer has the value of '0'*
 618 *(FALSE), as would a highlight Printer. Supported values are discoverable via the SCPD.*

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

622 *All UPnP Printers MUST support either the '0' or the '1' value.*

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

624 **Table 5: allowedValueList for ColorSupported**

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

625

626 **2.6.3.6. ContentCompleteList**

627 *Contains a list of all jobs in the JobIdList (see 2.6.3.18) that are content complete. For a definition of content*
 628 *complete, see Sec 2.2.2 b). The ContentCompleteList is evented; it is triggered when the printer holds all*
 629 *information necessary to finish printing the job. The ContentCompleteList is not an OUT parameter of any action,*
 630 *so it is not available to a client (Control Point) via polling. This feature allows any device that holds content for a*
 631 *particular print job to leave the network as soon as all content for the job has been fetched.*

632 *The behavior of the Printer is dependent on the implementation. If a Printer implementation does not know when*
 633 *"content complete" has occurred, then it may return the ContentCompleteList event when the job is completed*
 634 *printing, is aborted or has been canceled. In this case, the client (Control Point) will receive the*
 635 *ContentCompleteList event at the same time as the JobEndState event.*

636 *Note: Content Complete status for a print job does not guarantee that it has been or will be successfully printed.*
637 *Even after the Printer has received all content for a job, there could still be content errors, processing errors or*
638 *mechanical problems. The only way to know that a print job has completed successfully is to monitor the evented*
639 *variable JobEndState.*

640 **2.6.3.7. Copies**

641 *Contains the number of copies of the document to be printed for the job. See [MODEL] section 4.2.5. Supported*
642 *values are discoverable via the SCPD.*

643 *The '0' Distinguished Value indicates that the Control Point wants the Printer to use its <defaultValue> value for*
644 *Copies, which MUST be greater than 0, but to allow that value to be overridden if a corresponding value is*
645 *encountered in the PDL Data Stream.*

646 *Vendors MAY subset the allowed values, but MUST support the '0' Distinguished Value.*

647 *Vendors MUST NOT extend the allowed values.*

648 **2.6.3.8. CriticalAttributesSupported**

649 *An attribute of a print job that the Printer can detect at print time and that the Printer guarantees to support fully or*
650 *else abort the job. See definition for Critical Attribute in section 2.2.2c). There are no required values in the*
651 *allowedValueList. Supported values are discoverable via the SCPD. When the Printer does not support any*
652 *Critical Attributes, the value "none" MUST be specified in CriticalAttributesSupported. The value "none" MUST*
653 *NOT be combined with any other values in CriticalAttributesSupported.*

654 **2.6.3.8.1. Values With Corresponding IN Arguments**

655 *The first several values in the allowedValueList correspond directly to CreateJobV2 and CreateURIJob IN*
656 *arguments (i.e., copies, sides, number-up, orientation-requested, media-size, media-type and print-quality). The*
657 *presence of any of these values in the CriticalAttributesSupported list indicates that the Printer MUST abort a job*
658 *when the value is included in the CriticalAttributesList if it cannot satisfy the value requested in the corresponding*
659 *IN argument. Additionally, for layout attributes (orientation-requested, media-size, and media-type), the Printer*
660 *MUST abort a job when the PDL data stream requests a corresponding value that cannot be honored. (See*
661 *sections 2.9.3.1.2 and 2.9.3.2.)*

662 **2.6.3.8.2. Text-layout**

663 *When text-layout is included in the CriticalAttributesSupported list and the CriticalAttributesList, the Printer MUST*
664 *abort any job which requests a text layout that the Printer cannot satisfy.*

665 **Example 1:**

666 *CriticalAttributesSupported contains text-layout*

667 *CreateJobV2 IN: MediaSize='device-setting', CriticalAttributesList contains text-layout*

668 *The PDL indicates that a page break should be avoided anywhere within a long span of text*

669 *which cannot be printed on one sheet of the Printer's default media size. The Printer MUST abort*
670 *the job when it discovers this problem.*

671 **Example 2:**

672 *CriticalAttributesSupported contains text-layout*

673 *CreateJobV2 IN: CriticalAttributesList contains text-layout*

674 *The PDL indicates that text should be placed 0.1 mm to the right of the left edge of the page. The*

675 *Printer cannot reliably position text at that location, so it MUST abort the job when it discovers*
676 *this problem*

677 **2.6.3.8.3. Image-layout**

678 *When image-layout is included in the CriticalAttributesSupported list and the CriticalAttributesList, the Printer*
679 *MUST abort any job which requests an image layout that the Printer cannot satisfy.*

680 Example 1:

681 *CriticalAttributesSupported contains image-layout*

682 *CreateJobV2 IN: CriticalAttributesList contains image-layout*

683 *The PDL indicates that an image should be printed so that it covers the surface of the medium*
684 *except for a 1 mm margin around the edge. The Printer is not capable of reliably printing images*
685 *with such a narrow margin, so the Printer MUST abort the job when it discovers this problem.*

686 Example 2:

687 *CriticalAttributesSupported contains image-layout*

688 *CreateJobV2 IN: CriticalAttributesList contains image-layout*

689 *The PDL indicates that 10 images should be placed side-by-side across the page. When the*
690 *Printer retrieves the source information, it discovers that it cannot buffer sufficient image data to*
691 *compose the required output. The Printer MUST abort the job when it discovers this problem.*

692 2.6.3.8.4. Image-orientation

693 *When image-orientation is included in the CriticalAttributesSupported list and the CriticalAttributesList, the Printer*
694 *MUST abort any job which requests an image orientation that the Printer cannot satisfy.*

695 *NOTE: Image-orientation applies only to individual images on the page. When the DocumentFormat is*
696 *application/xhtml-print-e, image rotation is controlled by the image-orientation attribute. This is not to be confused*
697 *with OrientationRequested, which applies to the page contents as a whole.*

698 2.6.3.8.5. Pdl-fidelity

699 *When pdl-fidelity is included in the CriticalAttributesSupported list and the CriticalAttributesList, the Printer MUST*
700 *abort any job which contains a PDL directive that the Printer cannot satisfy.*

701 *Pdl-fidelity applies to **all** constructs contained within the PDL data stream; it MAY therefore encompass other*
702 *Critical Attributes such as image-layout and font-family. It SHOULD be used only when very strict adherence to the*
703 *letter of the job instructions is required.*

704 *When a PDL attribute is overridden by an IN production argument (see section 2.9.3.1.2), pdl-fidelity is not*
705 *compromised and the job MUST NOT be aborted, so long as the Printer can perform the requested override and*
706 *pdl-fidelity is not compromised elsewhere.*

707 Example 1:

708 *CriticalAttributesSupported contains pdl-fidelity.*

709 *CreateJobV2 IN: CriticalAttributesList contains pdl-fidelity.*

710 *The PDL indicates that an image should be printed so that it covers the surface of the medium*
711 *except for a 1 mm margin around the edge. The Printer is not capable of reliably printing images*
712 *with such a narrow margin, so the Printer MUST abort the job when it discovers this problem.*

713 **Note:** *In this example, image-layout is compromised. Since image-layout is a function of the*
714 *PDL, pdl-fidelity is also compromised.*

715 Example 2:

716 *CriticalAttributesSupported contains pdl-fidelity.*

717 *CreateJobV2 IN: CriticalAttributesList contains pdl-fidelity.*

718 *The PDL indicates that a table should be nested inside another table, but the Printer does not*
719 *support the nesting of tables. The Printer MUST abort the job when it discovers this problem.*

720 Example 3:

721 *CriticalAttributesSupported contains pdl-fidelity.*

722 *CreateURIJob IN: CriticalAttributesList contains pdl-fidelity.*
723 *The PDL indicates that the job should be printed with content imposed on both sides of the media.*
724 *The IN argument indicates that the job should be printed 'one-sided'. The Printer is unable to*
725 *meet the two-sided request in the PDL data stream, but MUST NOT abort the job, because it is*
726 *able to satisfy the requirement to override that request with the IN Production argument request*
727 *for single-sided output.*

728 **2.6.3.8.6. Font-family**

729

730 *When font-family is included in the CriticalAttributesSupported list and the CriticalAttributesList, the Printer MUST*
731 *abort any job which requests a font typeface (such as Arial) or font family qualifier (such as sans-serif) that the*
732 *Printer cannot satisfy.*

733 **2.6.3.8.7. Font-size**

734 *When font-size is included in the CriticalAttributesSupported list and the CriticalAttributesList, the Printer MUST*
735 *abort any job which requests a font size that the Printer cannot satisfy.*

736

737 **Table 6: allowedValueList for CriticalAttributesSupported**

Value	Req. or Opt.
<i>none</i>	<u>Q</u>
The value for CriticalAttributesSupported MUST be “none” or an allowedValueList of the following values:	
<i>copies</i>	<u>Q</u>
<i>sides</i>	<u>Q</u>
<i>number-up</i>	<u>Q</u>
<i>orientation-requested</i>	<u>Q</u>
<i>media-size</i>	<u>Q</u>
<i>media-type</i>	<u>Q</u>
<i>print-quality</i>	<u>Q</u>
<i>text-layout</i>	<u>Q</u>
<i>image-layout</i>	<u>Q</u>
<i>image-orientation</i>	<u>Q</u>
<i>pdl-fidelity</i>	<u>Q</u>
<i>font-family</i>	<u>Q</u>
<i>font-size</i>	<u>Q</u>
<i>vendor-defined</i>	<u>Q</u>

738 **2.6.3.9. DataSink**

739 *Contains the URI to which the Control Point is to send the HTTP Post operation (see section 2.8.10) for the job.*
 740 *This value is returned by the Printer in the CreateJob* action response, rather than being supplied by the Control*
 741 *Point in the request.*

742 **2.6.3.10. DeviceId**

743 *The value of this variable MUST exactly match the IEEE 1284-2000 Device ID string, except the length field MUST*
 744 *NOT be specified. The supported value for DeviceId is discoverable in the <defaultValue> value via the SCPD.*
 745 *The length of DeviceId, defined as a string, is limited to 512 bytes.*
 746 *The IEEE 1284-2000 Device ID consists of a length field followed by a case-sensitive string of ASCII characters*
 747 *defining peripheral characteristics and/or capabilities. For the purposes of this specification, the length bytes*
 748 *MUST NOT be included. The Device ID sequence is composed of a series of keys and values of the form:*
 749 *key: value {,value} repeated for each key*

750 As indicated, each key *MUST* have one value, and *MAY* have more than one value. The minimum necessary keys
 751 (case-sensitive) are MANUFACTURER, COMMAND SET, and MODEL. (These keys *MAY* be abbreviated as MFG,
 752 CMD, and MDL respectively.) Each implementation *MUST* supply these three keys and possibly additional ones as
 753 well. Each key (and each value) is a string of characters. Any characters except colon (:), comma (,), and semi-
 754 colon (;) *MAY* be included as part of the key (or value) string. Any leading or trailing white space (SPACE[x'20'],
 755 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
 756 is still counted as part of the overall length of the sequence).

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

```
759
760 MANUFACTURER:ACME Manufacturing;
761 COMMAND SET:PCL,PJL,PS,XHTML-Print;
762 MODEL:LaserBeam 9;
763 COMMENT:Anything you like;
764 ACTIVE COMMAND SET:PCL;
```

765 (See IEEE 1284-2000 clause 7.6)

766 Note: One of the purposes of the DeviceId variable is to select a printer driver for those Control Points that need a
 767 printer driver. The values of the COMMAND SET key are interpreted by the printer driver provided by the vendor
 768 and so are vendor-defined, rather than being standardized.

769 2.6.3.11. DocumentFormat

770 Identifies the DocumentFormat of the job as a MIME media type. Supported values are discoverable via the SCPD
 771 in the <allowedValueList>.

772 All UPnP Printers *MUST* support XHTML-Print [XHTML-PRINT] and CSS-Print [CSSPP], including the
 773 Enhanced Layout extension. Accordingly, all Printers *MUST* support the following MIME types as identifiers for
 774 this document format:

- 775 • 'application/vnd.pwg-xhtml-print': This MIME media type is deprecated in favor of
 776 'application/xhtml-print'. It *SHOULD NOT* be used by Control Points, and *MUST* be
 777 supported by Printers.
- 778 • 'application/xhtml-print': This MIME type identifies the base level of XHTML-Print/CSSPP
 779 support.
- 780 • 'application/xhtml-print-e': This MIME type identifies documents conforming to the Enhanced
 781 Layout profile of XHTML-Print/CSSPP.

782 In addition, all Printers *MUST* support the 'unknown' value as described below.

783 One special value is 'application/octet-stream'. If the Printer service supports this value, the Printer service *MUST*
 784 be capable of auto-sensing the format of the document data.

785 Another special value is 'unknown'. This value is intended for the Control Point to supply that does not know the
 786 document format of the document data. The behavior of the Printer when receiving the 'unknown' value is
 787 IMPLEMENTATION DEFINED. However, if the Printer can perform auto sensing of the data, (the
 788 'application/octet-stream' behavior), it is RECOMMENDED that it do so.

789 If the Control Point (client) does not know the document format, it *SHOULD* supply the 'application/octet-stream'
 790 value and let the Printer determine the format, unless the Printer doesn't support the 'application/octet-stream'
 791 value, in which case the Control Point's only recourse is to supply the special 'unknown' value.

792 (See [MODEL] section 4.1.9)

793 The vendors *MAY* extend the allowed values for this attribute, but *MUST NOT* support the 'device-setting'
 794 Distinguished Value.

795 **Table 7: allowedValueList for DocumentFormat**

Value	Req. or Opt.
<i>unknown</i>	<u>R</u>
<i>application/vnd.pwg-xhtml-print</i> <i>[deprecated in favor of application/xhtml-print]</i> <i>See NOTE below.</i>	<u>R</u>
<i>application/xhtml-print</i>	<u>R</u>
<i>application/vnd.pwg-xhtml-print+xml</i> <i>[deprecated in favor of application/xhtml-print]</i> <i>See NOTE below.</i>	<u>O</u>
<i>application/xhtml-print-e</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><Registered MIME media types for other document formats> See NOTE below.</i>	<u>O</u>
<i>Vendor-defined</i> <i>See NOTE below.</i>	<u>O</u>

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

799 **2.6.3.12. DocumentUTF16Supported**

800 *Identifies whether the Printer supports UTF-16 for the DocumentFormats supported. Supported values are*
801 *discoverable via the SCPD.*

802 *Vendors MUST support one of the following: “none” OR “all” OR specified allowed values of DocumentFormat.*
803 *The Printer MUST NOT combine “none” with any other values. The Printer MUST NOT combine “all” with any*
804 *other values.*

805 **Table 8: allowedValueList for DocumentUTF16Supported**

Value	Req. or Opt.
<i>none</i>	<u>0</u>
<i>all</i>	<u>0</u>
The value for DocumentUTF16Supported MUST be “none” or “all” or an <allowedValue> list of the following values:	
<i>application/vnd.pwg-xhtml-print</i> [deprecated in favor of application/xhtml-print] See NOTE below.	<u>0</u>
<i>application/xhtml-print</i>	<u>0</u>
<i>application/vnd.pwg-xhtml-print+xml</i> [deprecated in favor of application/xhtml-print] See NOTE below.	<u>0</u>
<i>application/xhtml-print-e</i>	<u>0</u>
<i>text/plain</i>	<u>0</u>
<i>text/plain; charset=utf-8</i>	<u>0</u>
<i>application/octet-stream</i>	<u>0</u>
<i>application/postscript</i>	<u>0</u>
<i>application/vnd.hp-PCL</i>	<u>0</u>
<i><Registered MIME media types for other document formats> See NOTE below.</i>	<u>0</u>
<i>Vendor-defined</i> See NOTE below.	<u>0</u>

806 **2.6.3.13. FullBleedSupported**

807 *Indicates whether or not the Printer supports full-bleed printing for a particular media size / type combination. See*
 808 *section 2.8.6 for further details. A ‘0’ indicates that full-bleed printing is not supported for the associated media*
 809 *size / type, whereas a ‘1’ indicates that full-bleed printing is supported for the associated media size / type.*

810 *All UPnP Printers MUST support either the ‘0’ or the ‘1’ value.*

811 *Vendors MUST NOT extend the allowedValueList.*

812

813 **Table 9: allowedValueList for FullBleedSupported**

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

814

815 **2.6.3.14. InternetConnectState**

816 *InternetConnectState* tells the client (Control Point) whether the Printer currently has a connection to the Internet.
 817 Its three possible values and meanings are:

- 818 *unknown* — it is not known whether the Printer has a connection to the Internet.
- 819 *connected* — the Printer has access to the Internet.
- 820 *not-connected* — the Printer does not have access to the Internet.

