PrintEnhanced:1 Service Template Version 1.01

For UPnP™ Version 1.0
Status: *Standardized DCP*
Date: May 4, 2005

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.

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

Contents

List of Tables

168

1. Overview and Scope

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

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

172 This service-type enables the following functions:

173 • *Printing using both "push" and "pull" models:*

174   • Control Point MAY push the print document using HTTP POST.

175   • Control Point MAY provide a URI and request the print service to pull the print document from that
176   location using HTTP GET.

177 • *Enhanced Layout Printing: Allows precise positioning and size capability, box properties, EXIF file format,*
178   *etc. (for more details, see Enhanced Layout Extension Conformance, section 2.4 of XHTML-Print* [XHTML-
179   PRINT] *and section 2.1 of CSS Print Profile* [CSSPP]*).*

180 • *Flexible Job Control with respect to User Intents: CreateJobV2 and CreateURIJob allow the Control Point to*
181   *request a job be printed either in a 'best effort' manner or if and only if all "critical" aspects of the job request*
182   *can be honored by the Printer.*

2. Service Modeling Definitions

2.1. Service Type

185 A service that is compliant with this specification is identified with the following service type: **urn:schemas-upnp-**
186 **org:service:***PrintEnhanced:1*.

2.2. Terminology and Notations

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

2.2.1. Conformance Terminology

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

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

194   b) MUST NOT - This phrase means that the definition is an absolute prohibition of the specification.

195   c) SHOULD - This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in
196   particular circumstances to ignore a particular item, but the full implications MUST be understood and
197   carefully weighed before choosing a different course.

198   d) SHOULD NOT - This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid
199   reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full
200   implications should be understood and the case carefully weighed before implementing any behavior
201   described with this label.

202   e)   MAY - This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may
203   choose to include the item because a particular marketplace requires it or because the vendor feels that it
204   enhances the product while another vendor may omit the same item. An implementation which does not
205   include a particular option MUST be prepared to interoperate with another implementation which does
206   include the option, though perhaps with reduced functionality. An implementation which does include a
207   particular option MUST be prepared to interoperate with another implementation which does not include
208   the option

209   2.2.2. Other Terminology

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

213   a)   Comma Separated Value (CSV) - a variable that contains multiple string values separated by the US-ASCII
214   COMMA (',') character (see section 2.4.1).

215   b)   Content Complete — A job is said to be Content Complete when the Printer holds all information
216   necessary to finish printing the job—it will not need to access any more external data. For example, an
217   XHTML-Print job MUST satisfy two conditions to be content complete. First, the Printer will have
218   fetched the complete source document object and all objects that are referenced either directly by URIs in
219   the source or indirectly by URIs in previously referenced objects. Second, all remaining unprinted content
220   from these objects is locally buffered by the Printer and will not be released until it has been printed or
221   canceled.

222   c)   Critical Attribute – a print job attribute whose value the Printer can determine at print time and that a
223   Control Point is allowed to declare as critical to the successful completion of a print job. Specifically,
224   when a Control Point indicates that a particular attribute is Critical and the Printer is unable to satisfy the
225   requested value for that attribute at print time, the Printer MUST abort the job. The value of the Critical
226   Attribute can either be directly detectable by the Printer or it MAY be supplied by implementation-defined
227   means that are outside the scope of this specification, such as a user-controlled front panel setting. See
228   section 2.9.3.2

229   d)   Deprecated – A construct which is deprecated is targeted for obsolescence from the PrintBasic:1 service
230   specification. It SHOULD NOT be used by Control Points for new applications or extended functionality.
231   Since PrintEnhanced:1 is a superset of PrintBasic:1, the deprecated construct MUST be supported by
232   Printers conforming to the PrintEnhanced:1 service.

233   e)   Distinguished Value – a special value defined by this specification for some action IN parameters. Use of
234   Distinguished Value IN parameter allows a Page Description Language (PDL) Data Stream corresponding
235   value to take effect when it would normally be overridden by the IN parameter. In the case where the
236   Distinguished Value is absent in the PDL data stream and the IN parameter value is specified as 'device-
237   setting', the Service uses its <defaultValue> value for the IN parameter.   See section 2.6.2.

238   f)   Full Bleed – A method of printing allowing the entire surface of the medium to be marked. *I.e.,* there is no
239   white (or, more accurately, media-colored) edge around the printed content. Within the context of this
240   Service, its meaning is restricted to include only image content and simple superimposed annotation. That
241   is, print content containing arbitrary text and objects other than images is not considered full bleed, even
242   though that print content might not have an edge.
243   NOTE: In general, media registration and skew tolerances imply that a printer will need to do some special
244   processing to achieve full-bleed output. One common technique is to scale the image up to a size slightly
245   larger than the medium, implying some of the edge pixels will be lost.

246   g)   Impression – The print content affixed to one surface of a sheet of print medium. When printing only
247   single-sided, there is one impression per physical page, regardless of whether the *n* in n-up is 1 or greater

248   than 1. When duplex printing (printing on both sides of the medium), there are two impressions per
249   physical page, regardless of whether the *n* in n-up is 1 or greater than 1.

250 h) Layout Job Attributes - job attributes that are inherent to the integrity of the print content and are not
251   overridden by supplying corresponding IN parameters when submitting the job (see section 2.9.3.1.1).
252   (*E.g.,* page orientation.)

253 i) N-up – A method of printing where, when *n* is greater than 1, multiple logical pages are reduced in size
254   and printed on a single medium surface. For example, a 4-up printout has 4 logical pages imprinted on one
255   side of a single page at approximately ¼ of their usual size.

256 j) Non-printable Area – As defined by the CSS3 Paged Media Module [CSS3_PM], the area around the edge
257   of the physical medium that the printer is not capable of marking. In this specification it identifies the area
258   around the edge of the physical medium where individual pixels cannot be reliably positioned. For
259   example, a Printer may print in this area when using special techniques such as full-bleed processing, but
260   be unable to reliably place text in this area.

261 k) PDL – the Page Description Language. Any of numerous mechanisms to define document content and
262   formatting. Examples include XHTML and CSS, PostScript, PCL, etc.

263 l) PDL Data Stream - the stream of data to be printed as represented in a specified document format.

264 m) Print Service (or Printer) - the UPnP entity that accepts actions from Control Point (clients), returns
265   responses, sends events, and generates printed output.

266 n) Production Job Attributes - job attributes that are not inherent to the integrity of the print content, and so
267   the Control Point MAY override the PDL Data Stream instructions, if any, by supplying corresponding IN
268   parameters when submitting the job (see section 2.9.3.1.1). (*E.g.,* number of copies.)

269 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
270   JobId and appears in the JobIdList, and on which the Control Point can perform any of the Job operations
271   defined in this document.

272 p) Untracked Job - a non-UPnP job that is not visible to a UPnP Control Point; i.e., it does not have a JobId
273   and does not appear in the JobIdList, and on which the Control Point cannot perform any of the Job
274   operations defined in this document.

275 2.2.3. Notation: Use of Quotation Marks

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

279 2.2.4. Notation: Use of Asterisks in Action Names

280 PrintEnhanced:1 defines three separate actions for creating a print job—CreateJob, CreateJobV2 and CreateURIJob.
281 Some job processing behaviors depend on which action created the job, but many behaviors are common to two or
282 all three Create actions. To avoid many name repetitions, we will use the following shorthand notations when
283 referring collectively to two or more Create actions:
284   Create* —all three actions
285   CreateJob* — CreateJob and CreateJobV2
286

287 ## 2.3. References

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

290 [DEVICE] - UPnP Device Architecture, version 1.0 and UPnP Vendor's Implementation Guide. Available at:
291   http://www.upnp.org/standardizeddcps/documents/upnpresource20040907.zip

292 [HTTP] - RFC 2616 "Hypertext Transfer Protocol -- HTTP/1.1", R. Fielding, J. Gettys,   J. Mogul, H. Frystyk, L.
293   Masinter, P. Leach, T. Berners-Lee. June 1999. (Format: TXT=422317, PS=5529857, PDF=550558 bytes)
294   (Obsoletes RFC2068) (Updated by RFC2817) (Status: DRAFT STANDARD) Available at: ftp://ftp.rfc-
295   editor.org/in-notes/rfc2616.txt

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

298 [PWG5101.1] *IEEE-ISTO 5101.1-2001 The Printer Working Group Standard for Media Standardized Names 26*
299   *February 2002. Available at:* ftp://ftp.pwg.org/pub/pwg/standards/pwg5101.1.pdf, .doc, .rtf

300 [PWG5101.2] *IEEE-ISTO 5101.2-2004 The Printer Working Group Standard for Repertoires Supported Element 1*
301   *February 2004. Available at: ftp://ftp.pwg.org/pub/pwg/candidates/cs-crrepsup10-20040201-5101.2.pdf*

302 [XHTML-PRINT] – XHTML-Print, W3C Candidate Recommendation, 20 January 2004. Available at:
303   http://www.w3.org/TR/2004/CR-xhtml-print-20040120

304 [CSSPP] – CSS Print Profile, W3C Candidate Recommendation, 25 January 2004. Available at:
305   http://www.w3.org/TR/2004/CR-css-print-20040225

306 [CSS3_PM] - CSS3 Paged Media Module, W3C Candidate Recommendation, 25 February 2004. Available at:
307   http://www.w3.org/TR/2004/CR-css3-page-20040225

308 [MULTIPLEXED] – RFC 3391 "The MIME Application/Vnd.pwg-multiplexed Content-type", R. Herriot.
309 December 2002. (Status: INFORMATIONAL) Available at: ftp://ftp.rfc-editor.org/in-notes/rfc3391.txt
310
311 [XML] – *Extensible Markup Language (XML) 1.0 (Second Edition)*, T. Bray, J.Paoli, C. M. Sperberg-McQueen, E
312 Maler, eds. W3C Recommendations, 6 October 2000.

313 [XPCSSGUIDE] – XHTML-PRINT/CSS-Print Profile Guidelines for PrintEnhanced:1, January 2005.
314   Available at:
315   http://www.upnp.org/standardizeddcps/documents/PrintEnhanced1_guidelines_v1_050504.pdf

316 ## 2.4. Derived Data Types

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

318 ### 2.4.1. Comma Separated Value (CSV) Lists

319 The UPnP PrintEnhanced:1 Service uses variables that represent lists, or one-dimensional arrays, of values.
320 Examples include the supported sets of document formats and media stock. The UPnP Device Architecture,
321 Version 1.0 [DEVICE], does not provide for either an array type or a list type, so a list type is defined here. Lists
322 MAY either be homogeneous (all values are the same type) or heterogeneous (values of different types are
323 allowed). The data type of a homogeneous list is *string (CSV x)*, where *x* is the type of the individual values. The
324 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
325 element values. If the number of elements in the heterogeneous list is too large to show each type individually, that

326 variable type is represented as *string (CSV heterogeneous)*, and the variable description includes additional
327 information as to the expected sequence of values appearing in the list and their corresponding types.

328   o A list is represented as a UPnP String type.

329   o Values within a list are separated by commas.

330   o Only three value types are used as CSV elements in this specification—string, integer and boolean.

331   o Integer values are represented in CSVs with the same syntax as the int data type specified in [DEVICE]
332   (i.e., optional leading sign, optional leading zeroes).