821 This information provides a best-effort indication as to whether or not a Printer is likely to be able to successfully
 822 process a job which requires retrieving information from the Internet. It cannot be absolutely relied upon, because
 823 many conditions must be met in order for the job to complete successfully. E.g., the connection must remain
 824 uninterrupted, the particular servers providing the information to be retrieved must be up and available at the time
 825 of access, the files holding the information must be present and accessible, etc.

826 The Control Point **SHOULD NOT** proceed with creating a job which requires such connectivity if the Printer
 827 reports that it is ‘not-connected’. The Control Point **SHOULD** proceed with creating the job if the
 828 *InternetConnectState* is ‘connected’ or ‘unknown’.

829 The method used to determine the *InternetConnectState* is implementation specific.

830 All UPnP Printers **MUST** support one of the following values (i.e., *unknown*, *connected*, or *not-connected*) in the
 831 *GetPrinterAttributesV2* response.

832

833 **Table 10: allowedValueList for *InternetConnectState***

Value	Req. or Opt.
<i>unknown</i>	<u>O</u>
<i>connected</i>	<u>O</u>
<i>not-connected</i>	<u>O</u>

834

835 **2.6.3.15. JobAbortState**

836 This variable holds the “terminating” state of the job most recently aborted by the Printer. It is evented; it is
 837 triggered when any job terminates by being aborted, instead of being canceled or ending successfully.

838 *JobAbortState* is not an OUT parameter of any action, so it is not available to a Control Point via polling.

839 *JobAbortState* is a heterogeneous CSV list of six items: *JobId*, *JobName*, *JobOriginatingUserName*,
 840 *JobMediaSheetsCompleted*, *job-completion-state*, *job-abort-reason*.

841 The first five are the same items, in the same order, as the state variable *JobEndState* (refer to 2.6.3.16).

842 Furthermore, the values of these five items will be the same as the values of *JobEndState*, for the corresponding
 843 print *JobId*. In particular, note that the value of *job-completion-state* will always be ‘**aborted**’. The sixth value will
 844 be from the combined allowedValueLists of *CriticalAttributesSupported* and *A_ARG_TYPE_PrinterAbortReason*.

845 Multiple conditions **MAY** exist. The vendor chooses the single value for the *job-abort-reason* variable to indicate
 846 the most important condition.

847 **Table 11: allowedValueList for *job-abort-reason***

Value	Req. or Opt.
<i>hardware-error</i>	<u>Q</u>
<i>external-access-uri-not-found</i>	<u>Q</u>
<i>external-access-object-failure</i>	<u>Q</u>
<i>external-access-doc-format-err</i>	<u>Q</u>
<i>external-access-http-error</i>	<u>Q</u>
<i>copies</i>	<u>Q</u>
<i>sides</i>	<u>Q</u>
<i>number-up</i>	<u>Q</u>
<i>orientation-requested</i>	<u>Q</u>
<i>media-size</i>	<u>Q</u>
<i>media-type</i>	<u>Q</u>
<i>print-quality</i>	<u>Q</u>
<i>text-layout</i>	<u>Q</u>
<i>image-layout</i>	<u>Q</u>
<i>image-orientation</i>	<u>Q</u>
<i>pdl-fidelity</i>	<u>Q</u>
<i>font-family</i>	<u>Q</u>
<i>font-size</i>	<u>Q</u>
<i>vendor-defined</i>	<u>Q</u>

848

849 **2.6.3.16. JobEndState**

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

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

856 ***JobId:** the JobId of the job being removed. See section 2.6.3.17.*

857 ***JobName:** The name of the job. See section 2.6.3.20.*

858 ***JobOriginatingUserName:** The name of the user that submitted the job. See section 2.6.3.21.*

859 ***JobMediaSheetsCompleted:** If JobId was the “active” job, i.e., the first job in JobIdList, this is the final*
 860 *value of JobMediaSheetsCompleted for the job. Otherwise, this value is ‘0’. See section 2.6.3.19.*

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

862 ***aborted:** The job did not complete successfully, for one of two reasons—either (1) the Printer*
863 *encountered a non-recoverable error while processing the job or attempting to receive the data,*
864 *or (2) the job was created by the CreateJobV2 or CreateURIJob and the Printer detected during*
865 *processing that the job requirements covered by the CriticalAttributesList parameter could not be*
866 *met.*

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

869 ***canceled:** The job was canceled either by a CancelJob action or the equivalent in another*
870 *protocol, or by user intervention.*

871 **2.6.3.17. JobId**

872 *An i4 value identifying a particular job which has been submitted to the Printer. The JobId is assigned by the*
873 *Printer upon a successful Create* action. See section 2.8.2 and 2.8.3 for further details.*

874 *(See [MODEL] section 4.3.2)*

875 **2.6.3.18. JobIdList**

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

879 *The list is a sequence of Comma Separated i4 Values (CSV i4 - see section 2.4.1). Each value is a JobId of a job on*
880 *the Printer. The values range from 1 to $2^{31}-1$. The list is in the order that the jobs are expected to be completed.*

881 *The first job in the list is either currently printing, attempting to print (but the Printer is stopped), or is the next job*
882 *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*
883 *printed last. The first JobId in the list is removed when the job completes or is aborted. The corresponding JobId*
884 *in the list is removed when a job is canceled (see sections 2.8.2 and 2.8.3).*

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

886 *The Print Service, on receipt of a new job, generates a JobId which identifies the new Job on that Print Service.*
887 *The JobId is placed in the appropriate place in the JobIdList. The Print Service returns the value of the JobId*
888 *parameter as part of the response to a Create* action.*

889 **2.6.3.19. JobMediaSheetsCompleted**

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

894 *It is possible in some implementations that the final value of JobMediaSheetsCompleted is known, but that*
895 *intermediate values are not known. In this case the Printer SHOULD return 0 for a job that is not active, -1 for an*
896 *active job and the proper final value for completed jobs. The Printer MUST still return -1 for*
897 *JobMediaSheetsCompleted when it does not know the value, even in situations that it normally would know the*
898 *value. A Control Point MUST NOT conclude that receipt of a value of -1 for JobMediaSheetsCompleted means that*
899 *the Printer will always return -1. Even implementations that can never successfully count media sheets completed*
900 *might still know that a canceled or aborted job never marked any paper, so it could properly return a value of '0'*
901 *for JobMediaSheetsCompleted in the JobEndState variable.*

902 **2.6.3.20. JobName**

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

905 **2.6.3.21. JobOriginatingUserName**

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

909 **2.6.3.22. MediaSize**

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

918 *For sizes that do not have standard names, a Control Point or a Print Service can create a customized name using*
919 *the ‘custom_XXX’ class and name, where XXX indicates the custom name of the medium, followed by the dimensions*
920 *in inches or millimeters as for standard names. For example, a custom 3.5 by 5.0 inch medium that, say, represents*
921 *an index card, could be indicated by the string value:*

922 `custom_index-card_3.5x5in`

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

924 *If a Printer supports the Control Point supplying custom names that are not one of the values in the Printer’s*
925 *<allowedValueList> element, the Printer’s <allowedValueList> element MUST include both the*
926 *‘custom_max_IIIxJJJmm’ and ‘custom_min_IIIxJJJmm’ (and/or ‘custom_max_IIIxJJJin’ and*
927 *‘custom_min_IIIxJJJin’) Allowed Values to indicate the minimum and maximum custom sizes that the Printer will*
928 *allow the Control Point to supply.*

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

931 *The ‘device-setting’ Distinguished Value indicates that the Control Point wants the Printer to use its*
932 *<defaultValue> value for MediaSize, but to allow that value to be overridden if a corresponding value is*
933 *encountered in the PDL Data Stream.*

934 *Vendors MAY subset and extend allowed values, but MUST support the ‘device-setting’ Distinguished Value.*
935 *Vendor-extended values MUST follow the naming guidelines provided in PWG5101.1.*

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

938 **Table 12: allowedValueList for MediaSize**

Value ³	Req. or Opt.
<i>device-setting</i>	<u>R</u>
<i>none</i>	<u>R</u>
<i>om_small-photo_100x150mm</i>	<u>Q</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>
< Other values defined for media size by [PWG5101.1] >	<u>Q</u>
Vendor-defined (see [PWG5101.1])	<u>Q</u>

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

940 2.6.3.23. MediaType

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

942	stationery	Separately cut sheets of an opaque material
943	transparency	Separately cut sheets of a transparent material
944	envelope	Envelopes that can be used for conventional mailing purposes
945	labels	Label stock [For example, a sheet of peel-off labels].
946	photographic	Separately cut sheets of an opaque material to produce photographic quality images
947	cardstock	Separately cut sheets of an opaque material that is heavier and stiffer than stationery.
948	device-setting	Indicates that the Control Point wants the Printer to use its <defaultValue> value for
949		MediaType.

950 The values are a subset of and the descriptions are taken verbatim from the Media Type Names in [PWG5101.1].

951 The 'device-setting' Distinguished Value indicates that the Control Point wants the Printer to use its
 952 <defaultValue> value for MediaType, but to allow that value to be overridden if a corresponding value is
 953 encountered in the PDL Data Stream.
 954 Vendors MAY subset or extend allowed values, but MUST support the 'device-setting' Distinguished Value. See
 955 [PWG5101.1] for additional example values.
 956 How the Printer's Service Description <defaultValue> and <allowedValueList> elements are configured with these
 957 values is implementation-specific, e.g., local console, Presentation Service (web access).

958 **Table 13: allowedValueList for MediaType**

Value ³	Req. or Opt.
<i>device-setting</i>	<u>R</u>
<i>none</i>	<u>R</u>
<i>stationery</i>	<u>Q</u>
<i>stationery-inkjet</i>	<u>Q</u>
<i>transparency</i>	<u>Q</u>
<i>envelope</i>	<u>Q</u>
<i>labels</i>	<u>Q</u>
<i>photographic</i>	<u>Q</u>
<i>photographic-glossy</i>	<u>Q</u>
<i>photographic-matte</i>	<u>Q</u>
<i>cardstock</i>	<u>Q</u>
< Other values defined for media type by [PWG5101.1] >	<u>Q</u>
Vendor-defined (see [PWG5101.1])	<u>Q</u>

959 **2.6.3.24. NumberUp**

960 *Description: Indicates the number of PDL Data Stream pages to impose upon a single side of an instance of a*
 961 *selected medium for the job. The device's supported values are discoverable via the SCPD. Examples:*

962 *1 - One page per side.*

963 *2 - Two pages per side.*

964 *4 - Four pages per side.*

965 ***device-setting***

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

968 *(See [MODEL] section 4.2.9)*

969 The ‘device-setting’ Distinguished Value indicates that the Control Point wants the Printer to use its
 970 <defaultValue> value for NumberUp, but to allow that value to be overridden if a corresponding value is
 971 encountered in the PDL Data Stream.
 972 Vendors MAY subset or extend allowed values, but MUST support the ‘device-setting’ Distinguished Value.

973 **Table 14: allowedValueList for NumberUp**

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

974

975 **2.6.3.25. OrientationRequested**

976 Indicates the desired orientation for printed pages for any DocumentFormat. Supported values are discoverable
 977 via the SCPD. Which MIME media type document formats a Printer is able to orient as requested depends on
 978 implementation and MAY depend on the actual document content. Values:

- 979 portrait
- 980 landscape
- 981 reverse-landscape
- 982 reverse-portrait
- 983 device-setting

984 NOTE: OrientationRequested applies to all content on the page. It is not to be confused with the CSSPP attribute,
 985 image-orientation. The latter applies only to individual images and not to the page contents as a whole. Support for
 986 image-orientation is required as part of the feature set mandated for Enhanced CSSPP [CSSPP]. Support for
 987 OrientationRequested is optional; supported values are discoverable via the SCPD.

988

989 (See [MODEL] section 4.2.10 which intends the “orientation-requested” attribute to apply to ‘text’ MIME types.)

990 The ‘device-setting’ Distinguished Value indicates that the Control Point wants the Printer to use its
 991 <defaultValue> value for OrientationRequested, but to allow that value to be overridden if a corresponding value
 992 is encountered in the PDL Data Stream.

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

994 Vendors MUST NOT extend allowed values.

995 **Table 15: 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>

996

997 **2.6.3.26. PageMargins**

998 *Identifies the four margin sizes that the PrintEnhanced:1 service uses for the specific Media Type and Media Size*
 999 *combination requested in the GetMargins action, so that a Control Point can determine the printable area for a*
 1000 *specified media. Each margin size is the absolute distance between the edge of the media and the nearest edge of*
 1001 *the printable area. The string value of this variable is a CSV consisting of exactly four string values with no spaces*
 1002 *anywhere. Each value MAY have leading zeroes. Each value MAY include a non-zero decimal fraction set off by a*
 1003 *period (.) and MAY have trailing zeroes. Each of the four values is separated by a comma (,) and the order of the*
 1004 *values indicates Top margin, Right margin, Bottom margin, and Left margin (as specified in CSS2). All media are*
 1005 *assumed to be portrait for purposes of defining Top, Right, Bottom and Left. Each value MUST include the Inch or*
 1006 *Millimeter dimension indicator: 'in' or 'mm', respectively, immediately after each dimension.*

1007 *Example: A na-letter medium that has a quarter of an inch margin on the Top, Right, and Left edges, and 0 on the*
 1008 *Bottom edge would be (no spaces): 0.25in, .25in, 0in, .25in.*

1009 This specification does not define an allowed value list for this attribute. Vendors MUST supply the allowed values
 1010 for this attribute.

1011 **2.6.3.27. PrinterLocation**

1012 *Indicates the location of the device. For example, "Bobby's room". How the Printer's Service Description*
 1013 *<defaultValue> element is configured with this value is implementation-specific; e.g., local console, Presentation*
 1014 *Service (web access).*

1015 *(See [MODEL] section 4.4.4)*

1016 **2.6.3.28. PrinterName**

1017 *The administratively assigned user-friendly name of the Printer. How the Printer's Service Description*
 1018 *<defaultValue> element is configured with this value is implementation-specific, e.g., local console, Presentation*
 1019 *Service (web access). If the physical device has only one device, then the Device's <friendlyName> and*
 1020 *PrinterName are recommended to have the same value. However, if the physical device contains several devices,*
 1021 *the PrinterName identifies the Printer device.*

1022 *(See [MODEL] section 4.4.4)*

1023 **2.6.3.29. PrintQuality**

1024 *Specifies the print quality requested for the job. Supported values are discoverable via the SCPD. Values:*

1025 *draft*
 1026 *normal*
 1027 *high*
 1028 *device-setting*

1029 (See [MODEL] section 4.2.13)

1030 The 'device-setting' Distinguished Value indicates that the Control Point wants the Printer to use its
1031 <defaultValue> value for PrintQuality, but to allow that value to be overridden if a corresponding value is
1032 encountered in the PDL Data Stream.

1033 Vendors MAY subset allowed values, but MUST support the 'device-setting' Distinguished Value.

1034 Vendors MUST NOT extend allowed values.

1035 **Table 16: 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>

1036

1037 2.6.3.30. PrinterState

1038 Identifies the current state of the service. Values:

1039 **idle** - new jobs can start processing immediately without waiting.

1040 **processing** - jobs (Tracked or Untracked) are being processed; new jobs will wait before processing.
1041 These jobs are said to be 'pending'.

1042 **stopped** - no jobs can be processed and intervention is needed.

1043 (See [MODEL] section 4.4.11)

1044 Vendors MUST NOT subset or extend allowed values.

1045 **Table 17: 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>

1046

1047 2.6.3.31. PrinterStateReasons

1048 Indicates additional information about why the Printer is in its current state. Multiple conditions MAY exist. The
1049 vendor chooses the single value for PrinterStateReasons variable to indicate the most important condition.

1050 Note: Some of these reasons describe states of the Printer that cannot be entered on the basis of the currently
1051 defined UPnP actions set. For example, the Printer can be 'paused'; there is no PausePrinter action. The reason
1052 these states are presented is because some other protocol (or console action) can have caused the Printer to enter
1053 that state. Reason values:

1054 **none** - Indicates that there are no current state reasons

1055 **attention-required** - The device has stopped for a reason other than the PrinterStateReasons listed here and
1056 requires human intervention before it can continue.

- 1057 **media-jam** - The device has a media jam.
- 1058 **paused** - Someone has paused the Printer and the PrinterState is 'stopped'. In this state, a Printer will not produce
1059 printed output.
- 1060 **door-open** - One or more covers on the device are open.
- 1061 **media-low** - At least one input tray is low on media.
- 1062 **media-empty** - At least one input tray is empty.
- 1063 **output-area-almost-full** - One or more output areas is almost full (e.g., tray, stacker, collator).
- 1064 **output-area-full** - One or more output areas is full (e.g., tray, stacker, collator).
- 1065 **marker-supply-low** - The device is low on at least one marker supply (e.g., toner, ink, ribbon).
- 1066 **marker-supply-empty** - The device is out of at least one marker supply (e.g., toner, ink, ribbon).
- 1067 **marker-failure** - The device has at least one marking device which has failed and requires service or replacement.
- 1068 **media-change-request** - A job has been submitted that is requesting media that is currently not loaded. The job
1069 has specified a particular MediaSize and MediaType parameter value combination that is not loaded, although the
1070 Printer supports that combination. The user is prompted to load the appropriate media. The Printer is paused until
1071 the user has responded to the prompt.
- 1072 (See [MODEL] section 4.4.12. The IPP severity suffix **MUST NOT** be included and, unlike IPP, only one value
1073 **MUST** occur at a time.)
- 1074 Vendors **MUST** support the values that represent conditions that are detectable in their implementation. Therefore,
1075 vendors **MAY** subset allowed values if specific PrinterStateReasons are undetectable in their implementation.
- 1076 Vendors **MAY** extend allowed values. However, Printer vendors need to understand the implications of extending
1077 this list for a Control Point. The Control Point usually localizes the PrinterStateReasons value (as with other string
1078 variable values) to the human language of the user. However, such a Printer vendor extension value will not be
1079 recognized by the Control Point. As a fallback presentation, the Control Point **MAY** display the value received as
1080 is, which **SHOULD** be in English and therefore, might not be understandable by the user. Alternatively, the vendor
1081 might use the general PrinterStateReasons value: 'attention-required' and then explain the problem on the Printer
1082 console which the user would see when they are by the Printer.

1083 **Table 18: allowedValueList for PrinterStateReasons**

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

1084 ³ Vendors **MUST** support the values that represent conditions that are detectable in their implementation.

1085 **2.6.3.32. Sides**

1086 *Specifies how print content is to be imposed upon the two surfaces (sides) of the media for the job. Supported*
 1087 *values are discoverable via the SCPD.*

1088

1089 *(See [MODEL] section 4.2.8.)*

1090 *The 'device-setting' Distinguished Value indicates that the Control Point wants the Printer to use its*
 1091 *<defaultValue> value for Sides, but to allow that value to be overridden if a corresponding value is encountered in*
 1092 *the PDL Data Stream.*

1093 *Vendors MAY subset allowed values, but MUST support the 'device-setting' Distinguished Value.*

1094 *Vendors MUST NOT extend allowed values.*

1095 **Table 19: 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>O</u>
<i>two-sided-short-edge</i>	<u>O</u>

1096

1097 **2.6.3.33. SourceURI**

1098 *Contains the URI to which the device will send the HTTP GET operation (see section 2.8.11) to get the print*
 1099 *document. This value is sent by the client (Control Point) in the CreateURIJob action request.*

1100 **2.6.3.34. XHTMLImageSupported**

1101 *Identifies the Image formats supported by the Printer. Supported values are discoverable via the SCPD. Although*
 1102 *the list of XHTMLImageSupported formats MAY be supported within other PDL contexts, there is no requirement*
 1103 *incumbent on the Printer to do so. The image is sent as part of an XHTML-Print document [XHTML-PRINT],*
 1104 *either interleaved within XHTML-Print using the MIME Application/Multiplexed Content Type [MULTIPLEXED]*
 1105 *or as a referenced object.*

1106 *It is **strongly** recommended that images SHOULD be referenced as URI's within the XHTML-Print file and not*
 1107 *interleaved via the Application/Multiplexed Content Type. This allows the Printer to pull swaths of the images as*
 1108 *needed for page composition. The server hosting the image is likewise **strongly** recommended to support HTTP 1.1*
 1109 *Partial Gets, enabling the Printer to pull the specific portions of the images as they are needed. The Printer MAY*
 1110 *retrieve pieces of a single image multiple times to facilitate rotation and other special processing. This approach is*
 1111 *key to achieving broad interoperability across a wide range of product capabilities, as it enables even very low-cost*
 1112 *printers to successfully print a collection of images on a single page.*

1113 *A printer device vendor MAY choose to support other XHTMLImageSupported formats: however, there is no*
 1114 *requirement to support the MIME Application/Multiplexed Content Type [MULTIPLEXED] for these other image*
 1115 *formats.*

1116 *All UPnP printers MUST support at least the 'image/jpeg' image format.*

1117 *Allowed values include all IANA-registered MIME media types for image formats. Vendors MAY extend the*
 1118 *allowed values for this attribute.*

1119 *Note: 'image/jpeg' is registered as a MIME Media Type with IANA.*

1120 **Table 20: allowedValueList for XHTMLImageSupported**

Value	Req. or Opt.
<i>image/jpeg</i>	<u>R</u>
<i><Registered MIME media types for other image formats></i>	<u>O</u>
<i>Vendor-defined</i>	<u>O</u>

1121

1122

1123

1124 **2.7. Eventing and Moderation**1125 **Table 21: 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>	5		<i>N/A</i>
<i>ContentCompleteList</i>	<i>Yes</i>	<i>No</i>	<i>N/A</i>		<i>N/A</i>
<i>JobAbortState</i>	<i>Yes</i>	<i>No</i>	<i>N/A</i>		<i>N/A</i>

1126 ¹ Events containing this variable value SHOULD occur no more often than once every MaxEventRate seconds.1127 ² See 4.4, Eventing: Augmenting the UPnP Template Language in [DEVICE].1128 **2.7.1. Event Model**

1129 The eventing model for the print service has three main purposes.

1130 First is to inform the Control Point when there is a change in condition of the print device. Examples: the
 1131 Printer becomes idle, a paper jam occurs or the Printer is low on paper. The *PrinterState* and
 1132 *PrinterStateReasons* variables provide this information.

1133 Second is for job tracking. Events inform a Control Point when a job is submitted, when all data for the
 1134 job has been received by the Printer, and when a job has completed or been removed from the job queue,
 1135 and whether or not it completed successfully. The *JobIdList*, *ContentCompleteList*, *JobEndState*, and
 1136 *JobAbortState* provide this information. *JobEndState* indicates the final status of each job. It lets Control
 1137 Points know whether it completed successfully or was canceled or aborted.

1138 Third is to inform a Control Point of the progress of the current job. *JobMediaSheetsCompleted* is a
 1139 moderated evented variable that updates an interested Control Point on the number of impressions printed
 1140 for the current job.

1141 2.7.2. Synchronization of Evented Variables

1142 Table 4 below describes how internal printer state changes affect the values of the seven evented state variables,
1143 plus the non-evented variable, JobId. These state changes can be forced by any of: a Control Point invoking one of
1144 the print service actions documented herein, a non-UPnP external action or printer internal events and conditions.
1145 The effect of some non-UPnP external actions is indirect, i.e., they affect internal printer state immediately, but, if
1146 they result in any UPnP-visible effect, the affect appears later. All of these indirect effects have to do with
1147 management of Untracked Jobs. They are included in this table because their ultimate effect can be visible at some
1148 later time. A Control Point should be aware of this to fully understand observed behavior. For PrintEnhanced:1
1149 service implementers, the complete table is a guideline to the information that MUST be kept and how it is
1150 synchronized to guarantee that the externally visible state variables are always correct.

1151 In Table 4, column 1 contains the current value of the variable PrinterState. Column 2 lists the events that can
1152 trigger an internal printer state change. Column 3 gives the new printer state and the complete set of actions taken
1153 by the Printer on the transition that is triggered by the column 2 event. In several cases, the actions taken depend on
1154 other printer conditions in addition to the triggering event. Those situations are identified in the table by dividing
1155 the lower right portion of the corresponding event “cell” into multiple subcells, one for each condition or set of
1156 conditions that requires a different set of transition actions. The upper portion of the event cell is extended into
1157 column 3, signifying that no transition action(s) can be specified for this event except when the conditions in the
1158 event’s subcells are also considered. The word *invisible* in column 3 means there is no state change that could be
1159 observed by a UPnP Control Point. All of the actions listed in column 3 MUST be completed atomically relative to
1160 all external UPnP observations.

1161 For the purposes of this document, atomically means:

- 1162 1. From the viewpoint of any Control Point observer external to the Print Service, all of the values change at
1163 the same time. To achieve this, all evented variables changed by this collected set of actions SHOULD
1164 appear in a single event message.
- 1165 2. It is not possible through any query action for a Control Point to detect that any single state variable has
1166 changed unless it detects that all have changed and been properly updated to their new values.

1167 To help understand the actions, let’s follow one transition through the tables. Find the entry in column 2 “Terminate
1168 active job that was tracked”. Its termination condition, *T*, is one of ‘successful’, ‘canceled’ or ‘aborted’.” Since it has subcell
1169 entries, there is no direct entry in column 3. Assume the normal situation of a busy Printer with more jobs queued
1170 and that all of them are tracked. The relevant added condition is “Next job is tracked.” That takes us to column 3
1171 with actions of “J3, M0, E1(*T*)”. Looking in Table 5 we see that J3 says to remove the first element of the JobIdList and set the
1172 new value of JobId to the new first element of JobIdList. M0 says to reset JobMediaSheetsCompleted to ‘0’ if we track it, or
1173 leave it at ‘-1’ if we don’t. E1(*T*) says to set JobEndState with all the corresponding values for the job just completed, including
1174 whether it was ‘successful’, ‘canceled’ or ‘aborted’. Also note that the M2 value inside JobEndState is set according to the
1175 actual final value of the sheets printed, if known.

1176 NOTE: If the Printer implementation is unable to detect “content complete”, then the ContentCompleteList event is returned at
1177 the same time as the JobEndState event.

1178

1179 **Table 22: Synchronization of Evented Variables**

State	Transition events (and conditions)	Transition actions
?	Initialize PrintEnhanced:1 service	I, R0, J0, M0, E0, A0, C0
idle	CreateJob or CreateJobV2 or CreateURIJob 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 CreateJobV2 or CreateURIJob 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, but for which all job data had not yet been received. Its termination condition, <i>T</i> , is one of 'successful' or 'canceled'.	
	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 active job that was tracked, and for which all job data had been received. Its termination condition, <i>T</i> , is one of 'successful' or 'canceled'.	
	No more jobs.	I, J0, M0, E1(<i>T</i>), C2
	Next job is tracked.	J3, M0, E1(<i>T</i>), C2
	Next job is untracked, and there are no more Tracked Jobs.	J0, M0, E1(<i>T</i>), C2
	Next job is untracked, but there are still Tracked Jobs in the queue.	J4, M0, E1(<i>T</i>), C2
	Terminate Tracked Job that was not active, but for which all job data had not yet been received. Its termination condition, <i>T</i> , is one of 'canceled'.	
	Terminate Tracked Job that was not active, and for which all job data had been received. Its termination condition, <i>T</i> , is one of 'canceled'.	
	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.	
	Abort active job that was tracked, but for which all job data had not yet been received.	
	No more jobs.	I, J0, M0, E1('aborted'), A1(<i>R</i>)
	Next job is tracked.	J3, M0, E1('aborted'), A1(<i>R</i>)
	Next job is untracked, and there are no more Tracked Jobs.	J0, M0, E1('aborted'), A1(<i>R</i>)

	Next job is untracked, but there are still Tracked Jobs in the queue.	J4, M0, E1('aborted'), A1(R)
	Abort active job that was tracked, and for which all job data had been received.	
	No more jobs.	I, J0, M0, E1('aborted'), A1(R), C2
	Next job is tracked.	J3, M0, E1('aborted'), A1(R), C2
	Next job is untracked, and there are no more Tracked Jobs.	J0, M0, E1('aborted'), A1(R), C2
	Next job is untracked, but there are still Tracked Jobs in the queue.	J4, M0, E1('aborted'), A1(R), C2
	Abort Tracked Job that was not active, but for which all job data had not yet been received.	
	J5, E2('aborted'), A2(R)	
	Abort Tracked Job that was not active and for which all job data had been received.	
	J5, E2('aborted'), A2(R), C2	
	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>
	Last byte of data needed to print a job is received.	
	Job is tracked.	C1
	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'), A1
	The active job was lost. It was tracked; the next job is untracked.	S, R1, J4, M0, E1('aborted'), A1
	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.	
	R2	
	CreateJob or CreateJobV2 or CreateURIJob or create non-UPnP Tracked Job	
	JobIdList is empty.	J1, M0
JobIdList is not empty.	J2	
	Create Untracked Job.	<i>invisible</i>

1181 Table 23: Transition Actions Used in Table 19

	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 (§ 2.6.3.30)
	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 newValue=oldValue+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'.
JobAbortState	A0	{}	JobAbortState is initialized to the empty list.

Variable(s) affected			
Label	New variable value(s)	Action Descriptions	
A1(R)	{ id ₁ , JobName_of_id ₁ , JobOriginatingUserName_of_id ₁ , M2, 'aborted', R }	The active job (first element in JobIdList) was aborted. <i>R</i> indicates the reason the job was aborted.	
A2(R)	{ id _i , JobName_of_id _i , JobOriginatingUserName_of_id _i , M2, 'aborted', R }	The job in i th position (i > 1) of JobIdList was aborted. <i>R</i> indicates the reason the job was aborted.	
ContentCompleteList	C0	{ }	ContentCompleteList is initialized to the empty list.
	C1	{ ..., id _i }	Old ContentCompleteList MAY or MAY NOT have been empty. New list has same contents as old list <i>plus</i> one new job added. All data for the job in the i th position (i >= 1) of the JobIdList has been received by the Printer.
	C2	{ ..., ... }	Old ContentCompleteList contained at least one JobId, id _i . The job associated with id _i has completed or been terminated and it is removed from the ContentCompleteList. The new list MAY or MAY NOT be empty.

1182

1183 2.8. Actions

1184 Immediately following this table is detailed information about these actions, including short descriptions of the
 1185 actions, the effects of the actions on state variables, and error codes defined by the actions.

1186 **Table 24: Actions**

Name	Req. or Opt. ¹
<i>CancelJob</i>	<i>R</i>
<i>CreateJob (Deprecated)</i>	<i>R</i>
<i>CreateJobV2</i>	<i>R</i>
<i>CreateURIJob</i>	<i>R</i>
<i>GetJobAttributes</i>	<i>R</i>
<i>GetMargins</i>	<i>R</i>
<i>GetMediaList</i>	<i>R</i>
<i>GetPrinterAttributes (Deprecated)</i>	<i>R</i>
<i>GetPrinterAttributesV2</i>	<i>R</i>
<i>Non-standard actions implemented by a UPnP vendor go here.</i>	<i>X</i>

1187 ¹ R = REQUIRED, O = Optional, X = Non-standard.

1188 Note: the error codes are derived from IPP status codes as follows (see [MODEL] for the detailed definition of each
 1189 error code):

1190 (Client Error minus 400₁₆) convert to decimal + 10 + 700

1191 (Server Error minus 400₁₆) convert to decimal + 60 + 700

1192

1193 Error codes are returned in the <SOAP:Fault> element. A vendor MAY subset or extend these error codes, first by
 1194 supporting additional IPP error codes defined [MODEL] in the UPnP 700 range, and then by supporting private
 1195 error codes in the UPnP 800 range, if no suitable IPP error code exists.