333   o Boolean values are represented in CSVs as either '0' for false or '1' for true (which is a subset of the
334   defined boolean data type values specified in [DEVICE]: '0', 'false', 'no', '1', 'true', 'yes'.

335   o String values are represented in CSVs with the same syntax as the string data types specified in [DEVICE]
336   (i.e., any Unicode string), with two exceptions that are represented using a backslash escape character:

337   o The comma (',') is represented as '\,'.

338   o The backslash ('\') is represented as '\\'.

339   o Any white space before, after, or interior to a string value is part of that string value. White space before,
340   after, or interior to any other data type is not allowed.

341 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 *string*, a department *string* and years-of-service *u2*.

342

343   2.4.2. XML Fragments as UPnP Arguments

344 When an XML fragment is used for a UPnP argument, it places restrictions on the XML string data type. It needs
345 to be represented as well-formed XML. An XML fragment used within SOAP actions, in adherence to the UPnP
346 V1.0 architecture [DEVICE], MUST be escaped by using the normal XML rules, [XML] Section 2.4 Character
347 Data and Markup, before embedding it in a SOAP request or response message. The service action GetMediaList
348 described in this document requires the out argument MediaList to be specified as an XML fragment as defined in
349 section 2.6.3.2.   The XML escaping rules are summarized from the [XML] reference mentioned above:

350   • The (<) character MUST be encoded as (<)

351   • The (>) character MUST be encoded as (>)

352   • The (&) character MUST be encoded as (&)

353   • The (") character MUST be encoded as (")

354   • The (') character MUST be encoded as (')

355

2.5. Naming Conventions

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

2.6. State Variables

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

2.6.1. The Printer's Supported and Default Values

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

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

2.6.2. The Distinguished Value

386 Some Print Service actions have IN parameters that will always override any corresponding value that might be
387 provided in the PDL data stream (see section 2.9.3.1.1). For those situations where the Control Point prefers to let
388 the PDL data stream value override the IN parameter, the PrintEnhanced:1 Service has added the Distinguished

389   Value 'device-setting' to the <allowedValueList> of the associated state variable.   In the case where the attribute is
390   absent in the PDL data stream and the IN parameter value is specified as the Distinguished Value 'device-setting',
391   the Service uses its <defaultValue> value for the IN parameter.   For example, see CreateJob action, section 2.8.2.
392   When the Control Point supplies the Distinguished Value for such an IN parameter, the Print Service MUST
393   process the action following the corresponding print instruction in the PDL Data Stream, if present. If absent, the
394   Print Service MUST process the action as if the Service's current <defaultValue> for that IN parameter value had
395   been supplied by the Control Point. In other words, the Service's then current <defaultValue> value has lower
396   precedence than the PDL Data Stream. All implementations MUST support all Distinguished Value parameters
397   defined herein. The two preceding requirements also mean that the Distinguished Value for a variable MUST be
398   included in the variable's allowed value set, even if the vendor is sub-setting the allowed value set. However, the
399   Distinguished Value itself MUST NOT be used for the actual value of the <defaultValue> element in the SCPD.
400   Note: the Distinguished Values defined herein for a variable/parameter are not otherwise valid values for the
401   variable/parameter.

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

410   The Distinguished Value for all string variables defined herein is the string '**device-setting**'. For any vendor
411   extensions, the Distinguished Value for all string variables MUST be '**device-setting**'. The Distinguished Value for
412   all integer variables defined herein is the value '**0**'. For any vendor extensions, the Distinguished Value for integer
413   variables SHOULD be '**0**' (or '**-1**' if '**0**' is otherwise a useful value).

2.6.3. PrintEnhanced:1 Service Variables

414

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

421   **Table 1: State Variables**

Variable Name	Req. or Opt.[1]	Data Type	Allowed Value	Default Value (Mandatory except where indicated below)	Eng. Units
Printer & Job Attributes (in alphabetical order)					
A_ARG_TYPE_CriticalAttrib List	*R*	*string (CSV string)*	*See section 2.6.3.1*	*N/A*	*N/A*
A_ARG_TYPE_MediaList	*R*	*string (XML fragment)*	*See section 2.6.3.2*	*N/A*	*N/A*

Variable Name	Req. or Opt.[1]	Data Type	Allowed Value	Default Value (Mandatory except where indicated below)	Eng. Units
A_ARG_TYPE_PrinterAbort Reason	*R*	*string*	*See section 2.6.3.3*	*N/A*	*N/A*
CharRepSupported	*R*	*string*	*See section 2.6.3.4*	*<implementation specific>*	*N/A*
ColorSupported	*R*	*boolean*	*See section 2.6.3.5*	*<implementation specific>*	*N/A*
ContentCompleteList	*R*	*string (CSV i4)*	*See section 2.6.3.6*	*<empty string>*	*N/A*
Copies	*R*	*i4*	*Range: 0 to 2^{31}-1*	*<implementation specific> RECOMMENDED value: 1*	*N/A*
CriticalAttributesSupported	*R*	*String (CSV string)*	*See section 2.6.3.8*	*<implementation specific>*	*N/A*
DataSink	*R*	*uri*	*See Section 2.6.3.9*	*<empty string>*	*N/A*
DeviceId	*R*	*string – MUST be limited to 512 bytes.*		*<implementation specific>*	*N/A*
DocumentFormat	*R*	*string*	*See section 2.6.3.11*	*<implementation specific> RECOMMENDED value: application/xhtml-print-e*	*N/A*
DocumentUTF16Supported	*R*	*string*	*See section 2.6.3.12*	*<implementation specific>*	*N/A*
FullBleedSupported	*R*	*boolean*	*See section 2.6.3.13*	*<implementation specific>*	
InternetConnectState	*R*	*string*	*See section 2.6.3.14*	*<implementation specific>*	*N/A*
JobAbortState	*R*	*string (CSV i4, string, string, i4, string, string)*	*See section 2.6.3.15*	*<empty string>*	*N/A*

Variable Name	Req. or Opt.[1]	Data Type	Allowed Value	Default Value (Mandatory except where indicated below)	Eng. Units
JobEndState	*R*	*string (CSV i4,string,string,i4,string)*	*See section 2.6.3.16*	*<empty String>*	*N/A*
JobId	*R*	*i4*	*Range: 0 to 2^{31}-1*	*0*	*N/A*
JobIdList	*R*	*string (CSV i4)*	*See section 2.6.3.18*	*<empty String>*	*N/A*
JobMediaSheetsCompleted	*R*	*i4*	*Range: -1 to 2^{31}-1*	*0 or -1*	*N/A*
JobName	*R*	*string*		*<empty string>*	*N/A*
JobOriginatingUserName	*R*	*string*		*<empty string>*	*N/A*
MediaSize	*R*	*string*	*See section 2.6.3.22*	*<implementation specific>*	*N/A*
MediaType	*R*	*string*	*See section 2.6.3.23*	*<implementation specific> RECOMMENDED value: Stationery (if supported)*	*N/A*
NumberUp	*R*	*string*	*See section 2.6.3.24*	*<implementation specific> RECOMMENDED value: 1*	*N/A*
OrientationRequested	*R*	*string*	*See section 2.6.3.25*	*<implementation specific> RECOMMENDED value: portrait*	*N/A*
PageMargins	*R*	*String (CSV string, string, string, string)*	*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*	*string*		*<implementation specific>*	*N/A*
PrinterName	*R*	*string*		*<implementation specific>*	*N/A*
PrintQuality	*R*	*string*	*See section 2.6.3.29*	*<implementation specific>* *RECOMMENDED value: normal*	*N/A*
PrinterState	*R*	*string*	*See section 2.6.3.30*	*idle*	*N/A*
PrinterStateReasons	*R*	*string*	*See section 2.6.3.31*	*none*	*N/A*
Sides	*R*	*string*	*See section 2.6.3.32*	*<implementation specific>* *RECOMMENDED value: one-sided*	*N/A*
SourceURI	*R*	*uri*	*See Section 2.6.3.33*	*<empty string>*	
XHTMLImageSupported	*R*	*string*	*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*

422   [1] R = REQUIRED, O = Optional, X = Non-standard.

2.6.3.1. A_ARG_TYPE_CriticalAttribList

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

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

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

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

Table 2: Values for *CriticalAttributesList*

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

440

2.6.3.2. A_ARG_TYPE_MediaList

This variable is used as the related state variable for the OUT argument MediaList for the action GetMediaList.

Example 1: MediaType as a function of MediaSize

```
IN: MediaSize="om_small-photo_100x150mm"
IN: MediaType="none"
OUT:
      <MediaList>
            <MediaType MediaSize="om_small-photo_100x150mm">
                  photographic-glossy
                  photographic-matte
                  cardstock
            </MediaType>
      </MediaList>
```

Example 2: MediaSize as a function of MediaType

```
IN:  MediaSize="none"
IN:  MediaType="photographic-glossy"
OUT:
      <MediaList>
```

```
460                 <MediaSize MediaType="photographic-glossy">
461                       na_index-4x6_4x6in
462                       na_5x5_5x7in
463                       na-8x10
464                       na_letter_8.5x11in
465                 </MediaSize>
466           </MediaList>
467
```

468 *Example 3: All types for all sizes*

```
469   IN:   MediaSize="none"
470   IN:   MediaType="none"
471   OUT:
472         <MediaList>
473               <MediaType MediaSize="om_small-photo_100x150mm">
474                       photographic-glossy
475                       photographic-matte
476                       cardstock
477               </MediaType>
478               <MediaType MediaSize="jpn_hagaki_100x148mm">
479                       photographic-glossy
480                       photographic-matte
481                       cardstock
482               </MediaType>
483                       …
484         </MediaList>
485
```

486 *Example 4: MediaType as a function of the default value of MediaSize*

```
487   SCPD: <defaultValue>om_small-photo_100x150mm</defaultValue>
488   IN: MediaSize="device-setting"
489   IN: MediaType="none"
490   OUT:
491         <MediaList>
492               <MediaType MediaSize="om_small-photo_100x150mm">
493                       photographic-glossy
494                       photographic-matte
495                       cardstock
496               </MediaType>
497         </MediaList>
498
```

499 *Example 5: MediaSize as a function of the default value of MediaType*

```
500   SCPD: <defaultValue>photographic-glossy</defaultValue>
501   IN: MediaSize="none"
502   IN: MediaType="device-setting"
503   OUT:
504         <MediaList>
505               <MediaSize MediaType="photographic-glossy">
506                       na_index-4x6_4x6in
507                       na_5x5_5x7in
508                       na-8x10
509                       na_letter_8.5x11in
510               </MediaSize>
511         </MediaList>
512
513
514
```

515 The following examples illustrate the XML fragment before and after escape conversion by the Printer (refer to
516 section 2.4.2 for details on depicting XML fragments as UPnP arguments):

517 *Example 6: MediaSize as a function of the default value of MediaType (before escape conversion)*
518 SCPD: <defaultValue>photographic-glossy</defaultValue>
519 IN: MediaSize="*none*"
520 IN: MediaType="device-setting"
521 OUT:
522   <MediaList>
523   <MediaSize MediaType="photographic-glossy">
524   na_index-4x6_4x6in
525   na_5x5_5x7in
526   na-8x10
527   na_letter_8.5x11in
528   </MediaSize>
529   </MediaList>
530

531 *Example 7: MediaSize as a function of the default value of MediaType (after escape conversion)*
532 SCPD: <defaultValue>photographic-glossy</defaultValue>
533 IN: MediaSize="*none*"
534 IN: MediaType="device-setting"
535 OUT:
536   <MediaList>
537   &lt;MediaSize MediaType=&quot;photographic-glossy&quot;&gt;
538   na_index-4x6_4x6in
539   na_5x5_5x7in
540   na-8x10
541   na_letter_8.5x11in
542   &lt;/MediaSize&gt;
543   </MediaList>

544 ### 2.6.3.3. A_ARG_TYPE_PrinterAbortReason

545 *Used for one of the positional values of the evented state variable JobAbortState—see description in section*
546 *2.6.3.15.* Multiple conditions MAY exist. The vendor chooses the single value for the
547 A_ARG_TYPE_PrinterAbortReason variable to indicate the most important condition.

548 **Table 3: allowedValueList for *A_ARG_TYPE_PrinterAbortReason***

Value	Req. or Opt.
hardware-error	*O*
external-access-uri-not-found	*O*
external-access-object-failure	*O*
external-access-doc-format-err	*O*
external-access-http-error	*O*
vendor-defined	*O*

549

550 ### 2.6.3.4. CharRepSupported

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

554

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

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

559   ·   *"iana_iso_8859-1" commonly known as ISO 8859-1*

560   ·   *"iana_Shift_JIS" commonly known as Shift-JIS*

561   ·   *"unicode_katakana" from the Unicode Code Charts*

562   ·   *"vendor_lexmark_specials" a vendor specific character set*

563

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

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

567   **Table *4:* allowedValueList for *CharRepSupported***

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

568
569

570   ***2.6.3.5.* ColorSupported**

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

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

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

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

579   **Table *5:* allowedValueList for *ColorSupported***

Value	Req. or Opt.
0	*O*
1	*O*

580

581   **2.6.3.6. ContentCompleteList**

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

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

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

595   ***2.6.3.7. Copies***

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

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

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

602   Vendors MUST NOT extend the allowed values.

603   **2.6.3.8. CriticalAttributesSupported**

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

609   2.6.3.8.1.   Values With Corresponding IN Arguments

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

617   2.6.3.8.2.   Text-layout

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

620   Example 1:

621   *CriticalAttributesSupported contains text-layout*

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

623   *The PDL indicates that a page break should be avoided anywhere within a long span of text*
624   *which cannot be printed on one sheet of the Printer's default media size. The Printer MUST abort*
625   *the job when it discovers this problem.*

626   Example 2:

627   *CriticalAttributesSupported contains text-layout*

628   *CreateJobV2 IN: CriticalAttributesList contains text-layout*

629   *The PDL indicates that text should be placed 0.1 mm to the right of the left edge of the page. The*
630   *Printer cannot reliably position text at that location, so it MUST abort the job when it discovers*
631   *this problem*

632   2.6.3.8.3.   Image-layout

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

635   Example 1:

636   *CriticalAttributesSupported contains image-layout*

637   *CreateJobV2 IN: CriticalAttributesList contains image-layout*

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

641   Example 2:

642   *CriticalAttributesSupported contains image-layout*

643   *CreateJobV2 IN: CriticalAttributesList contains image-layout*

644   *The PDL indicates that10 images should be placed side-by-side across the page. When the*
645   *Printer retrieves the source information, it discovers that it cannot buffer sufficient image data to*
646   *compose the required output. The Printer MUST abort the job when it discovers this problem.*

647   2.6.3.8.4.   Image-orientation

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

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

653   2.6.3.8.5.   Pdl-fidelity

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

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

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

662   Example 1:

663   *CriticalAttributesSupported contains pdl-fidelity.*

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

665   *The PDL indicates that an image should be printed so that it covers the surface of the medium*
666   *except for a 1 mm margin around the edge. The Printer is not capable of reliably printing images*
667   *with such a narrow margin, so the Printer MUST abort the job when it discovers this problem.*
668   **Note***:* In this example, image-layout is compromised. Since image-layout is a function of the
669   PDL, pdl-fidelity is also compromised.

670   Example 2:

671   *CriticalAttributesSupported contains pdl-fidelity.*

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

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

675   Example 3:

676   *CriticalAttributesSupported contains pdl-fidelity.*

677   *CreateURIJob IN: CriticalAttributesList contains pdl-fidelity.*

678   *The PDL indicates that the job should be printed with content imposed on both sides of the media.*
679   *The IN argument indicates that the job should be printed 'one-sided'. The Printer is unable to*
680   *meet the two-sided request in the PDL data stream, but MUST NOT abort the job, because it* is
681   *able to satisfy the requirement to override that request with the IN Production argument request*
682   *for single-sided output.*

683   ## 2.6.3.8.6.   Font-family

684

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

688   ## 2.6.3.8.7.   Font-size

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

691

692   **Table 6: allowedValueList for *CriticalAttributesSupported***

Value	Req. or Opt.
none	*O*
The value for CriticalAttributesSupported MUST be "none" or an allowedValueList of the following values:	
copies	*O*
sides	*O*
number-up	*O*
orientation-requested	*O*
media-size	*O*
media-type	*O*
print-quality	*O*
text-layout	*O*
image-layout	*O*
image-orientation	*O*
pdl-fidelity	*O*
font-family	*O*
font-size	*O*
vendor-defined	*O*

693   ### *2.6.3.9.* DataSink

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

697   ### *2.6.3.10.*   **DeviceId**

698   *The value of this variable MUST exactly match the IEEE 1284-2000 Device ID string, except the length field MUST*
699   *NOT be specified. The supported value for DeviceId is discoverable in the <defaultValue> value via the SCPD.*
700   *The length of DeviceId, defined as a string, is limited to 512 bytes.*

701   *The IEEE 1284-2000 Device ID consists of a length field followed by a case-sensitive string of ASCII characters*
702   *defining peripheral characteristics and/or capabilities. For the purposes of this specification, the length bytes*
703   *MUST NOT be included. The Device ID sequence is composed of a series of keys and values of the form:*
704   `key: value {,value} repeated for each key`

705 *As indicated, each key MUST have one value, and MAY have more than one value. The minimum necessary keys*
706 *(case-sensitive) are MANUFACTURER, COMMAND SET, and MODEL. (These keys MAY be abbreviated as MFG,*
707 *CMD, and MDL respectively.) Each implementation MUST supply these three keys and possibly additional ones as*
708 *well. Each key (and each value) is a string of characters. Any characters except colon (:), comma (,), and semi-*
709 *colon (;) MAY be included as part of the key (or value) string. Any leading or trailing white space (SPACE[x'20'],*
710 *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*
711 *is still counted as part of the overall length of the sequence).*

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

714

715 `MANUFACTURER:ACME Manufacturing;`

716 `COMMAND SET:PCL,PJL,PS,XHTML-Print;`

717 `MODEL:LaserBeam 9;`

718 `COMMENT:Anything you like;`

719 `ACTIVE COMMAND SET:PCL;`

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

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

724 ### *2.6.3.11.   DocumentFormat*

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

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

730 - *'application/vnd.pwg-xhtml-print' : This MIME media type is deprecated in favor of*
731   *'application/xhtml-print'. It SHOULD NOT be used by Control Points, and MUST be*
732   *supported by Printers.*

733 - *'application/xhtml-print': This MIME type identifies the base level of XHTML-Print/CSSPP*
734   *support.*

735 - *'application/xhtml-print-e': This MIME type identifies documents conforming to the Enhanced*
736   *Layout profile of XHTML-Print/CSSPP.*

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

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

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

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

747 *(See [MODEL] section 4.1.9)*

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

750 **Table *7:* allowedValueList for *DocumentFormat***

Value	Req. or Opt.
unknown	*R*
application/vnd.pwg-xhtml-print *[deprecated in favor of application/xhtml-print]* *See NOTE below.*	*R*
application/xhtml-print	*R*
application/vnd.pwg-xhtml-print+xml *[deprecated in favor of application/xhtml-print]* *See NOTE below.*	*O*
application/xhtml-print-e	*R*
text/plain	*O*
text/plain;charset=utf-8	*O*
application/octet-stream	*O*
application/postscript	*O*
application/vnd.hp-PCL	*O*
<Registered MIME media types for other document formats> See NOTE below.	*O*
Vendor-defined *See NOTE below.*	*O*

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

754 *2.6.3.12.* **DocumentUTF16Supported**

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

757 *Vendors MUST support one of the following: "none" OR "all" OR specified allowed values of DocumentFormat.*
758 *The Printer MUST NOT combine "none" with any other values. The Printer MUST NOT combine "all" with any*
759 *other values.*

760 **Table *8:* allowedValueList for *DocumentUTF16Supported***

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

761 **2.6.3.13.   FullBleedSupported**

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

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

766 *Vendors MUST NOT extend the allowedValueList.*

767

768 **Table 9: allowedValueList for *FullBleedSupported***

Value	Req. or Opt.
0	*O*
1	*O*

769

770   **2.6.3.14.   InternetConnectState**

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

773   *unknown — it is not known whether the Printer has a connection to the Internet.*

774   *connected — the Printer has access to the Internet.*

775   *not-connected — the Printer does not have access to the Internet.*

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

781   *The Control Point SHOULD NOT proceed with creating a job which requires such connectivity if the Printer*
782   *reports that it is 'not-connected'. The Control Point SHOULD proceed with creating the job if the*
783   *InternetConnectState is 'connected' or 'unknown'.*

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

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

787

788   **Table 10: allowedValueList for *InternetConnectState***

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

789

790   **2.6.3.15.   JobAbortState**

791   *This variable holds the "terminating" state of the job most recently aborted by the Printer. It is evented; it is*
792   *triggered when any job terminates by being aborted, instead of being canceled or ending successfully.*

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

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

796   *The first five are the same items, in the same order, as the state variable JobEndState (refer to 2.6.3.16).*
797   *Furthermore, the values of these five items will be the same as the values of JobEndState, for the corresponding*
798   *print JobId. In particular, note that the value of job-completion-state will always be '**aborted**'. The sixth value will*
799   *be from the combined allowedValueLists of CriticalAttributesSupported and A_ARG_TYPE_PrinterAbortReason.*
800   *Multiple conditions MAY exist. The vendor chooses the single value for the job-abort-reason variable to indicate*
801   *the most important condition.*

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

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

803

2.6.3.16.   JobEndState

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

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

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

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

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

    JobMediaSheetsCompleted*: If JobId was the "active" job, i.e., the first job in JobIdList, this is the final value of JobMediaSheetsCompleted for the job. Otherwise, this value is '0'. See section 2.6.3.19.*

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

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

822   *__successful:__ The job printed successfully all of the pages of the job and the sheets have been*
823   *stacked in the output bin.*

824   *__canceled:__ The job was canceled either by a CancelJob action or the equivalent in another*
825   *protocol, or by user intervention.*

826 *2.6.3.17.*   **JobId**

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

829 *(See [MODEL] section 4.3.2)*

830 *2.6.3.18.*   **JobIdList**

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

834 *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*
835 *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.*

836 *The first job in the list is either currently printing, attempting to print (but the Printer is stopped), or is the next job*
837 *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*
838 *printed last. The first JobId in the list is removed when the job completes or is aborted. The corresponding JobId*
839 *in the list is removed when a job is canceled (see sections 2.8.2 and 2.8.3).*