1196 **2.8.1. CancelJob**

1197 *This operation allows a client to cancel a print job from the time the job is created up to the time it is completed,*
 1198 *canceled or aborted.*

1199 **2.8.1.1. Arguments**

1200 **Table 25: Arguments for CancelJob**

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

1201

1202 **2.8.1.2. Errors**

errorCode	errorDescription	Description
Codes 401, 402, 501, 600-99 from the table Error Codes (below)	See the table Error Codes (below)	See the table Error Codes (below)
716	<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>
760	<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>
765	<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>

1203 **2.8.1.3. Effect on State**

1204 *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.,*
 1205 *JobId specified the current job), then JobId is set according to the transition actions described in Section 2.7.2.*

1206 **2.8.2. CreateJob (deprecated)**

1207 *[This action is deprecated in favor of CreateJobV2. See section 2.2.2d)]*

1208 *This action is the first step in submitting a job to the Printer. The Printer returns a unique JobId to identify the job*
 1209 *for this service. The Printer generates the JobId in an implementation-defined manner. The Printer MUST return*
 1210 *values in the range 1 to 2³¹-1; 0 and negative values are invalid. Furthermore, the Printer SHOULD NOT re-use*
 1211 *values recently assigned, since Control Points could confuse such jobs with older jobs.*

1212 *The <allowedValueList> element of the Service Description indicates the values of the parameters that the Print*
 1213 *Service instance (Printer) supports (see section 2.3). The Printer performs the following validation in the indicated*
 1214 *order:*

- 1215 1. If the DocumentFormat is not supported, the Printer MUST reject the request and return the
 1216 ClientErrorDocumentFormatNotSupported (720) error code.
- 1217 2. If the client (Control Point) supplies input parameters that are unsupported or their values are
 1218 unsupported (except DocumentFormat), the Printer (1) MUST accept the CreateJob request, (2) MUST
 1219 ignore or substitute supported values, respectively, and (3) MUST print the job.
- 1220 3. If a client (Control Point) supplies a conflicting combination of MediaSize and MediaType (or any other
 1221 set of IN parameters), the Printer MUST accept the CreateJob request, (2) MUST ignore or substitute the
 1222 conflicting values, and (3) MUST print the job. Whether or not a Printer can detect combinations of
 1223 different parameter values that are not supported, such as combinations of MediaType and MediaSize
 1224 values that are not supported, is IMPLEMENTATION-SPECIFIC. If an implementation does detect
 1225 combinations that are not supported, it substitutes values for one or more parameters to give a
 1226 combination that is supported.

1227 The client (Control Point) MUST send print data to the print service via a separate HTTP Post operation to the
 1228 DataSink URI (see section 2.8.10) returned by the Printer in the CreateJob action response.

1229 **2.8.2.1. Arguments**

1230 **Table 26: Arguments for CreateJob**

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

1231 Section 2.8.2.1 describes the CreateJob action IN/OUT argument’s related state variables. The State Variable
 1232 Table provides a description and data type as well as the allowed and default values.

1233 **2.8.2.2. Errors**

errorCode	errorDescription	Description
Codes 401, 402, 501, 600-99 from the table Error Codes (below)	See the table Error Codes (below)	See the table Error Codes (below)
720	ClientErrorDocumentFormatNotSupported	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.

760	<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>
765	<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>

1234 **2.8.3. CreateJobV2**

1235 *This action is the first step in submitting a job to the Printer. The CreateJobV2 action adds one new IN argument to*
 1236 *those provided by CreateJob:*

- 1237 • *CriticalAttributesList* - *Through the CriticalAttributesList argument, the submitting client has more*
 1238 *control over printer behavior than is available using CreateJob. CreateJobV2 is equivalent to Create Job*
 1239 *if the CriticalAttributesList value is "none".*

1240 *The Printer returns a unique JobId to identify the job for this service. The Printer generates the JobId in an*
 1241 *implementation-defined manner. However, the Printer MUST return values in the range 1 to 2³¹-1; 0 and negative*
 1242 *values are invalid. Furthermore, the Printer SHOULD NOT re-use values recently assigned, since clients (Control*
 1243 *Points) could confuse such jobs with older jobs.*

1244 *The <allowedValueList> element of the Service Description indicates the values of the parameters that the Print*
 1245 *Service instance (Printer) supports (see section 3). The Printer performs the following validation in the indicated*
 1246 *order:*

- 1247 1. *If the DocumentFormat is not supported, the Printer MUST reject the request and return the*
 1248 *ClientErrorDocumentFormatNotSupported (720) error code.*
- 1249 2. *If the client (Control Point) supplies input parameters that are unsupported or their values are*
 1250 *unsupported (except DocumentFormat) then:*
 - 1251 a. *If the unsupported parameters are not included in the CriticalAttributesList, the Printer 1) MUST*
 1252 *accept the CreateJobV2 request, 2) MUST ignore or substitute supported values, respectively, and*
 1253 *3) MUST print the job.*
 - 1254 b. *If the unsupported parameters are included in the CriticalAttributesList, the Printer MUST reject*
 1255 *the request and return the ClientErrorAttributesOrValuesNotSupported (721) error code (unlike*
 1256 *the CreateJob action where the Printer MUST accept the request and process the job).*
- 1257 3. *If a client (Control Point) supplies a conflicting combination of MediaSize and MediaType (or any other*
 1258 *set of IN parameters), and at least one of the conflicting parameters represents a job attribute that is also*
 1259 *included in the parameter CriticalAttributesList, the Printer MUST reject the action and return the*
 1260 *ClientErrorConflictingAttributes (724) error code.*
- 1261 4. *If a client (Control Point) combines "none" with any other value in CriticalAttributesList, the Printer*
 1262 *MUST reject the action and return the ClientErrorConflictingAttributes (724) error code.*
- 1263 5. *If a client (Control Point) supplies "none" in either MediaSize or MediaType then:*
 - 1264 a. *If at least one of the parameters with the value "none" represents a job attribute that is included*
 1265 *in the CriticalAttributesList, the printer MUST reject the action and return the*
 1266 *ClientErrorConflictingAttributes (724) error code.*
 - 1267 b. *If neither MediaSize nor MediaType is a job attribute in the CriticalAttributesList, the Printer 1)*
 1268 *MUST accept the CreateJobV2 request, 2) MUST ignore or substitute supported values,*
 1269 *respectively, and 3) MUST print the job.*

- 1270 6. If a client (Control Point) supplies a combination of *MediaSize* and *MediaType* IN parameter values that
 1271 does not match the Printer’s currently loaded media (see section 2.9.3.2) and the corresponding
 1272 attribute(s) is/are included in *CriticalAttributesList*, the Printer MUST take one of the following actions:
- 1273 a. If the implementation does not support the ‘media-change-request’ *PrinterStateReasons*
 1274 mechanism (see section 2.6.3.31), the Printer MUST reject the action and return the
 1275 *ClientErrorMediaNotLoaded* (734) error code.
 - 1276 b. If the implementation does support the ‘media-change-request’ *PrinterStateReasons* mechanism,
 1277 the Printer MUST accept the request, but not print the job until the requested media is loaded.

1278 The client (Control Point) MUST send print data to the print service via a separate HTTP Post operation to the
 1279 *DataSink* URI (see section 2.8.10).

1280 During job processing, if the Printer encounters a condition in the PDL Data Stream that it cannot honor (for
 1281 example, media-type mismatch) and *Pdl-fidelity* is included in the *CriticalAttributesList*, the Printer MUST abort
 1282 the job and supply the reason for the abort in the evented state variable *JobAbortState*. If the *CriticalAttributesList*
 1283 includes none, the Printer completes processing as it would if the job had been created by *Create Job*.

1284 **2.8.3.1. Arguments**

1285 All relatedStateVariables, except *CriticalAttributesList*, are the same as for the *CreateJob* action.

1286 **Table 27: Arguments for *CreateJobV2***

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

1287 **2.8.3.2. Errors**

Error Code	errorDescription	Description
Codes 401, 402, 501, 600-99 from the table Error Codes (below)	See the table Error Codes (below)	See the table Error Codes (below)

720	<i>ClientErrorDocumentFormatNotSupported</i>	<i>The supplied DocumentFormat parameter value is not supported by the Printer. The Printer 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 IN parameters</i>
721	<i>ClientErrorAttributesOrValuesNotSupported</i>	<i>The DocumentFormat IN parameter value is supported by the Printer, but the client (Control Point) supplied other IN parameter values that are not supported by the Printer, i.e., are not values in the Printer's corresponding <allowedValueList> elements and these IN parameters are included in the CriticalAttributesList.</i>
724	<i>ClientErrorConflictingAttributes</i>	<i>All IN parameter values are supported, but the client (Control Point) supplied some IN parameter values that conflict with other IN parameter values, such as MediaType and MediaSize. (Unlike the PrintBasic:1 service specification, this specification does not permit the Printer to substitute or ignore any IN parameter values that are included in the CriticalAttributesList).</i>
760	<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>
765	<i>ServerErrorTemporaryError</i>	<i>A temporary error such as a memory overflow or a disk full condition 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.</i>
800-899	TBD	Action-specific errors for non-standard actions. Defined by the UPnP vendor.

1288

1289 The Service State Table (Section 2.6.3) describes the CreateJobV2 action IN/OUT arguments related state variables.
1290 The Service State Table provides a description and data type as well as the allowed and default values.

1291 **2.8.4. CreateURIJob**

1292 *This action is very similar to CreateJobV2, but requires the Printer to pull the print data from a specified location*
1293 *rather than requiring the Control Point to push the print data to the Printer. It adds one IN argument to*
1294 *CreateJobV2 —SourceURI, which specifies the location of the document to be retrieved by the Printer. It removes*
1295 *the OUT argument DataSink from CreateJobV2, since no data will be POSTed by the Control Point.*

1296 *The Printer returns a unique JobId to identify the job for this service. The Printer generates the JobId in an*
1297 *implementation-defined manner. However, the Printer MUST return values in the range 1 to 2³¹-1; 0 and negative*
1298 *values are invalid values to be returned as a result of a CreateURIJob action. Furthermore, the Printer SHOULD*
1299 *NOT re-use values recently assigned, since clients (Control Points) would confuse such jobs with older jobs.*

1300 *The <allowedValueList> element of the Service Description indicates the values of the parameters that the Print*
1301 *Service instance (Printer) supports (see section 3). The Printer performs the following validation in the indicated*
1302 *order:*

- 1303 1. *If the DocumentFormat is not supported, the Printer MUST reject the request and return the*
 1304 *ClientErrorDocumentFormatNotSupported (720) error code.*
- 1305 2. *If the client (Control Point) supplies input parameters that are unsupported or their values are*
 1306 *unsupported (except DocumentFormat) then:*
- 1307 a. *If the unsupported parameters are not included in the CriticalAttributesList, the Printer 1)*
 1308 *MUST accept the CreateURIJob request, 2) MUST ignore or substitute supported values,*
 1309 *respectively, and 3) MUST print the job.*
- 1310 b. *If the unsupported parameters are included in the CriticalAttributesList, the Printer MUST*
 1311 *reject the request and return the ClientErrorAttributesOrValuesNotSupported (721) error code*
 1312 *(unlike the CreateJob action where the Printer MUST accept the request and process the job).*
- 1313 3. *If a client (Control Point) supplies a conflicting combination of MediaSize and MediaType (or any other*
 1314 *set of IN parameters), and at least one of the conflicting parameters represents a job attribute that is also*
 1315 *included in the parameter CriticalAttributesList, the Printer MUST reject the action and return the*
 1316 *ClientErrorConflictingAttributes (724) error code.*
- 1317 4. *If a client (Control Point) combines “none” with any other value in CriticalAttributesList, the Printer*
 1318 *MUST reject the action and return the ClientErrorConflictingAttributes (724) error code.*
- 1319 5. *If a client (Control Point) supplies “none” in either MediaSize or MediaType then:*
- 1320 a. *If at least one of the parameters with the value “none” represents a job attribute that is*
 1321 *included in the CriticalAttributesList, the printer MUST reject the action and return the*
 1322 *ClientErrorConflictingAttributes (724) error code.*
- 1323 b. *If neither MediaSize nor MediaType is a job attribute in the CriticalAttributesList, the Printer*
 1324 *1) MUST accept the CreateJobV2 request, 2) MUST ignore or substitute supported values,*
 1325 *respectively, and 3) MUST print the job.*
- 1326 6. *If a client (Control Point) supplies a combination of MediaSize and MediaType IN parameter values that*
 1327 *does not match the Printer’s currently loaded media (see section 2.9.3.2) and the corresponding*
 1328 *attribute(s) is/are included in CriticalAttributesList, the Printer MUST take one of the following actions:*
- 1329 a. *If the implementation does not support the ‘media-change-request’ PrinterStateReasons*
 1330 *mechanism (see 2.6.3.31), the Printer MUST reject the action and return the*
 1331 *ClientErrorMediaNotLoaded (734) error code.*
- 1332 b. *If the implementation does support the ‘media-change-request’ PrinterStateReasons mechanism,*
 1333 *the Printer MUST accept the request, but not print the job until the requested media is loaded.*
- 1334 *The device MUST get the print data via a separate HTTP GET operation to the SourceURI (see section 2.8.11).*
- 1335 *During job processing, if the Printer encounters a condition in the PDL Data Stream that it cannot honor (for*
 1336 *example, in-line side-by-side images exceed its buffer) and image-layout is included in the CriticalAttributesList, the*
 1337 *Printer MUST abort the job and supply the reason for the abort in the evented state variable*
 1338 *A_ARG_TYPE_PrinterAbortReason. Otherwise, if the CriticalAttributesList is none, it completes processing as it*
 1339 *would if the job had been created by CreateJob.*

1340 2.8.4.1. Arguments

1341 All relatedStateVariables, except SourceURI, are the same as for the CreateJobV2 action.

1342 Table 28: Arguments for CreateURIJob

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>CriticalAttributesList</i>	<i>IN</i>	<i>A_ARG_TYPE_CriticalAttribList</i>
<i>SourceURI</i>	<i>IN</i>	<i>SourceURI</i>
<i>JobId</i>	<i>OUT</i>	<i>JobId</i>

1343 **2.8.4.2. Errors**

Error Code	errorDescription	Description
Codes 401, 402, 501, 600-99 from the table Error Codes (below)	See the table Error Codes (below)	See the table Error Codes (below)
720	<i>ClientErrorDocumentFormatNotSupported</i>	<i>The supplied DocumentFormat parameter value is not supported by the Printer. The Printer 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 IN parameters</i>
721	<i>ClientErrorAttributesOrValuesNotSupported</i>	<i>The DocumentFormat IN parameter value is supported by the Printer, but the client (Control Point) supplied other IN parameter values that are not supported by the Printer, i.e., are not values in the Printer's corresponding <allowedValueList> elements and these IN parameters are included in the CriticalAttributesList.</i>
724	<i>ClientErrorConflictingAttributes</i>	<i>All IN parameter values are supported, but the client (Control Point) supplied some IN parameter values that conflict with other IN parameter values, such as MediaType and MediaSize. (Unlike the PrintBasic:1 service specification, this specification does not permit the Printer to substitute or ignore any IN parameter values that are included in the CriticalAttributesList).</i>

760	<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>
765	<i>ServerErrorTemporaryError</i>	<i>A temporary error such as a memory overflow or a disk full condition 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.</i>
800-899	TBD	Action-specific errors for non-standard actions. Defined by the UPnP vendor.

1344

1345 The Service State Table (Section 2.6.3) describes the CreateURIJob action IN/OUT arguments related state
 1346 variables. The Service State Table provides a description and data type as well as the allowed and default values.