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

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

844 *2.6.3.19.*   **JobMediaSheetsCompleted**

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

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

857 *2.6.3.20.*   **JobName**

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

860 ### *2.6.3.21.   JobOriginatingUserName*

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

864 ### *2.6.3.22.   **MediaSize***

865 *Identifies the medium size name and dimensions that the Printer Service uses for all sheets of the job. Each value*
866 *MUST include the name of the size followed by the dimensions in inches or millimeters followed by the "in" or*
867 *"mm" suffix to indicate the units. Both the Inch and Millimeter dimension MAY include a non-zero decimal*
868 *fraction set off by a period (.). The name of the size consists of a class part and a name part separated by an*
869 *underscore (_). The class part MUST be "na", "asme", or "oe" for inch units and "iso", "jis", "jpn", "prc",*
870 *"roc", or "om" for metric units (see [PWG5101.1] for additional class names). The name part is set off by a*
871 *second underscore (_) and the dimensions are separated by the lower case letter x. The shorter dimension MUST*
872 *come first. See the Allowed Values for examples.*
873 *For sizes that do not have standard names, a Control Point or a Print Service can create a customized name using*
874 *the 'custom_xxx' class and name, where xxx indicates the custom name of the medium, followed by the dimensions*
875 *in inches or millimeters as for standard names. For example, a custom 3.5 by 5.0 inch medium that, say, represents*
876 *an index card, could be indicated by the string value:*

877   *custom_index-card_3.5x5in*

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

879 *If a Printer supports the Control Point supplying custom names that are not one of the values in the Printer's*
880 *<allowedValueList> element, the Printer's <allowedValueList> element MUST include both the*
881 *'custom_max_IIIxJJJmm' and 'custom_min_IIIxJJJmm' (and/or 'custom_max_IIIxJJJin' and*
882 *'custom_min_IIIxJJJin') Allowed Values to indicate the minimum and maximum custom sizes that the Printer will*
883 *allow the Control Point to supply.*

884 *(See [PWG5101.1] for suggested media size names and their dimensions. These names SHOULD NOT use the*
885 *"custom" class name.)*

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

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

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

893 **Table *12:* allowedValueList for *MediaSize***

Value[3]	Req. or Opt.
device-setting	*R*
none	*R*
om_small-photo_100x150mm	*Q*
na_letter_8.5x11in	*Q*
na_legal_8.5x14in	*Q*
iso_a4_210x297mm	*Q*
iso_c5_162x229mm	*Q*
iso_dl_110x220mm	*Q*
jis_b4_257x364mm	*Q*
custom_xxx_IIIxJJJmm	*Q*
custom_xxx_IIIxJJJin	*Q*
custom_min_IIIxJJJmm	*Q*
custom_max_IIIxJJJin	*Q*
< Other values defined for media size by [PWG5101.1] >	*Q*
Vendor-defined (see [PWG5101.1]	*Q*

894   [3] *These values represent examples and are not intended to be exhaustive (see [PWG5101.1].*

895 ### *2.6.3.23.*   **MediaType**

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

897   **stationery**   *Separately cut sheets of an opaque material*
898   **transparency**   *Separately cut sheets of a transparent material*
899   **envelope**   *Envelopes that can be used for conventional mailing purposes*
900   **labels**   *Label stock [For example, a sheet of peel-off labels].*
901   **photographic**   *Separately cut sheets of an opaque material to produce photographic quality images*
902   **cardstock**   *Separately cut sheets of an opaque material that is heavier and stiffer than stationery.*
903   **device-setting**   *Indicates that the Control Point wants the Printer to use its <defaultValue> value for*
904   *MediaType.*

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

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

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

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

913 **Table *13:* allowedValueList for *MediaType***

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

914 *2.6.3.24.* **NumberUp**

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

917 *1 - One page per side.*
918 *2 - Two pages per side.*
919 *4 - Four pages per side.*
920 **device-setting**

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

923 *(See [MODEL] section 4.2.9)*

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

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

928 **Table *14:* allowedValueList for *NumberUp***

Value	Req. or Opt.
device-setting	*R*
1	*R*
2	*O*
4	*O*
Vendor-defined	*O*

929

930 ***2.6.3.25.*** **OrientationRequested**

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

934 ***portrait***
935 ***landscape***
936 ***reverse-landscape***
937 ***reverse-portrait***
938 ***device-setting***

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

943

944 *(See [MODEL] section 4.2.10 which intends the "orientation-requested" attribute to apply to 'text' MIME types.)*

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

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

949 *Vendors MUST NOT extend allowed values.*

950 **Table *15:* allowedValueList for *OrientationRequested***

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

951

952 **2.6.3.26.   PageMargins**

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

962 *Example: A na-letter medium that has a quarter of an inch margin on the Top, Right, and Left edges, and 0 on the*
963 *Bottom edge would be (no spaces): 0.25in,.250in,0in,.25in.*

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

966 *2.6.3.27.   PrinterLocation*

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

970 <u>(See [MODEL] section 4.4.4)</u>

971 *2.6.3.28.   PrinterName*

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

977 <u>*(See [MODEL] section 4.4.4)*</u>

978 *2.6.3.29.   PrintQuality*

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

980   <u>***draft***</u>
981   <u>***normal***</u>
982   <u>***high***</u>
983   <u>***device-setting***</u>

984   *(See [MODEL] section 4.2.13)*

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

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

989   *Vendors MUST NOT extend allowed values.*

990   **Table *16:* allowedValueList for *PrintQuality***

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

991

992   *2.6.3.30.*   **PrinterState**

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

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

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

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

998   *(See [MODEL] section 4.4.11)*

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

1000   **Table *17:* allowedValueList for *PrinterState***

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

1001

1002   *2.6.3.31.*   **PrinterStateReasons**

1003   *Indicates additional information about why the Printer is in its current state. Multiple conditions MAY exist. The*
1004   *vendor chooses the single value for PrinterStateReasons variable to indicate the most important condition.*

1005   *Note: Some of these reasons describe states of the Printer that cannot be entered on the basis of the currently*
1006   *defined UPnP actions set. For example, the Printer can be 'paused'; there is no PausePrinter action. The reason*
1007   *these states are presented is because some other protocol (or console action) can have caused the Printer to enter*
1008   *that state. Reason values:*

1009   ***none****- Indicates that there are no current state reasons*

1010   ***attention-required*** *- The device has stopped for a reason other than the PrinterStateReasons listed here and*
1011   *requires human intervention before it can continue.*

1012   ***media-jam*** *- The device has a media jam.*

1013   ***paused*** *- Someone has paused the Printer and the PrinterState is 'stopped'. In this state, a Printer will not produce*
1014   *printed output.*

1015   ***door-open*** *- One or more covers on the device are open.*

1016   ***media-low*** *- At least one input tray is low on media.*

1017   ***media-empty*** *- At least one input tray is empty.*

1018   ***output-area-almost-full*** *- One or more output areas is almost full (e.g., tray, stacker, collator).*

1019   ***output-area-full*** *- One or more output areas is full (e.g., tray, stacker, collator).*

1020   ***marker-supply-low****- The device is low on at least one marker supply (e.g., toner, ink, ribbon).*

1021   ***marker-supply-empty*** *- The device is out of at least one marker supply (e.g., toner, ink, ribbon).*

1022   ***marker-failure*** *- The device has at least one marking device which has failed and requires service or replacement.*

1023   ***media-change-request*** *- A job has been submitted that is requesting media that is currently not loaded. The job*
1024   *has specified a particular MediaSize and MediaType parameter value combination that is not loaded, although the*
1025   *Printer supports that combination. The user is prompted to load the appropriate media. The Printer is paused until*
1026   *the user has responded to the prompt.*

1027   *(See [MODEL] section 4.4.12. The IPP severity suffix MUST NOT be included and, unlike IPP, only one value*
1028   *MUST occur at a time.)*

1029   *Vendors MUST support the values that represent conditions that are detectable in their implementation. Therefore,*
1030   *vendors MAY subset allowed values if specific PrinterStateReasons are undetectable in their implementation.*

1031   *Vendors MAY extend allowed values. However, Printer vendors need to understand the implications of extending*
1032   *this list for a Control Point. The Control Point usually localizes the PrinterStateReasons value (as with other string*
1033   *variable values) to the human language of the user. However, such a Printer vendor extension value will not be*
1034   *recognized by the Control Point. As a fallback presentation, the Control Point MAY display the value received as*
1035   *is, which SHOULD be in English and therefore, might not be understandable by the user. Alternatively, the vendor*
1036   *might use the general PrinterStateReasons value: 'attention-required' and then explain the problem on the Printer*
1037   *console which the user would see when they are by the Printer.*

1038   **Table *18:* allowedValueList for *PrinterStateReasons***

Value	Req. or Opt. [3]
none	*R*
attention-required	*O*
media-jam	*O*
paused	*O*
door-open	*O*
media-low	*O*
media-empty	*O*
output-area-almost-full	*O*
output-area-full	*O*
marker-supply-low	*O*
marker-supply-empty	*O*
marker-failure	*O*
media-change-request	*O*
Vendor-defined	*O*

1039   *[3] Vendors MUST support the values that represent conditions that are detectable in their implementation.*

1040   ***2.6.3.32.***   **Sides**

1041   *Specifies how print content is to be imposed upon the two surfaces (sides) of the media for the job. Supported*
1042   *values are discoverable via the SCPD.*

1043

1044   *(See [MODEL] section 4.2.8.)*

1045   *The 'device-setting' Distinguished Value indicates that the Control Point wants the Printer to use its*
1046   *<defaultValue> value for Sides, but to allow that value to be overridden if a corresponding value is encountered in*
1047   *the PDL Data Stream.*

1048   *Vendors MAY subset allowed values, but MUST support the 'device-setting' Distinguished Value.*

1049   *Vendors MUST NOT extend allowed values.*

1050   **Table *19:* allowedValueList for *Sides***

Value	Req. or Opt.
device-setting	*R*
one-sided	*R*
two-sided-long-edge	*O*
two-sided-short-edge	*O*

1051

1052   *2.6.3.33.*   **SourceURI**

1053   *Contains the URI to which the device will send the HTTP GET operation (see section 2.8.11) to get the print*
1054   *document. This value is sent by the client (Control Point) in the CreateURIJob action request.*

1055   *2.6.3.34.*   **XHTMLImageSupported**

1056   *Identifies the Image formats supported by the Printer. Supported values are discoverable via the SCPD. Although*
1057   *the list of XHTMLImageSupported formats MAY be supported within other PDL contexts, there is no requirement*
1058   *incumbent on the Printer to do so. The image is sent as part of an XHTML-Print document* [XHTML-PRINT]*,*
1059   *either interleaved within XHTML-Print using the MIME Application/Multiplexed Content Type [MULTIPLEXED]*
1060   *or as a referenced object.*

1061   It is ***strongly*** recommended that images SHOULD be referenced as URI's within the XHTML-Print file and not
1062   interleaved via the Application/Multiplexed Content Type. This allows the Printer to pull swaths of the images as
1063   needed for page composition. The server hosting the image is likewise ***strongly*** recommended to support HTTP 1.1
1064   Partial Gets, enabling the Printer to pull the specific portions of the images as they are needed. The Printer MAY
1065   retrieve pieces of a single image multiple times to facilitate rotation and other special processing. This approach is
1066   key to achieving broad interoperability across a wide range of product capabilities, as it enables even very low-cost
1067   printers to successfully print a collection of images on a single page.