1347 **2.8.5. GetJobAttributes**

1348 *The GetJobAttributes action allows a client (Control Point) to determine some of the values of job-related variables*
 1349 *of the specified job with a JobId from 1 to 2³¹-1. Only active and queued jobs can be queried since only these jobs*
 1350 *are maintained in the JobIdList variable. These variables allow end users to identify their job (i.e., "JobName",*
 1351 *"JobOriginatingUserName"). Other information can be derived from the GetJobAttributes action.*

1352 *If the specified job is found, its parameters are returned whether the job is active or queued. If the specified job is*
 1353 *not found, the ClientErrorNotFound (716) is returned. Any job not found either never existed or has reached its*
 1354 *terminating state (i.e., completed, cancelled, aborted) and is no longer known to the Print Service. If the value of*
 1355 *JobMediaSheetsCompleted is greater than 0, the referenced job is active and the Printer has physically completed*
 1356 *printing and stacking the number of media sheets indicated. If the value of JobMediaSheetsCompleted is 0 or -1,*
 1357 *the client can determine whether the referenced job is active according to whether it is the first entry in JobIdList.*
 1358 *The value of JobIdList can be retrieved either from its most recent event value or from the action*
 1359 *GetPrinterAttributes.*

1360 **2.8.5.1. Arguments**

1361 **Table 29: 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>

1362 **2.8.5.2. Errors**

errorCode	errorDescription	Description
Codes 401, 402, 501, 600-99 from the table Error Codes (below)	See the table Error Codes (below)	See the table Error Codes (below)

716	<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>
760	<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>
765	<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>

1363

1364 **2.8.6. GetMargins**

1365 *The GetMargins action allows a client (Control Point) to determine:*

- 1366 • *The Non-Printable Area (see section 2.2.2j) for a specified MediaSize and MediaType combination.*
- 1367 • *Valid combinations of MediaType and MediaSize that the Printer supports.*
- 1368 • *Whether or not full-bleed printing is supported for the associated media size / type combination; i.e.,*
1369 *whether or not the Printer is capable of printing one or more photos that cover the entire surface of one*
1370 *side of the medium sheet with no white edges (or more accurately, media-colored edges).*

1371 *Because of mechanical tolerances in printer media loading and feeding mechanisms, a Printer might not be able to*
1372 *print right up to the edge of the medium, or might be unable to accurately position objects very close to the edge of*
1373 *the medium. The Printer informs the CP of the extent of this Non-Printable Area with the OUT parameter*
1374 *PageMargins. (See section 2.6.3.26) The Control Point can then position all content in 'safe' or reliably*
1375 *reproducible regions to ensure the Printer can correctly render the job.*

1376 *If the Printer returns zero for all four margins, the CP SHOULD assume the Printer has no unprintable region,*
1377 *and that the Printer can appropriately render output consisting of arbitrary content positioned anywhere on the*
1378 *surface of the medium. In this case, the Printer is obviously also capable of full-bleed output; therefore, when the*
1379 *Printer returns all zeros as its PageMargins, the OUT argument FullBleedSupported SHOULD be ignored.*

1380 *Full-bleed content MAY include multiple and/or overlaid images, and MAY include simple annotation so long as*
1381 *the annotation is not positioned within the Printer's Non-Printable Area. If full-bleed content is sent to a Printer*
1382 *which does not report zero PageMargins and returns FullBleedSupported as 'false', results are implementation*
1383 *specific.*

1384 *If the client (Control Point) supplies an unsupported combination of the MediaType and MediaSize IN parameters,*
1385 *the Printer MUST reject the action and return the ClientErrorConflictingAttributes (724) error code.*

1386 *If a client (Control Point) supplies "none" in either MediaType or MediaSize IN parameters, the Printer MUST*
1387 *reject the action and return the ClientErrorConflictingAttributes (724) error code.*

1388 *A client (Control Point) MAY supply the 'device-setting' value for one or both of the IN arguments, in which case*
1389 *the Printer MUST use the corresponding values in its SCPD <defaultValue> entry.*

1390 *Note: This action does not provide any way for the Control Point (client) to determine what media is currently*
1391 *loaded or whether the current media has run out.*

1392 **Example 1:** *The Control Point wishes to print a "borderless" or full-bleed 4 inch by 6 inch photo. It sends a*
1393 *GetMargin action with MediaSize set to custom_photo_4x6in and MediaType set to photographic.*

- 1394 **Case 1a:** The Printer returns *PageMargins* of 0mm,0mm,2mm,0mm and *FullBleedSupported=false*.
 1395 The page margins indicate the Printer can reliably position print content right up to the top, right, and left
 1396 edges of the medium, and up to 2 mm from the bottom edge of the medium. Since these values are not all
 1397 zero, the CP MUST look at *FullBleedSupported*, and determines that this Printer is not capable of
 1398 generating full-bleed output. The CP offers the user a choice of printing with a white border around the
 1399 photo or canceling the print request.
- 1400 **Case 1b:** The Printer returns *PageMargins* of 0mm,0mm,0mm,.2mm and *FullBleedSupported=true*.
 1401 The page margins indicate the Printer can reliably position print content right up to the top, right, and left
 1402 edges of the medium, and up to .2 mm from the bottom edge of the medium. Since these values are not all
 1403 zero, the CP MUST look at *FullBleedSupported*, and determines that this Printer is capable of generating
 1404 full-bleed output. The CP creates the full-bleed job and the Printer renders it successfully.
- 1405 **Case 1c:** The Printer returns *PageMargins* of 0mm,0mm,0mm,0mm and *FullBleedSupported=false*.
 1406 The page margins indicate the Printer can reliably position print content right up to the top, right, bottom,
 1407 and left edges of the medium. Since these values are all zero, the CP knows that the Printer can reliably
 1408 position arbitrary content anywhere on the medium surface; it need not look at *FullBleedSupported*, and
 1409 determines that this Printer is capable of generating full-bleed output. The CP creates the full-bleed job
 1410 and the Printer renders it successfully.
- 1411 **Example 2:** The Control Point wishes to print a collection of images with text, and wants to use the maximum area
 1412 of the medium surface that can be reliably utilized by the Printer. It sends a *GetMargin* action with
 1413 *MediaSize* set to *device-setting* and *MediaType* set to *device-setting*.
- 1414 **Case 2a:** The Printer's default *MediaSize* is *custom_photo_4x6in* and its default *MediaType* is *photographic*. The
 1415 Printer returns *PageMargins* of 0mm,0mm,2mm,0mm and *FullBleedSupported=false*. The Control Point
 1416 generates XHTML-Print content containing:
 1417

```
<style type="text/css">
```
- 1418

```
    @page { size: auto; margin: 0mm 0mm 2mm; }
```
- 1419

```
    ...
```
- 1420

```
</style>
```
- 1421 The Control Point uses relative sizing and positioning to lay out the document. The Printer generates the
 1422 appropriate output on 4x6 photo paper.
- 1423 **Case 2b:** The Printer's default *MediaSize* is *iso_a4_210x297mm* and its default *MediaType* is *stationery*. The
 1424 Printer returns *PageMargins* of 0in,.25in,.5in,.25in and *FullBleedSupported=true*. The Control Point
 1425 generates XHTML-Print content containing:
 1426

```
<style type="text/css">
```
- 1427

```
    @page { size: auto; margin: 0in .25in .5in .25in; }
```
- 1428

```
    ...
```
- 1429

```
</style>
```
- 1430 Otherwise, the Control Point sends the same content as generated for Case 2a above. The Printer generates
 1431 the appropriate output on size A4 plain paper.
- 1432

1433 2.8.6.1. Arguments

1434 Table 30: Arguments for *GetMargins*

Argument	Direction	relatedStateVariable
<i>MediaSize</i>	<i>IN</i>	<i>MediaSize</i>
<i>MediaType</i>	<i>IN</i>	<i>MediaType</i>
<i>PageMargins</i>	<i>OUT</i>	<i>PageMargins</i>
<i>FullBleedSupported</i>	<i>OUT</i>	<i>FullBleedSupported</i>

1435

1436 **2.8.6.2. Errors**

Error Code	errorDescription	Description
Codes 401, 402, 501, 600-99 from the table Error Codes (below)	See the table Error Codes (below)	See the table Error Codes (below)
721	<i>ClientErrorAttributesOrValuesNot Supported</i>	<i>The request is rejected because the client (Control Point) supplied some IN parameter values that are not supported by the Printer, i.e., are not values in the corresponding Printer's <allowedValueList> elements.</i>
724	<i>ClientErrorConflictingAttributes</i>	<i>All IN parameter values are supported, but the client (Control Point) supplied IN MediaSize and MediaType parameter values that conflict with each other (i.e., the combination is not a supported combination) or are not allowed with this action. (This specification does not permit the Printer to substitute values or ignore such conflicts).</i>
760	<i>ServerErrorInternalError</i>	<i>The Printer encountered an unexpected condition that prevented it from fulfilling the request. This error differs from "ServerErrorTemporaryError" in that it implies a more permanent type of internal error.</i>
765	<i>ServerErrorTemporaryError</i>	<i>A temporary error such as a memory overflow or a disk full condition 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.</i>
800-899	TBD	Action-specific errors for non-standard actions. Defined by the UPnP vendor.

1437

1438 **2.8.6.3. Effect of Action on State**1439 *This action does not affect the state in any way.*

1440 **2.8.7. GetMediaList**

1441 *The GetMediaList action allows a client (Control Point) to determine valid combinations of MediaType and*
 1442 *MediaSize that the Printer supports. One or both of the IN arguments MediaType and MediaSize MUST be*
 1443 *specified as "none". If both are specified as "none", the Printer returns a list of lists that gives all supported media*
 1444 *type and size combinations. The format of the output is shown under A_ARG_TYPE_MediaList in section 2.6.3.2.*
 1445 *If either of MediaType or MediaSize is specified as anything other than "none", it MUST contain a valid value from*
 1446 *the MediaType or MediaSize allowedValueList, respectively. If both MediaType and MediaSize are specified as*
 1447 *anything other than "none", the Printer MUST reject the action and return the ClientErrorConflictingAttributes*
 1448 *(724) error code.*

1449 *When MediaSize is specified as values other than "none" and MediaType is "none", the OUT parameter contains a*
 1450 *list of supported MediaTypes for that MediaSize. The format of the output list is as shown in Example 1 under*
 1451 *A_ARG_TYPE_MediaList in section 2.6.3.2.*

1452 *When MediaType is specified as values other than "none" and MediaSize is "none", the OUT parameter contains*
 1453 *a list of supported MediaSizes for that MediaType. The format of the output list is as shown in Example 2 under*
 1454 *A_ARG_TYPE_MediaList in section 2.6.3.2.*

1455 *A client (Control Point) could supply the 'device-setting' value for either but not both of the IN arguments. In this*
 1456 *case the Printer MUST use the corresponding value in its SCPD <defaultValue> entry, and return the list of sizes*
 1457 *or types supported for that value. See the example in section 2.6.3.2.*

1458 *Note: This action does not provide any way for the Control Point (client) to determine what media size or type is*
 1459 *currently loaded or whether the current media has run out.*

1460 **2.8.7.1. Arguments**

1461 **Table 31: Arguments for GetMediaList**

Argument	Direction	relatedStateVariable
MediaSize	IN	MediaSize
MediaType	IN	MediaType
MediaList	OUT	A_ARG_TYPE_MediaList

1462

1463 **2.8.7.2. Errors**

Error Code	errorDescription	Description
Codes 401, 402, 501, 600-99 from the table Error Codes (below)	See the table Error Codes (below)	See the table Error Codes (below)
721	ClientErrorAttributesOrValues NotSupported	The request is rejected because the client (Control Point) supplied some IN parameter values that are not supported by the Printer, i.e., are not values in the corresponding Printer's <allowedValueList> elements.
724	ClientErrorConflictingAttributes	All IN parameter values are supported, but the client (Control Point) supplied values other than "none" for both MediaSize and MediaType IN parameter.
760	ServerErrorInternalError	The Printer encountered an unexpected condition that

		<i>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>
765	<i>ServerErrorTemporaryError</i>	<i>A temporary error such as a memory overflow or a disk full condition 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.</i>
800-899	TBD	Action-specific errors for non-standard actions. Defined by the UPnP vendor.

1464

1465 **2.8.7.3. Effect of Action on State**

1466 *This action does not affect the state in any way.*

1467 **2.8.8. GetPrinterAttributes (deprecated)**

1468 [This action is deprecated in favor of GetPrinterAttributesV2. See section 2.2.2d)]

1469 *The GetPrinterAttributes action allows a client (Control Point) to determine the state of the Printer and values of*
 1470 *certain state variables that represent Printer attributes. In particular, the Control Point can determine the number*
 1471 *of pending jobs. The Control Point can also determine the state of the Print Service, and which job, if any, is the*
 1472 *current job.*

1473 *The JobId OUT argument is the JobId of the current job; i.e., the job that has caused the PrinterState variable to be*
 1474 *'processing' or 'stopped'. The JobId MUST be the first JobId in the JobIdList or 0. If there is no current job, i.e.,*
 1475 *the PrinterState is 'idle' (there are no jobs, or all jobs are pending or held), then JobId contains a 0 which is an*
 1476 *invalid JobId for a job). If JobId is 0, the Printer is either idle OR a non-UPnP job is printing (and the Printer*
 1477 *implementation has chosen NOT to display non-UPnP jobs, i.e., the job is an Untracked Job).*

1478

1479 *Note: The GetPrinterAttributes action does not allow a client to discover the supported values of standard*
 1480 *attributes. The client can discover what is supported from the <allowedValueList> element in the Service*
 1481 *Description (see section 3). Neither does the GetPrinterAttributes action allow a client to discover vendor added*
 1482 *attributes. Vendors MUST define their own private actions to return such additional attributes.*

1483 **2.8.8.1. Arguments**

1484 **Table 32: 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>

1485

1486 **2.8.8.2. Errors**

errorCode	errorDescription	Description
Codes 401, 402, 501, 600-99 from the table Error Codes (below)	See the table Error Codes (below)	See the table Error Codes (below)
760	<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>
765	<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>

1487 **2.8.9. GetPrinterAttributesV2**

1488 *The GetPrinterAttributesV2 action allows a client (Control Point) to determine various aspects of the Printer's*
 1489 *current state, including all information returned by GetPrinterAttributes plus an indication of whether or not the*
 1490 *Printer currently has an active connection to the internet. When a Control Point invokes this action, the Printer*
 1491 *SHOULD make an immediate attempt to determine the state of its Internet connection. Once the state is*
 1492 *determined, the Printer sets the value of the state variable InternetConnectState and returns the newly determined*
 1493 *value, along with the PrinterState, PrinterStateReasons, JobIdList, and JobId, as described for GetPrinterAttributes*
 1494 *in section 2.8.8.*

1495 *Note: After this query, there is no guarantee how long the Printer's internet connection status will remain*
 1496 *unchanged.*

1497 **2.8.9.1. Arguments**1498 **Table 33: Arguments for GetPrinterAttributesV2**

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>
<i>InternetConnectState</i>	<i>OUT</i>	<i>InternetConnectState</i>

1499

1500 **2.8.9.2. Errors**

Error Code	errorDescription	Description
Codes 401, 402, 501, 600-99 from the table Error Codes (below)	See the table Error Codes (below)	See the table Error Codes (below)
760	<i>ServerErrorInternalError</i>	<i>The Printer encountered an unexpected condition that prevented it from fulfilling the request. This error</i>

		<i>differs from "server-error-temporary-error" in that it implies a more permanent type of internal error.</i>
765	<i>ServerErrorTemporaryError</i>	<i>A temporary error such as a memory overflow or a disk full condition 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.</i>
800-899	TBD	Action-specific errors for non-standard actions. Defined by the UPnP vendor.

1501

1502 *2.8.9.3. Effect of Action on State*1503 *This action does not affect the state in any way.*1504 **2.8.10. HTTP POST**

1505 The client (Control Point) sends the print data using an HTTP [HTTP] Post operation (with chunking if desired), to
 1506 the URI returned as the DataSink output parameter of the CreateJob* actions. Having received this DataSink URI
 1507 in the Create Job* response, the client MUST then open a connection to the device using the URI and send the data.

1508 The client MUST open the data connection on the DataSink URI within 30 seconds after receiving the CreateJob*
 1509 response. Otherwise, the Printer MUST time out, discard jobs for which no data has been received, and remove its
 1510 JobId from the JobIdList variable. If no data at all is received for a job then the Printer SHOULD delete the job
 1511 after a wait of at least 30 seconds and remove its JobId from the JobIdList variable. If data has been received for a
 1512 job but a subsequent chunked HTTP POST operation does not arrive for an implementation-defined period of time
 1513 (at least 30 seconds) then the data received so far is printed and the A_ARG_TYPE_PrinterAbortReason is set to
 1514 *external-access-http-error*. If the Printer receives an HTTP Post for the DataSink URI after the timeout period, the
 1515 Printer returns the HTTP 408 (Request Timeout) status code, if the job still exists, otherwise, the HTTP 404 (Not
 1516 Found) status code.

1517 If the Printer accepts the CreateJob* action, but subsequently cannot accept the HTTP Post (because it is too busy or
 1518 is accepting another job), the Printer MUST reject the HTTP Post and return the HTTP 503 (Service Unavailable).
 1519 The Printer SHOULD reset the timer to 30 seconds or some other implementation-specific value and SHOULD
 1520 return that value in the Retry-After HTTP header in the error response. The Printer SHOULD ensure that the Retry-
 1521 After value is less than the maximum amount of time before which the device will timeout. If the Printer does not
 1522 return Retry-After header, the HTTP spec [HTTP] says that the Control Point assumes an HTTP 500 error (internal
 1523 server error) and no retry is allowed and the Printer aborts the job.

1524 An event will be sent to the client whenever the JobId is removed from the JobIdList.

1525 The URI MUST be a valid HTTP URI [HTTP]. The Printer MUST support HTTP/1.1 chunking [HTTP] for the
 1526 Post operation. The client MUST send the DocumentFormat MIME Media Type value in the HTTP Content-Type
 1527 header (or the 'unknown' special value, if the client doesn't know the actual document format - see section 2.6.3.11).
 1528 If the DocumentFormat value does not match the HTTP Content-Type header value, the Printer MUST reject the
 1529 request and return the HTTP 409 (Conflict) status code.

1530 **2.8.11. HTTP GET**

1531 The Printer retrieves print data using an HTTP [HTTP] GET operation (with Range headers for a partial GET, if
 1532 desired) to the URI received as the SourceURI IN parameter of the CreateURIJob action or to a URI specified
 1533 within the print content of a job. Having received this SourceURI in the CreateURIJob request or having processed

1534 a request for retrieval of information from a URI within the PDL, the Printer MUST then open a connection to the
 1535 server indicated by the URI and request the data.

1536 The Printer MUST open the data connection on the SourceURI within 30 seconds after the job becomes the current
 1537 job (*i.e.*, within 30 seconds of issuing the event notification which placed the target job at the top of the JobIdList).

1538 If no data at all is received for the job within 30 seconds of issuing the GET request, then the Printer SHOULD
 1539 delete the job and remove its JobId from the JobIdList variable. The job is considered aborted, and JobAbortState is
 1540 updated appropriately, triggering an event notification. If data has been received for a job but a subsequent HTTP
 1541 GET response does not arrive for an implementation-defined period of time (at least 30 seconds) then the data
 1542 received so far is printed and the job is aborted. If the Printer receives an HTTP GET response after the timeout
 1543 period, the Printer SHOULD ignore the response and discard the data.

1544 An event will be sent to the client whenever the JobId is removed from the JobIdList.

1545 The URI MUST conform to RFC 2396 or RFC 2732.

1546 If a Content-Type header is not included in the GET response, the Printer SHOULD assume the content type
 1547 matches the DocumentFormat provided in the CreateURIJob action. If no Content-Type header is provided and the
 1548 CreateURIJob indicated the DocumentFormat is 'unknown', then if the Printer supports the *application/octet-*
 1549 *stream* document format, it MUST process the data as such; else the Printer SHOULD abort the job and set the
 1550 A_ARG_TYPE_PrinterAbortReason to *external-access-doc-format-err*. If a Content-Type header is returned
 1551 which is in conflict with the DocumentFormat provided in the CreateURIJob action, a Printer which supports the
 1552 *application/octet-stream* format SHOULD process the job as such; otherwise the Printer SHOULD assume the
 1553 content type matches the Content-Type header. NOTE: This case is intended to cover the situation where the
 1554 document is being retrieved from a non-UPnP Client (such as a web server) serving the content.

1555 If the print data indicates that multiple objects or images are to be composed on the page, and the Printer requires
 1556 support for partial GETs to achieve such layout, and the HTTP server at the targeted URI does not implement partial
 1557 GET capability, then:

- 1558 • If reformatting the print output will not compromise any attribute included in the CriticalAttributesList,
 1559 then the print output SHOULD be reformatted and the job completed normally.
- 1560 • If reformatting the output would conflict with an attribute indicated as critical, the Printer MUST abort the
 1561 job and, if supported, set the A_ARG_TYPE_PrinterAbortReason to *external-access-http-error*.

2.8.12. Error Codes

1562 The following table lists the possible error codes to actions for this service type. If an action results in multiple
 1563 errors, the most specific error SHOULD be returned.
 1564

Table 34: Error Codes

errorCode	errorDescription	Description
400-499		See UPnP Device Architecture section on Control.
500-599		See UPnP Device Architecture section on Control.
600-699		Common action errors. Defined by UPnP Forum Technical Committee. See UPnP Device Architecture section on Control.
716	<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>

errorCode	errorDescription	Description
720	ClientErrorDocumentFormatNotSupported	<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>
721	ClientErrorAttributesOrValuesNotSupported	<i>The DocumentFormat IN parameter value is supported by the Printer, but the client (Control Point) supplied other IN parameter values that are not supported by the Printer, i.e., are not values in the Printer's corresponding <allowedValueList> elements and these IN parameters are included in the CriticalAttributesList.</i>
724	ClientErrorConflictingAttributes	<i>All IN parameter values are supported, but the client (Control Point) supplied some IN parameter values that conflict with other IN parameter values, such as MediaType and MediaSize. (Unlike the PrintBasic:1 service specification, this specification does not permit the Printer to substitute or ignore any IN parameter values that are included in the CriticalAttributesList).</i>
760	ServerErrorInternalError	<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>
765	ServerErrorTemporaryError	<i>A temporary error such as a memory overflow or a disk full condition 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.</i>
800-899	TBD	<i>(Specified by UPnP vendor.)</i>

1566 2.9. Theory of Operation

1567 The UPnP Printer device ('Printer') has one REQUIRED service called PrintBasic:1 Service. A UPnP device which
1568 supports printing MUST support PrintBasic:1 and MAY support other optional services. As an example, this might
1569 include basic power functions and a banner printing service.

1570 2.9.1. The Print Model

1571 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
1572 started printing and when it has finished printing. In addition, a user can cancel a previously submitted job. Also a
1573 client (Control Point) can determine which Create* action parameter values a Print Service implementation supports
1574 using the values returned in the <allowedValueList> element of the Service Description.

1575 Enhanced feature support is available through the inclusion of optional actions and SST variables.

1576 2.9.2. Jobs

1577 The Print Service's main task is to accept print jobs from clients, queue them up (if the Printer is capable of
1578 handling more than one job at a time) and then print them. A job is identified by an integer, the JobId, which is

1579 allocated by the device. The [MODEL] describes the rules for JobId production (1 to 2**31-1). The JobId is
1580 returned by the Create* actions.

1581 The set of jobs that a Printer has in its queue is exposed in a very simple way.

- 1582 o The complete list of known jobs is made available as a state variable represented as CSV list (see section
1583 2.4.1) called **JobIdList**.
- 1584 o All waiting jobs appear in the **JobIdList** variable-- even those that the device has decided not to print for
1585 some reason (they are in the IPP 'pending' or 'pending-held' job state). All UPnP jobs are considered
1586 Tracked Jobs.
- 1587 o The order of jobs in the **JobIdList** variable indicates the order in which the jobs will be initiated.
- 1588 o The job that is actually printing at the moment (or for which the Print Service is paused) is called the
1589 current job. If the current job is "Tracked" (see 2.2.2o) its job identifier is stored in the **JobId** Print
1590 Service state variable and that same JobId value is also the first **JobId** in the **JobIdList**. If there is no
1591 current job, i.e., there are no jobs, or all jobs are pending or held, or an Untracked Job (Section 2.2.2 p) is
1592 printing, the **JobId** is 0.
- 1593 o Once a job has been printed (or cancelled or aborted) it no longer appears in the **JobIdList**, whether or not
1594 the Printer has any other jobs to print.
- 1595 o When the Print Service has no Tracked Jobs to print, the **JobIdList** state variable is an empty string.

1596 2.9.3. Job Processing

1597 2.9.3.1. Intent of a Print Job

1598 The intent of a Print job is indicated by the job attributes as represented by either:

- 1599 - the IN parameters of the Create* action and/or
- 1600 - the print instructions in the PDL Data stream.

1601 Many job attributes MAY be specified by either or both methods. This section defines the precedence between
1602 these two representations of the intent of a print job.

1603 2.9.3.1.1. Production vs. Layout Job Attributes

1604 This specification distinguishes two classes of such job attributes—*Production* and *Layout*. A Layout Job Attribute
1605 is one that is inherent to the print output and cannot be overridden by IN parameters when the job is created. A
1606 Production Job Attribute is one that can reasonably change at the different times when the job is printed without
1607 affecting important job characteristics. Obvious examples of Production Attributes are number of copies, number of
1608 sides and number of logical pages per physical sheet of paper, provided that when such Production Attributes are
1609 represented in the PDL Data Stream they are represented as print instructions. However, if number of copies or
1610 number of logical pages per physical sheet of paper is represented by repetitions of the PDL Data Stream, instead of
1611 a print instruction in the PDL Data Stream, such a representation is not considered a Production Job Attribute and so
1612 an IN parameter does not override such a representation.

1613 Job attributes are partitioned between Production and Layout as follows:

1614 Production Job Attributes (Job Attributes takes precedence):
1615 JobName
1616 JobOriginatingUserName
1617 Copies

1618 Sides
 1619 NumberUp
 1620 PrintQuality
 1621

1622 Layout Job Attributes (data stream takes precedence):
 1623 OrientationRequested
 1624 MediaSize
 1625 MediaType

1626 **2.9.3.1.2. Precedence of Production vs. Layout Job Attributes**

1627 The Control Point MUST supply an allowed value for each of the IN parameters defined for the Create* action.
 1628 The PDL Data Stream MAY also have a value for any Production or Layout attribute represented as a print
 1629 instruction. The Control Point MAY supply the Distinguished Value defined by this document for each IN
 1630 parameter to request the Printer to use its <defaultValue> value (see section 2.2.2 e) and section 2.6.2) in case the
 1631 corresponding print instruction in the PDL Data Stream is absent. The Printer SHOULD take the following action,
 1632 depending on the values supplied by the Control Point in the Create* IN parameter and provided in the PDL Data
 1633 Stream, for each given job attribute:

1634 **Table 35: 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) **

1635 ** Only when both are supplied does the precedence depend on whether the attribute is a Production
 1636 Attribute or a Layout Attribute. Production IN parameters take precedence, while Layout PDL print
 1637 instructions take precedence.

1638 NOTE: Even for Layout Attributes, the IN parameter value supplied in the Create* action will be used as long as
 1639 no overriding value is found in the PDL Data Stream itself.

1640 **2.9.3.2. Critical Attributes and the Intent of a Print Job**

1641 PrintEnhanced:1 offers CPs two distinct approaches to satisfying print job intent. In both cases, the Printer is
 1642 expected to honor input values for all print job attributes to the best of its ability. This includes both those attributes

1643 specified in the initiating Create* action and those found in the PDL as it is processed. Where the approaches differ
1644 is when the Printer encounters an attribute that it cannot satisfy. For jobs initiated by CreateJob, the single most
1645 important (implicit) job attribute is “content on paper”. This does not mean that the Printer is allowed to ignore
1646 explicit attribute requests, but it does give the Printer significant freedom to select an alternative value when it
1647 cannot satisfy any particular attribute request. In contrast, for jobs initiated by CreateJobV2 or CreateURIJob, the
1648 Control Point tells the Printer exactly which attributes are critical for successful output. For those critical attributes,
1649 when the Printer detects that it cannot faithfully render the output according to the attribute value, it MUST abort
1650 the job immediately.

1651 The most important reason to use CreateJobV2 or CreateURIJob with a CriticalAttributesList specified as any
1652 value other than “none” is to avoid wasting expensive paper and ink (or other marking material) for printed output
1653 that the end user would consider unacceptable. The Printer MUST meet the following four requirements:

- 1654 1. The Printer MUST inform the Control Point as to which Critical Attributes it supports (i.e., the set of job
1655 attributes whose settings the Printer is capable of detecting at print time and comparing to corresponding
1656 values requested by the submitting Control Point.) These are the Critical Attributes defined in Section
1657 2.2.2c). It is permitted that the set of critical attributes supported by the Printer is the NULL set (specified
1658 as the “none” value).
- 1659 2. The Printer MUST be able to abort a print job when it cannot satisfy one or more of the Critical Attributes
1660 submitted in the print request. The Printer SHOULD do the best job it can with respect to all other print
1661 job attributes not designated by the Control Point to be critical.
- 1662 3. The Printer MUST allow Control Points to select which of those attributes, if any, it considers critical for
1663 any given print job. Processing details for Critical Attributes are described below with the state variable
1664 CriticalAttributesSupported and the actions CreateJobV2 and CreateURIJob.
- 1665 4. If the Printer does abort a job due to Critical Attribute mismatch, the Printer MUST inform the Control
1666 Point of the type of the attribute (e.g., MediaSize, MediaType) whose mismatch resulted in the abort.

1667 For each job created using the action CreateJobV2 or CreateURIJob, the invoking Control Point gives the
1668 CriticalAttributesList as input. If the Printer discovers at any time that it cannot satisfy the requested value for an
1669 attribute in that input list, the Printer MUST abort that job.

1670 Critical Attributes are subject to the production versus layout precedence rules defined in section 2.9.3.1.2. That is,
1671 a critical attribute which is a production attribute such as *sides* is considered satisfied when the IN parameter for
1672 *sides* can be honored, whether or not there is a conflicting instruction in the PDL data stream. A critical attribute
1673 which is a layout attribute such as MediaSize is considered satisfied at job creation if the IN MediaSize requested
1674 can be honored; however, if the PDL data stream requests a different media size, the size requested by the PDL
1675 MUST be honored or the job aborted (assuming MediaSize is in the Printer’s CriticalAttributesSupported values.)

1676 The Printer SHOULD only attempt to verify the value of a Critical Attribute at the time in printing when the
1677 attribute matters to physical output. This is the time when a mismatch between requested and actual values for the
1678 attribute would produce incorrect output if the job proceeds. At that time, before aborting the job, the Printer MAY
1679 use any means it deems appropriate to “correct” the Printer’s inappropriate value, including asking for user
1680 intervention.

1681 Example 1:

1682 Printer A has an optical media type sensor that can determine whether it has plain paper, transparency, matte or
1683 glossy photo paper, etc., loaded. Printer B has no media type sensor, but it has a front panel selector that allows the
1684 user to “tell” the Printer the media type that is loaded. Printer C has no sensor and no front panel selector for media
1685 type. For printers A and B, media type *could be* considered a Critical Attribute. It is vendor choice whether *media-*
1686 *type* is included in the allowedValueList for CriticalAttributesSupported. For printer C, media type is not detectable
1687 and therefore cannot be included in the allowedValueList for CriticalAttributesSupported.

1688 Assume both printers A and B have included *media-type* in their respective CriticalAttributesSupported
1689 allowedValueLists, and the invoking Control Point also includes it in the CriticalAttributesList IN parameter. Upon
1690 receipt of a CreateJobV2 request with an attribute of media type set to glossy photo, printers A and B MUST verify
1691 that glossy photo is available for use (Printer A by sensing and Printer B via its front panel UI); if not available, they
1692 MUST either issue a *media-change-request* and wait for glossy photo to become available, or abort the job. Printer
1693 C SHOULD assume it doesn't know and proceed to print the job. If either printer A or B excludes *media-type* from
1694 CriticalAttributesSupported, or if it is included there but the invoking Control Point excludes *media-type* from
1695 CriticalAttributesList input to CreateJobV2 or CreateURIJob, then the printer behaves the same as printer C.

1696 Example 2:

1697 Printer D has a sensor in its paper tray that measures the length of media in its tray, but it has no way of knowing
1698 the length of a manually fed sheet of paper. Printer E cannot sense media in the tray, but does detect the trailing
1699 edge of a sheet as the sheet moves toward the print head. Printer E can determine the length of a sheet of paper, but
1700 only after it has printed the contents of the whole physical page. For printer D, page length is detectable when
1701 loading from the tray, but not detectable when feeding from the manual slot. For printer E, page length is
1702 detectable, even though it cannot verify the value until *after* the page content is printed. When the Control Point
1703 includes *media-size* in its CriticalAttributesList for either printer, the printer MUST abort the job as soon as it
1704 detects a mismatch. For printer D, that would be at the time it prepares to load a sheet for printing. For printer E,
1705 that would be at the end of the first page. (While this example is useful to clarify the treatment of Critical
1706 Attributes, printer D would likely not support *media-size* as a CriticalAttribute, since it cannot detect *media-size* on
1707 manual feed.)

1708 The Printer is considered to satisfy the intent of a job when the value of every attribute included in the
1709 CriticalAttributesList matches the job's utilized value for that attribute. Any attributes in the Printer's
1710 allowedValueList for CriticalAttributesSupported but *not* included by the Control Point in the CriticalAttributesList
1711 submitted with the CreateJobV2 or CreateURIJob action are processed on a best-effort basis and MUST NOT cause
1712 the job to be aborted.