1068   *A printer device vendor MAY choose to support other XHTMLImageSupported formats: however, there is no*
1069   *requirement to support the MIME Application/Multiplexed Content Type [MULTIPLEXED] for these other image*
1070   *formats.*

1071   *All UPnP printers MUST support at least the 'image/jpeg' image format.*

1072   *Allowed values include all IANA-registered MIME media types for image formats. Vendors MAY extend the*
1073   *allowed values for this attribute.*

1074   *Note: 'image/jpeg' is registered as a MIME Media Type with IANA.*

1075   **Table *20:* allowedValueList for *XHTMLImageSupported***

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

1076
1077
1078

1079 ## 2.7.   Eventing and Moderation

1080 **Table 21: Event Moderation**

Variable Name	Evented	Moderated Event	Max Event Rate[1] (sec)	Logical Combination	Min Delta per Event[2]
PrinterState	*Yes*	*No*	*N/A*		*N/A*
PrinterStateReasons	*Yes*	*No*	*N/A*		*N/A*
JobIdList	*Yes*	*No*	*N/A*		*N/A*
JobEndState	*Yes*	*No*	*N/A*		*N/A*
JobMediaSheetsCompleted	*Yes*	*Yes*	*5*		*N/A*
ContentCompleteList	*Yes*	*No*	*N/A*		*N/A*
JobAbortState	*Yes*	*No*	*N/A*		*N/A*

1081   [1] Events containing this variable value SHOULD occur no more often than once every MaxEventRate seconds.

1082   [2] See 4.4, Eventing: Augmenting the UPnP Template Language in [DEVICE].

1083 ### 2.7.1. Event Model

1084 The eventing model for the print service has three main purposes.

1085   First is to inform the Control Point when there is a change in condition of the print device. Examples: the
1086   Printer becomes idle, a paper jam occurs or the Printer is low on paper. The PrinterState and
1087   PrinterStateReasons variables provide this information.

1088   Second is for job tracking. Events inform a Control Point when a job is submitted, when all data for the
1089   job has been received by the Printer, and when a job has completed or been removed from the job queue,
1090   and whether or not it completed successfully. The JobIdList, ContentCompleteList, JobEndState, and
1091   JobAbortState provide this information. JobEndState indicates the final status of each job. It lets Control
1092   Points know whether it completed successfully or was canceled or aborted.

1093   Third is to inform a Control Point of the progress of the current job. JobMediaSheetsCompleted is a
1094   moderated evented variable that updates an interested Control Point on the number of impressions printed
1095   for the current job.

2.7.2. Synchronization of Evented Variables

1096

1097 Table 4 below describes how internal printer state changes affect the values of the seven evented state variables,
1098 plus the non-evented variable, JobId. These state changes can be forced by any of: a Control Point invoking one of
1099 the print service actions documented herein, a non-UPnP external action or printer internal events and conditions.
1100 The effect of some non-UPnP external actions is indirect, i.e., they affect internal printer state immediately, but, if
1101 they result in any UPnP-visible effect, the affect appears later. All of these indirect effects have to do with
1102 management of Untracked Jobs. They are included in this table because their ultimate effect can be visible at some
1103 later time. A Control Point should be aware of this to fully understand observed behavior. For PrintEnhanced:1
1104 service implementers, the complete table is a guideline to the information that MUST be kept and how it is
1105 synchronized to guarantee that the externally visible state variables are always correct.

1106 In Table 4, column 1 contains the current value of the variable PrinterState. Column 2 lists the events that can
1107 trigger an internal printer state change. Column 3 gives the new printer state and the complete set of actions taken
1108 by the Printer on the transition that is triggered by the column 2 event. In several cases, the actions taken depend on
1109 other printer conditions in addition to the triggering event. Those situations are identified in the table by dividing
1110 the lower right portion of the corresponding event "cell" into multiple subcells, one for each condition or set of
1111 conditions that requires a different set of transition actions. The upper portion of the event cell is extended into
1112 column 3, signifying that no transition action(s) can be specified for this event except when the conditions in the
1113 event's subcells are also considered. The word *invisible* in column 3 means there is no state change that could be
1114 observed by a UPnP Control Point. All of the actions listed in column 3 MUST be completed atomically relative to
1115 all external UPnP observations.

1116 For the purposes of this document, atomically means:

1117 1. From the viewpoint of any Control Point observer external to the Print Service, all of the values change at
1118 the same time. To achieve this, all evented variables changed by this collected set of actions SHOULD
1119 appear in a single event message.

1120 2. It is not possible through any query action for a Control Point to detect that any single state variable has
1121 changed unless it detects that all have changed and been properly updated to their new values.

1122 To help understand the actions, let's follow one transition through the tables. Find the entry in column 2 "Terminate
1123 active job that was tracked". Its termination condition, *T*, is one of 'successful', 'canceled' or 'aborted'." Since it has subcell
1124 entries, there is no direct entry in column 3. Assume the normal situation of a busy Printer with more jobs queued
1125 and that all of them are tracked. The relevant added condition is "Next job is tracked." That takes us to column 3
1126 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
1127 new value of JobId to the new first element of JobIdList. M0 says to reset JobMediaSheetsCompleted to '0' if we track it, or
1128 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
1129 whether it was 'successful', 'canceled' or 'aborted'. Also note that the M2 value inside JobEndState is set according to the
1130 actual final value of the sheets printed, if known.

1131 NOTE: If the Printer implementation is unable to detect "content complete", then the ContentCompleteList event is returned at
1132 the same time as the JobEndState event.

1133

1134   **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
idle	Create Untracked Job — action invoked by non-UPnP entity	P
idle	\<printer error\>	S, R1
processing	CreateJob or CreateJobV2 or CreateURIJob or create non-UPnP Tracked Job	P, J2
processing	Create Untracked Job — action invoked by non-UPnP entity	*invisible*
processing	Terminate active job that was tracked, but for which all job data had not yet been received. Its termination condition, T, is one of 'successful' or 'canceled'.	
processing	No more jobs.	I, J0, M0, E1(T)
processing	Next job is tracked.	J3, M0, E1(T)
processing	Next job is untracked, and there are no more tracked jobs.	J0, M0, E1(T)
processing	Next job is untracked, but there are still Tracked Jobs in the queue.	J4, M0, E1(T)
processing	Terminate active job that was tracked, and for which all job data had been received. Its termination condition, T, is one of 'successful' or 'canceled'.	
processing	No more jobs.	I, J0, M0, E1(T), C2
processing	Next job is tracked.	J3, M0, E1(T), C2
processing	Next job is untracked, and there are no more Tracked Jobs.	J0, M0, E1(T), C2
processing	Next job is untracked, but there are still Tracked Jobs in the queue.	J4, M0, E1(T), C2
processing	Terminate Tracked Job that was not active, but for which all job data had not yet been received. Its termination condition, T, is one of 'canceled'.	J5, E2(T)
processing	Terminate Tracked Job that was not active, and for which all job data had been received. Its termination condition, T, is one of 'canceled'.	J5, E2(T), C2
processing	Terminate active job that was untracked.	
processing	No more jobs.	I, M0
processing	Next job is tracked.	J6, M0
processing	Next job is untracked.	*invisible*
processing	Terminate inactive job that was untracked.	*invisible*
processing	Abort active job that was tracked, but for which all job data had not yet been received.	
processing	No more jobs.	I, J0, M0, E1(*'aborted'*), A1(R)
processing	Next job is tracked.	J3, M0, E1(*'aborted'*), A1(R)
processing	Next job is untracked, and there are no more Tracked Jobs.	J0, M0, E1(*'aborted'*), A1(R)

		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.	*invisible*
	Last byte of data needed to print a job is received.		
		Job is tracked.	C1
		Job is untracked.	*invisible*
	<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.		*invisible*

1135

1136   **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 *plus* 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 *except* 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 *except* 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 *except* the ith 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. *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.*
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. R 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. R 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 *plus* one new job added. All data for the job in the ith 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.

1137

1138 2.8. Actions

1139 Immediately following this table is detailed information about these actions, including short descriptions of the
1140 actions, the effects of the actions on state variables, and error codes defined by the actions.

1141 **Table 24: Actions**

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

1142 [1] R = REQUIRED, O = Optional, X = Non-standard.

1143 Note: the error codes are derived from IPP status codes as follows (see [MODEL] for the detailed definition of each
1144 error code):

1145   (Client Error minus 400_{16}) convert to decimal + 10 + 700
1146   (Server Error minus 400_{16}) convert to decimal + 60 + 700
1147

1148 Error codes are returned in the <SOAP:Fault> element. A vendor MAY subset or extend these error codes, first by
1149 supporting additional IPP error codes defined [MODEL] in the UPnP 700 range, and then by supporting private
1150 error codes in the UPnP 800 range, if no suitable IPP error code exists.

1151 ### *2.8.1. CancelJob*

1152 *This operation allows a client to cancel a print job from the time the job is created up to the time it is completed,*
1153 *canceled or aborted.*

1154 ### *2.8.1.1. Arguments*

1155 **Table 25: Arguments for *CancelJob***

Argument	Direction	relatedStateVariable
JobId	*IN*	*JobId*

1156

1157 ### *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	*ClientErrorNotFound*	*The Printer has not found a job matching the JobId parameter (including when the parameter was not in the range: 1 to 2^{31}-1).*
760	*ServerErrorInternalError*	*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.*
765	*ServerErrorTemporaryError*	*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.*

1158 ### *2.8.1.3. Effect on State*

1159 *The specified job with a JobId from 1 to 2^{31}-1 is removed from the **JobIdList**. If the job was the current job (i.e.,*
1160 *JobId specified the current job), then JobId is set according to the transition actions described in Section 2.7.2.*

1161 ### *2.8.2. CreateJob (deprecated)*

1162 *[This action is deprecated in favor of CreateJobV2. See section 2.2.2d)]*

1163 *This action is the first step in submitting a job to the Printer. The Printer returns a unique JobId to identify the job*
1164 *for this service. The Printer generates the JobId in an implementation-defined manner. The Printer MUST return*
1165 *values in the range 1 to 2^{31}-1; 0 and negative values are invalid. Furthermore, the Printer SHOULD NOT re-use*
1166 *values recently assigned, since Control Points could confuse such jobs with older jobs.*

1167 *The <allowedValueList> element of the Service Description indicates the values of the parameters that the Print*
1168 *Service instance (Printer) supports (see section 2.3). The Printer performs the following validation in the indicated*
1169 *order:*

1170   1. *If the DocumentFormat is not supported, the Printer MUST reject the request and return the*
1171   *ClientErrorDocumentFormatNotSupported (720) error code.*

1172   2. *If the client (Control Point) supplies input parameters that are unsupported or their values are*
1173   *unsupported (except DocumentFormat), the Printer (1) MUST accept the CreateJob request, (2) MUST*
1174   *ignore or substitute supported values, respectively, and (3) MUST print the job.*

1175   3. *If a client (Control Point) supplies a conflicting combination of MediaSize and MediaType (or any other*
1176   *set of IN parameters), the Printer MUST accept the CreateJob request, (2) MUST ignore or substitute the*
1177   *conflicting values, and (3) MUST print the job. Whether or not a Printer can detect combinations of*
1178   *different parameter values that are not supported, such as combinations of MediaType and MediaSize*
1179   *values that are not supported, is IMPLEMENTATION-SPECIFIC. If an implementation does detect*
1180   *combinations that are not supported, it substitutes values for one or more parameters to give a*
1181   *combination that is supported.*

1182 *The client (Control Point) MUST send print data to the print service via a separate HTTP Post operation to the*
1183 *DataSink URI (see section 2.8.10) returned by the Printer in the CreateJob action response.*

1184   ***2.8.2.1. Arguments***

1185   **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*

1186   Section 2.8.2.1 describes the CreateJob action IN/OUT argument's related state variables. The State Variable
1187   Table provides a description and data type as well as the allowed and default values.

1188   ***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	*ServerErrorInternalError*	*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.*
765	*ServerErrorTemporaryError*	*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.*

2.8.3. *CreateJobV2*

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

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

The Printer returns a unique JobId to identify the job for this service. The Printer generates the JobId in an implementation-defined manner. However, the Printer MUST return values in the range 1 to 2^{31}-1; 0 and negative values are invalid. Furthermore, the Printer SHOULD NOT re-use values recently assigned, since clients (Control Points) could confuse such jobs with older jobs.

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

1. *If the DocumentFormat is not supported, the Printer MUST reject the request and return the ClientErrorDocumentFormatNotSupported (720) error code.*

2. *If the client (Control Point) supplies input parameters that are unsupported or their values are unsupported (except DocumentFormat) then:*

   a. *If the unsupported parameters are not included in the CriticalAttributesList, the Printer 1) MUST accept the CreateJobV2 request, 2) MUST ignore or substitute supported values, respectively, and 3) MUST print the job.*

   b. *If the unsupported parameters are included in the CriticalAttributesList, the Printer MUST reject the request and return the ClientErrorAttributesOrValuesNotSupported (721) error code (unlike the CreateJob action where the Printer MUST accept the request and process the job).*

3. *If a client (Control Point) supplies a conflicting combination of MediaSize and MediaType (or any other set of IN parameters), and at least one of the conflicting parameters represents a job attribute that is also included in the parameter CriticalAttributesList, the Printer MUST reject the action and return the ClientErrorConflictingAttributes (724) error code.*

4. *If a client (Control Point) combines "none" with any other value in CriticalAttributesList, the Printer MUST reject the action and return the ClientErrorConflictingAttributes (724) error code.*

5. *If a client (Control Point) supplies "none" in either MediaSize or MediaType then:*

   a. *If at least one of the parameters with the value "none" represents a job attribute that is included in the CriticalAttributesList, the printer MUST reject the action and return the ClientErrorConflictingAttributes (724) error code.*

   b. *If neither MediaSize nor MediaType is a job attribute in the CriticalAttributesList, the Printer 1) MUST accept the CreateJobV2 request, 2) MUST ignore or substitute supported values, respectively, and 3) MUST print the job.*

1225	6.	*If a client (Control Point) supplies a combination of MediaSize and MediaType IN parameter values that*
1226		*does not match the Printer's currently loaded media (see section 2.9.3.2) and the corresponding*
1227		*attribute(s) is/are included in CriticalAttributesList, the Printer MUST take one of the following actions:*
1228		a. *If the implementation does not support the 'media-change-request' PrinterStateReasons*
1229		*mechanism (see section 2.6.3.31), the Printer MUST reject the action and return the*
1230		*ClientErrorMediaNotLoaded (734) error code.*
1231		b. *If the implementation does support the 'media-change-request' PrinterStateReasons mechanism,*
1232		*the Printer MUST accept the request, but not print the job until the requested media is loaded.*

1233 *The client (Control Point) MUST send print data to the print service via a separate HTTP Post operation to the*
1234 *DataSink URI (see section 2.8.10).*

1235 *During job processing, if the Printer encounters a condition in the PDL Data Stream that it cannot honor (for*
1236 *example, media-type mismatch) and Pdl-fidelity is included in the CriticalAttributesList, the Printer MUST abort*
1237 *the job and supply the reason for the abort in the evented state variable JobAbortState. If the CriticalAttributesList*
1238 *includes none, the Printer completes processing as it would if the job had been created by Create Job.*

1239 ### *2.8.3.1. Arguments*

1240 All relatedStateVariables, except CriticalAttributesList, are the same as for the CreateJob action.

1241 **Table 27: Arguments for *CreateJobV2***

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*
CriticalAttributesList	*IN*	*A_ARG_TYPE_CriticalAttribList*
JobId	*OUT*	*JobId*
DataSink	*OUT*	*DataSink*

1242 ### *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	*ClientErrorDocumentFormatNotSupported*	*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*
721	*ClientErrorAttributesOrValuesNotSupported*	*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 parmeters are included in the CriticalAttributesList.*
724	*ClientErrorConflictingAttributes*	*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).*
760	*ServerErrorInternalError*	*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.*
765	*ServerErrorTemporaryError*	*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.*
800-899	TBD	Action-specific errors for non-standard actions. Defined by the UPnP vendor.

1243

1244   The Service State Table (Section 2.6.3) describes the CreateJobV2 action IN/OUT arguments related state variables.
1245   The Service State Table provides a description and data type as well as the allowed and default values.

2.8.4. CreateURIJob

1246

1247   *This action is very similar to CreateJobV2, but requires the Printer to* pull *the print data from a specified location*
1248   *rather than requiring the Control Point to* push *the print data to the Printer. It adds one IN argument to*
1249   *CreateJobV2 —SourceURI, which specifies the location of the document to be retrieved by the Printer. It removes*
1250   *the OUT argument DataSink from CreateJobV2, since no data will be POSTed by the Control Point.*

1251   *The Printer returns a unique JobId to identify the job for this service. The Printer generates the JobId in an*
1252   *implementation-defined manner. However, the Printer MUST return values in the range 1 to 2^{31}-1; 0 and negative*
1253   *values are invalid values to be returned as a result of a CreateURIJob action. Furthermore, the Printer SHOULD*
1254   *NOT re-use values recently assigned, since clients (Control Points) would confuse such jobs with older jobs.*

1255   *The <allowedValueList> element of the Service Description indicates the values of the parameters that the Print*
1256   *Service instance (Printer) supports (see section 3). The Printer performs the following validation in the indicated*
1257   *order:*

1258   1. *If the DocumentFormat is not supported, the Printer MUST reject the request and return the*
1259   *ClientErrorDocumentFormatNotSupported (720) error code.*

1260   2. *If the client (Control Point) supplies input parameters that are unsupported or their values are*
1261   *unsupported (except DocumentFormat) then:*

1262   a. *If the unsupported parameters are not included in the CriticalAttributesList, the Printer 1)*
1263   *MUST accept the CreateURIJob request, 2) MUST ignore or substitute supported values,*
1264   *respectively, and 3) MUST print the job.*

1265   b. *If the unsupported parameters are included in the CriticalAttributesList, the Printer MUST*
1266   *reject the request and return the ClientErrorAttributesOrValuesNotSupported (721) error code*
1267   *(unlike the CreateJob action where the Printer MUST accept the request and process the job).*

1268   3. *If a client (Control Point) supplies a conflicting combination of MediaSize and MediaType (or any other*
1269   *set of IN parameters), and at least one of the conflicting parameters represents a job attribute that is also*
1270   *included in the parameter CriticalAttributesList, the Printer MUST reject the action and return the*
1271   ClientErrorConflictingAttributes (724) error code.

1272   4. *If a client (Control Point) combines "none" with any other value in CriticalAttributesList, the Printer*
1273   *MUST reject the action and return the ClientErrorConflictingAttributes (724) error code.*

1274   5. *If a client (Control Point) supplies "none" in either MediaSize or MediaType then:*

1275   a. *If at least one of the parameters with the value "none" represents a job attribute that is*
1276   *included in the CriticalAttributesList, the printer MUST reject the action and return* the
1277   *ClientErrorConflictingAttributes (724) error code.*

1278   b. *If neither MediaSize nor MediaType is a job attribute in the CriticalAttributesList, the Printer*
1279   *1) MUST accept the CreateJobV2 request, 2) MUST ignore or substitute supported values,*
1280   *respectively, and 3) MUST print the job.*

1281   6. *If a client (Control Point) supplies a combination of MediaSize and MediaType IN parameter values that*
1282   *does not match the Printer's currently loaded media (see section 2.9.3.2) and the corresponding*
1283   *attribute(s) is/are included in CriticalAttributesList, the Printer MUST take one of the following actions:*

1284   a. *If the implementation does not support the 'media-change-request' PrinterStateReasons*
1285   *mechanism (see 2.6.3.31), the Printer MUST reject the action and return the*
1286   *ClientErrorMediaNotLoaded (734) error code.*

1287   b. *If the implementation does support the 'media-change-request' PrinterStateReasons mechanism,*
1288   *the Printer MUST accept the request, but not print the job until the requested media is loaded.*

1289 *The device MUST get the print data via a separate HTTP GET operation to the SourceURI (see section 2.8.11).*

1290 *During job processing, if the Printer encounters a condition in the PDL Data Stream that it cannot honor (for*
1291 *example, in-line side-by-side images exceed its buffer) and* image-layout *is included in the CriticalAttributesList, the*
1292 *Printer MUST abort the job and supply the reason for the abort in the evented state variable*
1293 *A_ARG_TYPE_PrinterAbortReason. Otherwise, if the CriticalAttributesList is none, it completes processing as it*
1294 *would if the job had been created by CreateJob.*

1295 ### 2.8.4.1. Arguments

1296 All relatedStateVariables, except SourceURI, are the same as for the CreateJobV2 action.

1297 **Table 28: Arguments for *CreateURIJob***

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*
CriticalAttributesList	*IN*	*A_ARG_TYPE_CriticalAttribList*
SourceURI	*IN*	*SourceURI*
JobId	*OUT*	*JobId*

1298 *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	*ClientErrorDocumentFormatNotSupported*	*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*
721	*ClientErrorAttributesOrValuesNotSupported*	*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 parmeters are included in the CriticalAttributesList.*
724	*ClientErrorConflictingAttributes*	*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).*

760	*ServerErrorInternalError*	*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.*
765	*ServerErrorTemporaryError*	*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.*
800-899	TBD	Action-specific errors for non-standard actions. Defined by the UPnP vendor.

1299

1300 The Service State Table (Section 2.6.3) describes the CreateURIJob action IN/OUT arguments related state
1301 variables. The Service State Table provides a description and data type as well as the allowed and default values.