1713 This means that a request for plain paper can be considered met when photo paper is loaded if the Printer has no
1714 way of verifying its loaded media type. This allows manufacturers of lower cost printers to still take advantage of
1715 aborting jobs they know they can't meet intent for, without demanding that everything be detectable and included in
1716 CriticalAttributesSupported. For example, even if the Printer can't verify media type, it MAY still be able to verify
1717 media width. Even if it can't verify media width, it can verify that a request to print an 8" x 10" image on a
1718 requested media size of 4" x 6" cannot be met, and therefore SHOULD be aborted.

1719 This implicit acceptance of non-detectable attributes still allows manufacturers to add value by detecting more
1720 attributes and exposing them in CriticalAttributesSupported.

1721 Finally, the Printer is NOT REQUIRED to know the value of Critical Attributes before marking paper. If the job
1722 requests US legal size paper, but the Printer doesn't know it has US letter size paper until it reaches the bottom of
1723 the letter-size sheet, that is acceptable. As soon as the Printer does discover that the sheet is short, though, it
1724 considers the intent unmet and MUST abort the job immediately if it was created by the CreateJobV2 or
1725 CreateURIJob action with *media-size* in the CriticalAttributesList.

1726 **2.9.4. Side-by-side Images**

1727 Side-by-side images SHOULD be supported as specified in the XHTML-Print data without any reformatting. Side-
1728 by-side images MUST be supported when the images are "included by reference" (see XHTML-Print specification
1729 [XHTML-PRINT] section 4.4). If side-by-side images cannot be printed without reformatting when the job is
1730 created by CreateJobV2 or CreateURIJob, and '*image-layout*' is included in the CriticalAttributesList, the job
1731 MUST be aborted.

1732 2.9.5. Actions

1733 The following actions MUST be supported by conforming PrintEnhanced:1 Service implementations:

- 1734 ○ **Create Job (Deprecated)**. This action is used to submit a job to the Printer. The allocated JobId is
1735 returned.
- 1736 ○ **CancelJob**. This can be used to cancel a job using the JobId.
- 1737 ○ **GetPrinterAttributes (Deprecated)**. This action can be used to query some of the Printer attributes.
- 1738 ○ **GetJobAttributes**. This action can be used to query some of the job attributes of a specified job.
- 1739 ○ **CreateJobV2**. This action is used to submit a job to the Printer and the Printer MUST honor all supplied
1740 IN parameter values or reject the action. The allocated JobId is returned.
- 1741 ○ **CreateURIJob**. This action is the same as CreateJobV2, except that rather than pushing the print data to
1742 the Printer, the Control Point provides a SourceURI from which the Printer pulls the print data.
- 1743 ○ **GetMargins**. This action returns the four widths of the margins between the four edges and the edge of
1744 the printable area for the requested combination of MediaType and MediaSize, along with an indication as
1745 to whether the Printer supports full-bleed printing for the MediaSize / MediaType combination.
- 1746 ○ **GetMediaList**. This action returns the supported media sizes for a particular media type, or the supported
1747 media types for a particular media size, or a matrix of all types and sizes supported.
- 1748 ○ **GetPrinterAttributesV2**. Similar to GetPrinterAttributes, this action extends the set of printer attributes
1749 returned to include an indication of whether or not the Printer is currently connected to the internet.

1750 2.9.6. Events

1751 One of the primary goals of this specification is to allow a user to know when their print job has started and when it
1752 has finished. The UPnP eventing mechanism can be used for this purpose. There are seven evented state variables:
1753 JobIdList, JobEndState, PrinterState, PrinterStateReasons, JobMediaSheetsCompleted, JobAbortState, and
1754 ContentCompleteList that MAY change whenever a job stops or starts. A client implementation SHOULD
1755 therefore subscribe to UPnP events from the print service in order to monitor the progress of a job. A Control Point
1756 can determine when a particular job that it submitted has started printing by matching the first entry of the evented
1757 JobIdList variable with the JobId value returned to it by the Create* action. Similarly a Control Point can determine
1758 that a job has completed, whether successful or not, by matching the JobId for that job with the first element of the
1759 evented JobEndState and JobAbortState variables.

1760 Five of the seven evented variables are also available as OUT parameters of GetPrinterAttributes, GetJobAttributes,
1761 or GetPrinterAttributesV2; so a Control Point can obtain their values by polling. However, the JobEndState and
1762 JobAbortState are not OUT parameters of any action, so they are only available to a client by eventing, not by
1763 polling.

1764 2.9.7. Security

1765 In keeping with the lightweight approach to security taken by UPnP no security is defined by this specification.

1766 If a vendor decides to include some form of security they are strongly encouraged to utilize IPsec as defined by the
1767 IETF.

1768 2.9.8. Localization

1769 A UPnP printer is assumed to be operating within the locale of the user. No other localization mechanism is defined
1770 for the Print Service. The Control Point (client) is expected to localize the well-known string values (that

1771 correspond to IPP keyword values) to the locale of its user. The Control Point (client) is expected to convert the
 1772 enum integer values to human readable string values in the locale of the user.

1773 **2.9.9. IPP Data Type mapping to UPnP Data Types**

1774 Basic IPP data types are transformed as follows.

1775 **Table 36: 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>
dateTime	dateTime.tz

1776 The derived types in IPP are mapped onto the following UPnP data types.

1777 **Table 37: 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).

1778

1779 **Table 38: Structured Data Type mapping**

IPP Type	UPnP equivalent
----------	-----------------

resolution	This is represented as a pair of integers <Attribute Name>X and <attribute Name>Y
1setOf X	See the earlier discussion on arrays in section 2.4.1.

1780

1781 **2.9.10. Improving Output Consistency for XHTML-Print**

1782 Since the output of XHTML-Print [XHTML-PRINT] and CSS-Print [CSSPP] onto paged media (e.g., printed-
 1783 paper) is different from the output display on screen media, pagination needs to be considered. This, along with the
 1784 fact that some ambiguity exists in the interpretation of XHTML and CSS leads to various inconsistent outputs
 1785 among output devices (e.g., printers), which is unexpected.

1786 In supporting XHTML-Print [XHTML-PRINT] and CSS-Print [CSSPP] as a document format, compliance to
 1787 XHTML-PRINT/CSS Print Profile Guidelines for PrintEnhanced:1 [XPCSSGUIDE] is strongly recommended.
 1788 This guideline provides information for both printers and content creators to help achieve an improved level of
 1789 output consistency on print media among printers which support XHTML-Print and CSS-Print.

1790 In addition, in order to achieve a high level of consistency, sample templates for simple photo layouts is provided as
 1791 a reference in [XPCSSGUIDE] for both printer implementations and content authors.

1792 3. XML Service Description

1793 The following SCPD is intended as an example and vendors should adjust values based on their product specific
 1794 implementation. An exception to this is for Certification Tool testing where the SCPD provided below must match
 1795 the AllowedValues listed below.

```

1796 <?xml version="1.0"?>
1797 <scpd xmlns="urn:schemas-upnp-org:service-1-0">
1798   <specVersion>
1799     <major>1</major>
1800     <minor>0</minor>
1801   </specVersion>
1802   <actionList>
1803     <action>
1804       <name>CancelJob</name>
1805       <argumentList>
1806         <argument>
1807           <name>JobId</name>
1808           <direction>in</direction>
1809           <relatedStateVariable>JobId</relatedStateVariable>
1810         </argument>
1811       </argumentList>
1812     </action>
1813     <action>
1814       <name>CreateJob</name>
1815       <argumentList>
1816         <argument>
1817           <name>JobName</name>
1818           <direction>in</direction>
1819           <relatedStateVariable>JobName</relatedStateVariable>
1820         </argument>
1821         <argument>
1822           <name>JobOriginatingUserName</name>
1823           <direction>in</direction>
1824           <relatedStateVariable>JobOriginatingUserName</relatedStateVariable>
1825         </argument>
1826         <argument>
1827           <name>DocumentFormat</name>
1828           <direction>in</direction>
1829           <relatedStateVariable>DocumentFormat</relatedStateVariable>
1830         </argument>
1831         <argument>
1832           <name>Copies</name>
1833           <direction>in</direction>
1834           <relatedStateVariable>Copies</relatedStateVariable>
1835         </argument>
1836         <argument>
1837           <name>Sides</name>
1838           <direction>in</direction>
1839           <relatedStateVariable>Sides</relatedStateVariable>
1840         </argument>
1841         <argument>
1842           <name>NumberUp</name>
1843           <direction>in</direction>
1844           <relatedStateVariable>NumberUp</relatedStateVariable>

```

```

1845     </argument>
1846     <argument>
1847         <name>OrientationRequested</name>
1848         <direction>in</direction>
1849         <relatedStateVariable>OrientationRequested</relatedStateVariable>
1850     </argument>
1851     <argument>
1852         <name>MediaSize</name>
1853         <direction>in</direction>
1854         <relatedStateVariable>MediaSize</relatedStateVariable>
1855     </argument>
1856     <argument>
1857         <name>MediaType</name>
1858         <direction>in</direction>
1859         <relatedStateVariable>MediaType</relatedStateVariable>
1860     </argument>
1861     <argument>
1862         <name>PrintQuality</name>
1863         <direction>in</direction>
1864         <relatedStateVariable>PrintQuality</relatedStateVariable>
1865     </argument>
1866     <argument>
1867         <name>JobId</name>
1868         <direction>out</direction>
1869         <relatedStateVariable>JobId</relatedStateVariable>
1870     </argument>
1871     <argument>
1872         <name>DataSink</name>
1873         <direction>out</direction>
1874         <relatedStateVariable>DataSink</relatedStateVariable>
1875     </argument>
1876 </argumentList>
1877 </action>
1878 <action>
1879 <name>CreateJobV2</name>
1880 <argumentList>
1881 <argument>
1882 <name>JobName</name>
1883 <direction>in</direction>
1884 <relatedStateVariable>JobName</relatedStateVariable>
1885 </argument>
1886 <argument>
1887 <name>JobOriginatingUserName</name>
1888 <direction>in</direction>
1889 <relatedStateVariable>JobOriginatingUserName</relatedStateVariable>
1890 </argument>
1891 <argument>
1892 <name>DocumentFormat</name>
1893 <direction>in</direction>
1894 <relatedStateVariable>DocumentFormat</relatedStateVariable>
1895 </argument>
1896 <argument>
1897 <name>Copies</name>
1898 <direction>in</direction>
1899 <relatedStateVariable>Copies</relatedStateVariable>

```



```

1900     </argument>
1901     <argument>
1902         <name>Sides</name>
1903         <direction>in</direction>
1904         <relatedStateVariable>Sides</relatedStateVariable>
1905     </argument>
1906     <argument>
1907         <name>NumberUp</name>
1908         <direction>in</direction>
1909         <relatedStateVariable>NumberUp</relatedStateVariable>
1910     </argument>
1911     <argument>
1912         <name>OrientationRequested</name>
1913         <direction>in</direction>
1914         <relatedStateVariable>OrientationRequested</relatedStateVariable>
1915     </argument>
1916     <argument>
1917         <name>MediaSize</name>
1918         <direction>in</direction>
1919         <relatedStateVariable>MediaSize</relatedStateVariable>
1920     </argument>
1921     <argument>
1922         <name>MediaType</name>
1923         <direction>in</direction>
1924         <relatedStateVariable>MediaType</relatedStateVariable>
1925     </argument>
1926     <argument>
1927         <name>PrintQuality</name>
1928         <direction>in</direction>
1929         <relatedStateVariable>PrintQuality</relatedStateVariable>
1930     </argument>
1931     <argument>
1932         <name>CriticalAttributesList</name>
1933         <direction>in</direction>
1934     <relatedStateVariable>A_ARG_TYPE_CriticalAttribList</relatedStateVariable>
1935     </argument>
1936     <argument>
1937         <name>JobId</name>
1938         <direction>out</direction>
1939         <relatedStateVariable>JobId</relatedStateVariable>
1940     </argument>
1941     <argument>
1942         <name>DataSink</name>
1943         <direction>out</direction>
1944         <relatedStateVariable>DataSink</relatedStateVariable>
1945     </argument>
1946 </argumentList>
1947 </action>
1948 <action>
1949     <name>CreateURIJob</name>
1950     <argumentList>
1951         <argument>
1952             <name>JobName</name>
1953             <direction>in</direction>
1954             <relatedStateVariable>JobName</relatedStateVariable>

```

```

1955     </argument>
1956     <argument>
1957         <name>JobOriginatingUserName</name>
1958         <direction>in</direction>
1959         <relatedStateVariable>JobOriginatingUserName</relatedStateVariable>
1960     </argument>
1961     <argument>
1962         <name>DocumentFormat</name>
1963         <direction>in</direction>
1964         <relatedStateVariable>DocumentFormat</relatedStateVariable>
1965     </argument>
1966     <argument>
1967         <name>Copies</name>
1968         <direction>in</direction>
1969         <relatedStateVariable>Copies</relatedStateVariable>
1970     </argument>
1971     <argument>
1972         <name>Sides</name>
1973         <direction>in</direction>
1974         <relatedStateVariable>Sides</relatedStateVariable>
1975     </argument>
1976     <argument>
1977         <name>NumberUp</name>
1978         <direction>in</direction>
1979         <relatedStateVariable>NumberUp</relatedStateVariable>
1980     </argument>
1981     <argument>
1982         <name>OrientationRequested</name>
1983         <direction>in</direction>
1984         <relatedStateVariable>OrientationRequested</relatedStateVariable>
1985     </argument>
1986     <argument>
1987         <name>MediaSize</name>
1988         <direction>in</direction>
1989         <relatedStateVariable>MediaSize</relatedStateVariable>
1990     </argument>
1991     <argument>
1992         <name>MediaType</name>
1993         <direction>in</direction>
1994         <relatedStateVariable>MediaType</relatedStateVariable>
1995     </argument>
1996     <argument>
1997         <name>PrintQuality</name>
1998         <direction>in</direction>
1999         <relatedStateVariable>PrintQuality</relatedStateVariable>
2000     </argument>
2001     <argument>
2002         <name>CriticalAttributesList</name>
2003         <direction>in</direction>
2004     <relatedStateVariable>A_ARG_TYPE_CriticalAttribList</relatedStateVariable>
2005     </argument>
2006     <argument>
2007         <name>SourceURI</name>
2008         <direction>in</direction>
2009         <relatedStateVariable>SourceURI</relatedStateVariable>

```

```

2010     </argument>
2011 <argument>
2012     <name>JobId</name>
2013     <direction>out</direction>
2014     <relatedStateVariable>JobId</relatedStateVariable>
2015 </argument>
2016 </argumentList>
2017 </action>
2018 <action>
2019 <name>GetJobAttributes</name>
2020 <argumentList>
2021 <argument>
2022     <name>JobId</name>
2023     <direction>in</direction>
2024     <relatedStateVariable>JobId</relatedStateVariable>
2025 </argument>
2026 <argument>
2027     <name>JobName</name>
2028     <direction>out</direction>
2029     <relatedStateVariable>JobName</relatedStateVariable>
2030 </argument>
2031 <argument>
2032     <name>JobOriginatingUserName</name>
2033     <direction>out</direction>
2034     <relatedStateVariable>JobOriginatingUserName</relatedStateVariable>
2035 </argument>
2036 <argument>
2037     <name>JobMediaSheetsCompleted</name>
2038     <direction>out</direction>
2039     <relatedStateVariable>JobMediaSheetsCompleted</relatedStateVariable>
2040 </argument>
2041 </argumentList>
2042 </action>
2043 <action>
2044 <name>GetMargins</name>
2045 <argumentList>
2046 <argument>
2047     <name>MediaSize</name>
2048     <direction>in</direction>
2049     <relatedStateVariable>MediaSize</relatedStateVariable>
2050 </argument>
2051 <argument>
2052     <name>MediaType</name>
2053     <direction>in</direction>
2054     <relatedStateVariable>MediaType</relatedStateVariable>
2055 </argument>
2056 <argument>
2057     <name>PageMargins</name>
2058     <direction>out</direction>
2059     <relatedStateVariable>PageMargins</relatedStateVariable>
2060 </argument>
2061 <argument>
2062     <name>FullBleedSupported</name>
2063     <direction>out</direction>
2064     <relatedStateVariable>FullBleedSupported</relatedStateVariable>