1302 ### *2.8.5. GetJobAttributes*

1303 *The GetJobAttributes action allows a client (Control Point) to determine some of the values of job-related variables*
1304 *of the specified job with a JobId from 1 to 2^{31}-1. Only active and queued jobs can be queried since only these jobs*
1305 *are maintained in the JobIdList variable. These variables allow end users to identify their job (i.e., "JobName",*
1306 *"JobOriginatingUserName"). Other information can be derived from the GetJobAttributes action.*
1307 *If the specified job is found, its parameters are returned whether the job is active or queued. If the specified job is*
1308 *not found, the ClientErrorNotFound (716) is returned. Any job not found either never existed or has reached its*
1309 *terminating state (i.e., completed, cancelled, aborted) and is no longer known to the Print Service. If the value of*
1310 *JobMediaSheetsCompleted is greater than 0, the referenced job is active and the Printer has physically completed*
1311 *printing and stacking the number of media sheets indicated. If the value of JobMediaSheetsCompleted is 0 or –1,*
1312 *the client can determine whether the referenced job is active according to whether it is the first entry in JobIdList.*
1313 *The value of JobIdList can be retrieved either from its most recent evented value or from the action*
1314 *GetPrinterAttributes.*

1315 *2.8.5.1. Arguments*

1316 **Table 29: Arguments for *GetJobAttributes***

Argument	Direction	relatedStateVariable
JobId	*IN*	*JobId*
JobName	*OUT*	*JobName*
JobOriginatingUserName	*OUT*	*JobOriginatingUserName*
JobMediaSheetsCompleted	*OUT*	*JobMediaSheetsCompleted*

1317 *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	ClientErrorNotFound	The Printer has not found a job matching the JobId parameter (including when the parameter was not in the range: 1 to 2^{31}-1).
760	ServerErrorInternalError	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.
765	ServerErrorTemporaryError	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.

1318

2.8.6. GetMargins

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

- *The Non-Printable Area (see section 2.2.2j) for a specified MediaSize and MediaType combination.*

- *Valid combinations of MediaType and MediaSize that the Printer supports.*

- *Whether or not full-bleed printing is supported for the associated media size / type combination; i.e., whether or not the Printer is capable of printing one or more photos that cover the entire surface of one side of the medium sheet with no white edges (or more accurately, media-colored edges).*

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

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

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

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

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

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

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

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

1349   ***Case 1a***: The Printer returns *PageMargins* of 0mm,0mm,2mm,0mm and *FullBleedSupported=false.*
1350   The page margins indicate the Printer can reliably position print content right up to the top, right, and left
1351   edges of the medium, and up to 2 mm from the bottom edge of the medium. Since these values are not all
1352   zero, the CP MUST look at *FullBleedSupported,* and determines that this Printer is not capable of
1353   generating full-bleed output. The CP offers the user a choice of printing with a white border around the
1354   photo or canceling the print request.

1355   ***Case 1b***: The Printer returns *PageMargins* of 0mm,0mm,0mm,.2mm and *FullBleedSupported=true.*
1356   The page margins indicate the Printer can reliably position print content right up to the top, right, and left
1357   edges of the medium, and up to .2 mm from the bottom edge of the medium. Since these values are not all
1358   zero, the CP MUST look at *FullBleedSupported,* and determines that this Printer is capable of generating
1359   full-bleed output. The CP creates the full-bleed job and the Printer renders it successfully.

1360   ***Case 1c***: The Printer returns *PageMargins* of 0mm,0mm,0mm,0mm and *FullBleedSupported=false.*
1361   The page margins indicate the Printer can reliably position print content right up to the top, right, bottom,
1362   and left edges of the medium. Since these values are all zero, the CP knows that the Printer can reliably
1363   position arbitrary content anywhere on the medium surface; it need not look at *FullBleedSupported,* and
1364   determines that this Printer is capable of generating full-bleed output. The CP creates the full-bleed job
1365   and the Printer renders it successfully.

1366   ***Example 2***: The Control Point wishes to print a collection of images with text, and wants to use the maximum area
1367   of the medium surface that can be reliably utilized by the Printer. It sends a GetMargin action with
1368   MediaSize set to *device-setting* and MediaType set to *device-setting.*

1369   ***Case 2a***: The Printer's default MediaSize is *custom_photo_4x6in* and its default MediaType is *photographic*. The
1370   Printer returns PageMargins of 0mm,0mm,2mm,0mm and *FullBleedSupported=false*. The Control Point
1371   generates XHTML-Print content containing:
1372   <style type="text/css">
1373   @page { size: auto; margin: 0mm 0mm 2mm; }
1374   ...
1375   </style>
1376   The Control Point uses relative sizing and positioning to lay out the document. The Printer generates the
1377   appropriate output on 4x6 photo paper.

1378   ***Case 2b***: The Printer's default MediaSize is *iso_a4_210x297mm* and its default MediaType is *stationery*. The
1379   Printer returns PageMargins of 0in,.25in,.5in,.25in and *FullBleedSupported=true*. The Control Point
1380   generates XHTML-Print content containing:
1381   <style type="text/css">
1382   @page { size: auto; margin: 0in .25in .5in .25in; }
1383   ...
1384   </style>
1385   Otherwise, the Control Point sends the same content as generated for Case 2a above. The Printer generates
1386   the appropriate output on size A4 plain paper.

1387

1388   *2.8.6.1. Arguments*

1389   **Table 30: Arguments for *GetMargins***

Argument	Direction	relatedStateVariable
MediaSize	*IN*	*MediaSize*
MediaType	*IN*	*MediaType*
PageMargins	*OUT*	*PageMargins*
FullBleedSupported	*OUT*	*FullBleedSupported*

1390

1391 ***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	*ClientErrorAttributesOrValuesNot Supported*	*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 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).*
760	*ServerErrorInternalError*	*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.*
765	*ServerErrorTemporaryError*	*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.*
800-899	TBD	Action-specific errors for non-standard actions. Defined by the UPnP vendor.

1392

1393 ***2.8.6.3. Effect of Action on State***
1394 *This action does not affect the state in any way.*

2.8.7. GetMediaList

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

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

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

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

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

2.8.7.1. Arguments

Table 31: Arguments for *GetMediaList*

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

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	*ClientErrorConflictingAttribute s*	*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*

		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.
765	*ServerErrorTemporaryError*	*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.*
800-899	TBD	Action-specific errors for non-standard actions. Defined by the UPnP vendor.

1419

1420   *2.8.7.3. Effect of Action on State*

1421   *This action does not affect the state in any way.*

2.8.8. GetPrinterAttributes (deprecated)

1423   [This action is deprecated in favor of GetPrinterAttributesV2. See section 2.2.2d)]

1424   *The GetPrinterAttributes action allows a client (Control Point) to determine the state of the Printer and values of*
1425   *certain state variables that represent Printer attributes. In particular, the Control Point can determine the number*
1426   *of pending jobs.   The Control Point can also determine the state of the Print Service, and which job, if any, is the*
1427   *current job.*

1428   *The JobId OUT argument is the JobId of the current job; i.e., the job that has caused the PrinterState variable to be*
1429   *'processing' or 'stopped'. The JobId MUST be the first JobId in the JobIdList or 0. If there is no current job, i.e.,*
1430   *the PrinterState is 'idle' (there are no jobs, or all jobs are pending or held), then JobId contains a 0 which is an*
1431   *invalid JobId for a job). If JobId is 0, the Printer is either idle OR a non-UPnP job is printing (and the Printer*
1432   *implementation has chosen NOT to display non-UPnP jobs, i.e., the job is an Untracked Job).*

1433

1434   Note*: The GetPrinterAttributes action does not allow a client to discover the supported values of standard*
1435   *attributes. The client can discover what is supported from the <allowedValueList> element in the Service*
1436   *Description (see section 3). Neither does the GetPrinterAttributes action allow a client to discover vendor added*
1437   *attributes. Vendors MUST define their own private actions to return such additional attributes.*

1438   **2.8.8.1. Arguments**

1439   **Table 32: Arguments for *GetPrinterAttributes***

Argument	Direction	relatedStateVariable
PrinterState	*OUT*	*PrinterState*
PrinterStateReasons	*OUT*	*PrinterStateReasons*
JobIdList	*OUT*	*JobIdList*
JobId	*OUT*	*JobId*

1440

1441 *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	*ServerErrorInternalError*	*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.*
765	*ServerErrorTemporaryError*	*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.*

2.8.9. GetPrinterAttributesV2

1442

1443 *The GetPrinterAttributesV2 action allows a client (Control Point) to determine various aspects of the Printer's*
1444 *current state, including all information returned by GetPrinterAttributes plus an indication of whether or not the*
1445 *Printer currently has an active connection to the internet. When a Control Point invokes this action, the Printer*
1446 *SHOULD make an immediate attempt to determine the state of its Internet connection. Once the state is*
1447 *determined, the Printer sets the value of the state variable InternetConnectState and returns the newly determined*
1448 *value, along with the PrinterState, PrinterStateReasons, JobIdList, and JobId, as described for GetPrinterAttributes*
1449 *in section 2.8.8.*

1450 *Note: After this query, there is no guarantee how long the Printer's internet connection status will remain*
1451 *unchanged.*

1452 *2.8.9.1. Arguments*

1453 **Table 33: Arguments for *GetPrinterAttributesV2***

Argument	Direction	relatedStateVariable
PrinterState	*OUT*	*PrinterState*
PrinterStateReasons	*OUT*	*PrinterStateReasons*
JobIdList	*OUT*	*JobIdList*
JobId	*OUT*	*JobId*
InternetConnectState	*OUT*	*InternetConnectState*

1454

1455 *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	*ServerErrorInternalError*	*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.
765	*ServerErrorTemporaryError*	*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.*
800-899	TBD	Action-specific errors for non-standard actions. Defined by the UPnP vendor.

1456

1457 *2.8.9.3. Effect of Action on State*

1458 *This action does not affect the state in any way.*

2.8.10. HTTP POST

1459

1460 The client (Control Point) sends the print data using an HTTP [HTTP] Post operation (with chunking if desired), to
1461 the URI returned as the DataSink output parameter of the CreateJob* actions. Having received this DataSink URI
1462 in the Create Job* response, the client MUST then open a connection to the device using the URI and send the data.

1463 The client MUST open the data connection on the DataSink URI within 30 seconds after receiving the CreateJob*
1464 response. Otherwise, the Printer MUST time out, discard jobs for which no data has been received, and remove its
1465 JobId from the JobIdList variable. If no data at all is received for a job then the Printer SHOULD delete the job
1466 after a wait of at least 30 seconds and remove its JobId from the JobIdList variable. If data has been received for a
1467 job but a subsequent chunked HTTP POST operation does not arrive for an implementation-defined period of time
1468 (at least 30 seconds) then the data received so far is printed and the A_ARG_TYPE_PrinterAbortReason is set to
1469 *external-access-http-error*. If the Printer receives an HTTP Post for the DataSink URI after the timeout period, the
1470 Printer returns the HTTP 408 (Request Timeout) status code, if the job still exists, otherwise, the HTTP 404 (Not
1471 Found) status code.

1472 If the Printer accepts the CreateJob* action, but subsequently cannot accept the HTTP Post (because it is too busy or
1473 is accepting another job), the Printer MUST reject the HTTP Post and return the HTTP 503 (Service Unavailable).
1474 The Printer SHOULD reset the timer to 30 seconds or some other implementation-specific value and SHOULD
1475 return that value in the Retry-After HTTP header in the error response. The Printer SHOULD ensure that the Retry-
1476 After value is less than the maximum amount of time before which the device will timeout. If the Printer does not
1477 return Retry-After header, the HTTP spec [HTTP] says that the Control Point assumes an HTTP 500 error (internal
1478 server error) and no retry is allowed and the Printer aborts the job.

1479 An event will be sent to the client whenever the JobId is removed from the JobIdList.

1480 The URI MUST be a valid HTTP URI [HTTP]. The Printer MUST support HTTP/1.1 chunking [HTTP] for the
1481 Post operation. The client MUST send the DocumentFormat MIME Media Type value in the HTTP Content-Type
1482 header (or the 'unknown' special value, if the client doesn't know the actual document format - see section 2.6.3.11).
1483 If the DocumentFormat value does not match the HTTP Content-Type header value, the Printer MUST reject the
1484 request and return the HTTP 409 (Conflict) status code.