```

```

2065     </argument>
2066   </argumentList>
2067 </action>
2068 <action>
2069   <name>GetMediaList</name>
2070   <argumentList>
2071     <argument>
2072       <name>MediaSize</name>
2073       <direction>in</direction>
2074       <relatedStateVariable>MediaSize</relatedStateVariable>
2075     </argument>
2076     <argument>
2077       <name>MediaType</name>
2078       <direction>in</direction>
2079       <relatedStateVariable>MediaType</relatedStateVariable>
2080     </argument>
2081     <argument>
2082       <name>MediaList</name>
2083       <direction>out</direction>
2084       <relatedStateVariable>A_ARG_TYPE_MediaList</relatedStateVariable>
2085     </argument>
2086   </argumentList>
2087 </action>
2088 <action>
2089   <name>GetPrinterAttributes</name>
2090   <argumentList>
2091     <argument>
2092       <name>PrinterState</name>
2093       <direction>out</direction>
2094       <relatedStateVariable>PrinterState</relatedStateVariable>
2095     </argument>
2096     <argument>
2097       <name>PrinterStateReasons</name>
2098       <direction>out</direction>
2099       <relatedStateVariable>PrinterStateReasons</relatedStateVariable>
2100     </argument>
2101     <argument>
2102       <name>JobIdList</name>
2103       <direction>out</direction>
2104       <relatedStateVariable>JobIdList</relatedStateVariable>
2105     </argument>
2106     <argument>
2107       <name>JobId</name>
2108       <direction>out</direction>
2109       <relatedStateVariable>JobId</relatedStateVariable>
2110     </argument>
2111   </argumentList>
2112 </action>
2113 <action>
2114   <name>GetPrinterAttributesV2</name>
2115   <argumentList>
2116     <argument>
2117       <name>PrinterState</name>
2118       <direction>out</direction>
2119       <relatedStateVariable>PrinterState</relatedStateVariable>

```

```

2120     </argument>
2121     <argument>
2122         <name>PrinterStateReasons</name>
2123         <direction>out</direction>
2124         <relatedStateVariable>PrinterStateReasons</relatedStateVariable>
2125     </argument>
2126     <argument>
2127         <name>JobIdList</name>
2128         <direction>out</direction>
2129         <relatedStateVariable>JobIdList</relatedStateVariable>
2130     </argument>
2131     <argument>
2132         <name>JobId</name>
2133         <direction>out</direction>
2134         <relatedStateVariable>JobId</relatedStateVariable>
2135     </argument>
2136     <argument>
2137         <name>InternetConnectState</name>
2138         <direction>out</direction>
2139         <relatedStateVariable>InternetConnectState</relatedStateVariable>
2140     </argument>
2141 </argumentList>
2142 </action>
2143 </actionList>
2144 <serviceStateTable>
2145     <stateVariable sendEvents="no">
2146         <name>A_ARG_TYPE_CriticalAttribList</name>
2147         <dataType>string</dataType>
2148         <defaultValue></defaultValue>
2149     </stateVariable>
2150     <stateVariable sendEvents="no">
2151         <name>A_ARG_TYPE_MediaList</name>
2152         <dataType>string</dataType>
2153         <defaultValue></defaultValue>
2154     </stateVariable>
2155     <stateVariable sendEvents="no">
2156         <name>A_ARG_TYPE_PrinterAbortReason</name>
2157         <dataType>string</dataType>
2158         <defaultValue></defaultValue>
2159         <allowedValueList>
2160             <allowedValue>hardware-error</allowedValue>
2161             <allowedValue>external-access-uri-not-found</allowedValue>
2162             <allowedValue>external-access-object-failure</allowedValue>
2163             <allowedValue>external-access-doc-format-err</allowedValue>
2164             <allowedValue>external-access-http-error</allowedValue>
2165         </allowedValueList>
2166     </stateVariable>
2167     <stateVariable sendEvents="no">
2168         <name>CharRepSupported</name>
2169         <dataType>string</dataType>
2170         <defaultValue></defaultValue>
2171         <allowedValue>iana_iso_8859-1</allowedValue>
2172         <allowedValue>iana_Shift_JIS</allowedValue>
2173         <allowedValue>unicode_katakana</allowedValue>
2174     </stateVariable>

```

```

2175 <stateVariable sendEvents="no">
2176   <name>ColorSupported</name>
2177   <dataType>boolean</dataType>
2178   <defaultValue></defaultValue>
2179 </stateVariable>
2180 <stateVariable sendEvents="yes">
2181   <name>ContentCompleteList</name>
2182   <dataType>string</dataType>
2183   <defaultValue></defaultValue>
2184 </stateVariable>
2185 <stateVariable sendEvents="no">
2186   <name>Copies</name>
2187   <dataType>i4</dataType>
2188   <defaultValue>1</defaultValue>
2189   <allowedValueRange>
2190     <minimum>0</minimum>
2191     <maximum>2147483647</maximum>
2192     <step>1</step>
2193   </allowedValueRange>
2194 </stateVariable>
2195 <stateVariable sendEvents="no">
2196   <name>CriticalAttributesSupported</name>
2197   <dataType>string</dataType>
2198   <defaultValue></defaultValue>
2199   <allowedValueList>
2200     <allowedValue>none</allowedValue>
2201     <allowedValue>copies</allowedValue>
2202     <allowedValue>sides</allowedValue>
2203     <allowedValue>number-up</allowedValue>
2204     <allowedValue>orientation-requested</allowedValue>
2205     <allowedValue>media-size</allowedValue>
2206     <allowedValue>media-type</allowedValue>
2207     <allowedValue>print-quality</allowedValue>
2208     <allowedValue>text-layout</allowedValue>
2209     <allowedValue>image-layout</allowedValue>
2210     <allowedValue>image-orientation</allowedValue>
2211     <allowedValue>pdl-fidelity</allowedValue>
2212     <allowedValue>font-family</allowedValue>
2213     <allowedValue>font-size</allowedValue>
2214   </allowedValueList>
2215 </stateVariable>
2216 <stateVariable sendEvents="no">
2217   <name>DataSink</name>
2218   <dataType>uri</dataType>
2219   <defaultValue></defaultValue>
2220 </stateVariable>
2221 <stateVariable sendEvents="no">
2222   <name>DeviceId</name>
2223   <dataType>string</dataType>
2224   <defaultValue></defaultValue>
2225 </stateVariable>
2226 <stateVariable sendEvents="no">
2227   <name>DocumentFormat</name>
2228   <dataType>string</dataType>
2229   <defaultValue></defaultValue>

```

```

2230     <allowedValueList>
2231         <allowedValue>unknown</allowedValue>
2232         <allowedValue>application/xhtml-print</allowedValue>
2233         <allowedValue>application/xhtml-print-e</allowedValue>
2234         <allowedValue>application/octet-stream</allowedValue>
2235         <allowedValue>text/plain</allowedValue>
2236         <allowedValue>text/plain;charset=utf-8</allowedValue>
2237         <allowedValue>application/postscript</allowedValue>
2238         <allowedValue>application/vnd.hp-PCL</allowedValue>
2239     </allowedValueList>
2240 </stateVariable>
2241 <stateVariable sendEvents="no">
2242     <name>DocumentUTF16Supported</name>
2243     <dataType>string</dataType>
2244     <defaultValue></defaultValue>
2245     <allowedValueList>
2246         <allowedValue>none</allowedValue>
2247         <allowedValue>all</allowedValue>
2248         <allowedValue>application/xhtml-print</allowedValue>
2249         <allowedValue>application/xhtml-print-e</allowedValue>
2250         <allowedValue>application/octet-stream</allowedValue>
2251         <allowedValue>text/plain</allowedValue>
2252         <allowedValue>text/plain;charset=utf-8</allowedValue>
2253         <allowedValue>application/postscript</allowedValue>
2254         <allowedValue>application/vnd.hp-PCL</allowedValue>
2255     </allowedValueList>
2256 </stateVariable>
2257 <stateVariable sendEvents="no">
2258     <name>FullBleedSupported</name>
2259     <dataType>boolean</dataType>
2260     <defaultValue></defaultValue>
2261 </stateVariable>
2262 <stateVariable sendEvents="no">
2263     <name>InternetConnectState</name>
2264     <dataType>string</dataType>
2265     <defaultValue></defaultValue>
2266     <allowedValueList>
2267         <allowedValue>unknown</allowedValue>
2268         <allowedValue>connected</allowedValue>
2269         <allowedValue>not-connected</allowedValue>
2270     </allowedValueList>
2271 </stateVariable>
2272 <stateVariable sendEvents="yes">
2273     <name>JobAbortState</name>
2274     <dataType>string</dataType>
2275     <defaultValue></defaultValue>
2276 </stateVariable>
2277 <stateVariable sendEvents="yes">
2278     <name>JobEndState</name>
2279     <dataType>string</dataType>
2280     <defaultValue></defaultValue>
2281 </stateVariable>
2282 <stateVariable sendEvents="no">
2283     <name>JobId</name>
2284     <dataType>i4</dataType>

```

```

2285     <defaultValue>0</defaultValue>
2286     <allowedValueRange>
2287         <minimum>0</minimum>
2288         <maximum>2147483647</maximum>
2289         <step>1</step>
2290     </allowedValueRange>
2291 </stateVariable>
2292 <stateVariable sendEvents="yes">
2293     <name>JobIdList</name>
2294     <dataType>string</dataType>
2295     <defaultValue></defaultValue>
2296 </stateVariable>
2297 <stateVariable sendEvents="yes">
2298     <name>JobMediaSheetsCompleted</name>
2299     <dataType>i4</dataType>
2300     <defaultValue>0</defaultValue>
2301     <allowedValueRange>
2302         <minimum>-1</minimum>
2303         <maximum>2147483647</maximum>
2304         <step>1</step>
2305     </allowedValueRange>
2306 </stateVariable>
2307 <stateVariable sendEvents="no">
2308     <name>JobName</name>
2309     <dataType>string</dataType>
2310     <defaultValue></defaultValue>
2311 </stateVariable>
2312 <stateVariable sendEvents="no">
2313     <name>JobOriginatingUserName</name>
2314     <dataType>string</dataType>
2315     <defaultValue></defaultValue>
2316 </stateVariable>
2317 <stateVariable sendEvents="no">
2318     <name>MediaSize</name>
2319     <dataType>string</dataType>
2320     <defaultValue></defaultValue>
2321     <allowedValueList>
2322         <allowedValue>none</allowedValue>
2323         <allowedValue>om_small-photo_100x150mm</allowedValue>
2324         <allowedValue>na_letter_8.5x11in</allowedValue>
2325         <allowedValue>na_legal_8.5x14in</allowedValue>
2326         <allowedValue>iso_a4_210x297mm</allowedValue>
2327         <allowedValue>iso_c5_162x229mm</allowedValue>
2328         <allowedValue>iso_dl_110x220mm</allowedValue>
2329         <allowedValue>jis_b4_257x364mm</allowedValue>
2330         <allowedValue>device-setting</allowedValue>
2331     </allowedValueList>
2332 </stateVariable>
2333 <stateVariable sendEvents="no">
2334     <name>MediaType</name>
2335     <dataType>string</dataType>
2336     <defaultValue></defaultValue>
2337     <allowedValueList>
2338         <allowedValue>none</allowedValue>
2339         <allowedValue>stationery</allowedValue>

```



```

2340     <allowedValue>stationery-inkjet</allowedValue>
2341     <allowedValue>transparency</allowedValue>
2342     <allowedValue>envelope</allowedValue>
2343     <allowedValue>labels</allowedValue>
2344     <allowedValue>photographic</allowedValue>
2345     <allowedValue>photographic-glossy</allowedValue>
2346     <allowedValue>photographic-matte</allowedValue>
2347     <allowedValue>cardstock</allowedValue>
2348     <allowedValue>device-setting</allowedValue>
2349   </allowedValueList>
2350 </stateVariable>
2351 <stateVariable sendEvents="no">
2352   <name>NumberUp</name>
2353   <dataType>string</dataType>
2354   <defaultValue>1</defaultValue>
2355   <allowedValueList>
2356     <allowedValue>1</allowedValue>
2357     <allowedValue>2</allowedValue>
2358     <allowedValue>4</allowedValue>
2359     <allowedValue>device-setting</allowedValue>
2360   </allowedValueList>
2361 </stateVariable>
2362 <stateVariable sendEvents="no">
2363   <name>OrientationRequested</name>
2364   <dataType>string</dataType>
2365   <defaultValue>portrait</defaultValue>
2366   <allowedValueList>
2367     <allowedValue>portrait</allowedValue>
2368     <allowedValue>landscape</allowedValue>
2369     <allowedValue>reverse-landscape</allowedValue>
2370     <allowedValue>reverse-portrait</allowedValue>
2371     <allowedValue>device-setting</allowedValue>
2372   </allowedValueList>
2373 </stateVariable>
2374 <stateVariable sendEvents="no">
2375   <name>PageMargins</name>
2376   <dataType>string</dataType>
2377   <defaultValue></defaultValue>
2378 </stateVariable>
2379 <stateVariable sendEvents="no">
2380   <name>PrinterLocation</name>
2381   <dataType>string</dataType>
2382   <defaultValue></defaultValue>
2383 </stateVariable>
2384 <stateVariable sendEvents="no">
2385   <name>PrinterName</name>
2386   <dataType>string</dataType>
2387   <defaultValue></defaultValue>
2388 </stateVariable>
2389 <stateVariable sendEvents="no">
2390   <name>PrintQuality</name>
2391   <dataType>string</dataType>
2392   <defaultValue>normal</defaultValue>
2393   <allowedValueList>
2394     <allowedValue>draft</allowedValue>

```

```

2395     <allowedValue>normal</allowedValue>
2396     <allowedValue>high</allowedValue>
2397     <allowedValue>device-setting</allowedValue>
2398   </allowedValueList>
2399 </stateVariable>
2400 <stateVariable sendEvents="yes">
2401   <name>PrinterState</name>
2402   <dataType>string</dataType>
2403   <defaultValue>idle</defaultValue>
2404   <allowedValueList>
2405     <allowedValue>idle</allowedValue>
2406     <allowedValue>processing</allowedValue>
2407     <allowedValue>stopped</allowedValue>
2408   </allowedValueList>
2409 </stateVariable>
2410 <stateVariable sendEvents="yes">
2411   <name>PrinterStateReasons</name>
2412   <dataType>string</dataType>
2413   <defaultValue>none</defaultValue>
2414   <allowedValueList>
2415     <allowedValue>none</allowedValue>
2416     <allowedValue>attention-required</allowedValue>
2417     <allowedValue>media-jam</allowedValue>
2418     <allowedValue>paused</allowedValue>
2419     <allowedValue>door-open</allowedValue>
2420     <allowedValue>media-low</allowedValue>
2421     <allowedValue>media-empty</allowedValue>
2422     <allowedValue>output-area-almost-full</allowedValue>
2423     <allowedValue>output-area-full</allowedValue>
2424     <allowedValue>marker-supply-low</allowedValue>
2425     <allowedValue>marker-supply-empty</allowedValue>
2426     <allowedValue>marker-failure</allowedValue>
2427     <allowedValue>media-change-request</allowedValue>
2428   </allowedValueList>
2429 </stateVariable>
2430 <stateVariable sendEvents="no">
2431   <name>Sides</name>
2432   <dataType>string</dataType>
2433   <defaultValue>one-sided</defaultValue>
2434   <allowedValueList>
2435     <allowedValue>one-sided</allowedValue>
2436     <allowedValue>two-sided-long-edge</allowedValue>
2437     <allowedValue>two-sided-short-edge</allowedValue>
2438     <allowedValue>device-setting</allowedValue>
2439   </allowedValueList>
2440 </stateVariable>
2441 <stateVariable sendEvents="no">
2442   <name>SourceURI</name>
2443   <dataType>uri</dataType>
2444   <defaultValue></defaultValue>
2445 </stateVariable>
2446 <stateVariable sendEvents="no">
2447   <name>XHTMLImageSupported</name>
2448   <dataType>string</dataType>
2449   <defaultValue>image/jpeg</defaultValue>

```

```
2450     <allowedValueList>
2451         <allowedValue>image/jpeg</allowedValue>
2452     </allowedValueList>
2453 </stateVariable>
2454 </serviceStateTable>
2455 </scpd>
```