2.8.11. HTTP GET

1485

1486 The Printer retrieves print data using an HTTP [HTTP] GET operation (with Range headers for a partial GET, if
1487 desired) to the URI received as the SourceURI IN parameter of the CreateURIJob action or to a URI specified
1488 within the print content of a job. Having received this SourceURI in the CreateURIJob request or having processed

1489   a request for retrieval of information from a URI within the PDL, the Printer MUST then open a connection to the
1490   server indicated by the URI and request the data.

1491   The Printer MUST open the data connection on the SourceURI within 30 seconds after the job becomes the current
1492   job (*i.e.,* within 30 seconds of issuing the event notification which placed the target job at the top of the JobIdList).

1493   If no data at all is received for the job within 30 seconds of issuing the GET request, then the Printer SHOULD
1494   delete the job and remove its JobId from the JobIdList variable. The job is considered aborted, and JobAbortState is
1495   updated appropriately, triggering an event notification. If data has been received for a job but a subsequent HTTP
1496   GET response does not arrive for an implementation-defined period of time (at least 30 seconds) then the data
1497   received so far is printed and the job is aborted. If the Printer receives an HTTP GET response after the timeout
1498   period, the Printer SHOULD ignore the response and discard the data.

1499   An event will be sent to the client whenever the JobId is removed from the JobIdList.

1500   The URI MUST conform to RFC 2396 or RFC 2732.

1501   If a Content-Type header is not included in the GET response, the Printer SHOULD assume the content type
1502   matches the DocumentFormat provided in the CreateURIJob action. If no Content-Type header is provided and the
1503   CreateURIJob indicated the DocumentFormat is 'unknown', then if the Printer supports the *application/octet-*
1504   *stream* document format, it MUST process the data as such; else the Printer SHOULD abort the job and set the
1505   A_ARG_TYPE_PrinterAbortReason to *external-access-doc-format-err*. If a Content-Type header is returned
1506   which is in conflict with the DocumentFormat provided in the CreateURIJob action, a Printer which supports the
1507   *application/octet-stream* format SHOULD process the job as such; otherwise the Printer SHOULD assume the
1508   content type matches the Content-Type header. NOTE: This case is intended to cover the situation where the
1509   document is being retrieved from a non-UPnP Client (such as a web server) serving the content.

1510   If the print data indicates that multiple objects or images are to be composed on the page, and the Printer requires
1511   support for partial GETs to achieve such layout, and the HTTP server at the targeted URI does not implement partial
1512   GET capability, then:

1513   •   If reformatting the print output will not compromise any attribute included in the CriticalAttributesList,
1514   then the print output SHOULD be reformatted and the job completed normally.

1515   •   If reformatting the output would conflict with an attribute indicated as critical, the Printer MUST abort the
1516   job and, if supported, set the A_ARG_TYPE_PrinterAbortReason to *external-access-http-error*.

2.8.12. Error Codes

1518   The following table lists the possible error codes to actions for this service type. If an action results in multiple
1519   errors, the most specific error SHOULD be returned.

1520   **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	*ClientErrorNotFound*	*The Printer has not found a job matching the JobId parameter (including when the parameter was not in the range: 1 to 2^{31}-1).*

errorCode	errorDescription	Description
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.*
721	*ClientErrorAttributesOrValuesNotSupported*	*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 parmeters are included in the CriticalAttributesList.*
724	*ClientErrorConflictingAttributes*	*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).*
760	*ServerErrorInternalError*	*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.*
765	*ServerErrorTemporaryError*	*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.*
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.9. Theory of Operation

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

2.9.1. The Print Model

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

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

2.9.2. Jobs

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

1534   allocated by the device. The [MODEL] describes the rules for JobId production (1 to 2**31-1). The JobId is
1535   returned by the Create* actions.

1536   The set of jobs that a Printer has in its queue is exposed in a very simple way.

1537   o   The complete list of known jobs is made available as a state variable represented as CSV list (see section
1538   2.4.1) called **JobIdList**.

1539   o   All waiting jobs appear in the **JobIdList** variable-- even those that the device has decided not to print for
1540   some reason (they are in the IPP 'pending' or 'pending-held' job state).   All UPnP jobs are considered
1541   Tracked Jobs.

1542   o   The order of jobs in the **JobIdList** variable indicates the order in which the jobs will be initiated.

1543   o   The job that is actually printing at the moment (or for which the Print Service is paused) is called the
1544   current job. If the current job is "Tracked" (see 2.2.2o) its job identifier is stored in the **JobId** Print
1545   Service state variable and that same JobId value is also the first **JobId** in the **JobIdList**. If there is no
1546   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
1547   printing, the **JobId** is 0.

1548   o   Once a job has been printed (or cancelled or aborted) it no longer appears in the **JobIdList**, whether or not
1549   the Printer has any other jobs to print.

1550   o   When the Print Service has no Tracked Jobs to print, the **JobIdList** state variable is an empty string.

1551   ## 2.9.3. Job Processing

1552   ### *2.9.3.1. Intent of a Print Job*

1553   The intent of a Print job is indicated by the job attributes as represented by either:

1554   -   the IN parameters of the Create* action and/or

1555   -   the print instructions in the PDL Data stream.

1556   Many job attributes MAY be specified by either or both methods. This section defines the precedence between
1557   these two representations of the intent of a print job.

1558   #### 2.9.3.1.1.   Production vs. Layout Job Attributes
1559   This specification distinguishes two classes of such job attributes—*Production* and *Layout*. A Layout Job Attribute
1560   is one that is inherent to the print output and cannot be overridden by IN parameters when the job is created. A
1561   Production Job Attribute is one that can reasonably change at the different times when the job is printed without
1562   affecting important job characteristics. Obvious examples of Production Attributes are number of copies, number of
1563   sides and number of logical pages per physical sheet of paper, provided that when such Production Attributes are
1564   represented in the PDL Data Stream they are represented as print instructions. However, if number of copies or
1565   number of logical pages per physical sheet of paper is represented by repetitions of the PDL Data Stream, instead of
1566   a print instruction in the PDL Data Stream, such a representation is not considered a Production Job Attribute and so
1567   an IN parameter does not override such a representation.

1568   Job attributes are partitioned between Production and Layout as follows:

1569   Production Job Attributes (Job Attributes takes precedence):
1570   JobName
1571   JobOriginatingUserName
1572   Copies

1573	Sides
1574	NumberUp
1575	PrintQuality
1576	
1577	Layout Job Attributes (data stream takes precedence):
1578	OrientationRequested
1579	MediaSize
1580	MediaType

1581 ### 2.9.3.1.2. Precedence of Production vs. Layout Job Attributes

1582 The Control Point MUST supply an allowed value for each of the IN parameters defined for the Create* action.
1583 The PDL Data Stream MAY also have a value for any Production or Layout attribute represented as a print
1584 instruction. The Control Point MAY supply the Distinguished Value defined by this document for each IN
1585 parameter to request the Printer to use its <defaultValue> value (see section 2.2.2 e) and section 2.6.2) in case the
1586 corresponding print instruction in the PDL Data Stream is absent. The Printer SHOULD take the following action,
1587 depending on the values supplied by the Control Point in the Create* IN parameter and provided in the PDL Data
1588 Stream, for each given job attribute:

1589 **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 *higher* 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 *higher* than IN) **

1590 ** Only when both are supplied does the precedence depend on whether the attribute is a Production
1591 Attribute or a Layout Attribute. Production IN parameters take precedence, while Layout PDL print
1592 instructions take precedence.

1593 NOTE: Even for Layout Attributes, the IN parameter value supplied in the Create* action will be used as long as
1594 no overriding value is found in the PDL Data Stream itself.

1595 ### 2.9.3.2. Critical Attributes and the Intent of a Print Job

1596 PrintEnhanced:1 offers CPs two distinct approaches to satisfying print job intent. In both cases, the Printer is
1597 expected to honor input values for all print job attributes to the best of its ability. This includes both those attributes

1598   specified in the initiating Create* action and those found in the PDL as it is processed. Where the approaches differ
1599   is when the Printer encounters an attribute that it cannot satisfy. For jobs initiated by CreateJob, the single most
1600   important (implicit) job attribute is "content on paper". This does not mean that the Printer is allowed to ignore
1601   explicit attribute requests, but it does give the Printer significant freedom to select an alternative value when it
1602   cannot satisfy any particular attribute request. In contrast, for jobs initiated by CreateJobV2 or CreateURIJob, the
1603   Control Point tells the Printer exactly which attributes are critical for successful output. For those critical attributes,
1604   when the Printer detects that it cannot faithfully render the output according to the attribute value, it MUST abort
1605   the job immediately.

1606   The most important reason to use CreateJobV2 or CreateURIJob with a CriticalAttributesList specified as any
1607   value other than "none" is to avoid wasting expensive paper and ink (or other marking material) for printed output
1608   that the end user would consider unacceptable. The Printer MUST meet the following four requirements:

1609   1.   The Printer MUST inform the Control Point as to which Critical Attributes it supports (i.e., the set of job
1610   attributes whose settings the Printer is capable of detecting at print time and comparing to corresponding
1611   values requested by the submitting Control Point.) These are the Critical Attributes defined in Section
1612   2.2.2c). It is permitted that the set of critical attributes supported by the Printer is the NULL set (specified
1613   as the "none" value).

1614   2.   The Printer MUST be able to abort a print job when it cannot satisfy one or more of the Critical Attributes
1615   submitted in the print request.   The Printer SHOULD do the best job it can with respect to all other print
1616   job attributes not designated by the Control Point to be critical.

1617   3.   The Printer MUST allow Control Points to select which of those attributes, if any, it considers critical for
1618   any given print job. Processing details for Critical Attributes are described below with the state variable
1619   CriticalAttributesSupported and the actions CreateJobV2 and CreateURIJob.

1620   4.   If the Printer does abort a job due to Critical Attribute mismatch, the Printer MUST inform the Control
1621   Point of the type of the attribute (e.g., MediaSize, MediaType) whose mismatch resulted in the abort.

1622   For each job created using the action CreateJobV2 or CreateURIJob, the invoking Control Point gives the
1623   CriticalAttributesList as input. If the Printer discovers at any time that it cannot satisfy the requested value for an
1624   attribute in that input list, the Printer MUST abort that job.

1625   Critical Attributes are subject to the production versus layout precedence rules defined in section 2.9.3.1.2. That is,
1626   a critical attribute which is a production attribute such as *sides* is considered satisfied when the IN parameter for
1627   *sides* can be honored, whether or not there is a conflicting instruction in the PDL data stream. A critical attribute
1628   which is a layout attribute such as MediaSize is considered satisfied at job creation if the IN MediaSize requested
1629   can be honored; however, if the PDL data stream requests a different media size, the size requested by the PDL
1630   MUST be honored or the job aborted (assuming MediaSize is in the Printer's CriticalAttributesSupported values.)

1631   The Printer SHOULD only attempt to verify the value of a Critical Attribute at the time in printing when the
1632   attribute matters to physical output. This is the time when a mismatch between requested and actual values for the
1633   attribute would produce incorrect output if the job proceeds. At that time, before aborting the job, the Printer MAY
1634   use any means it deems appropriate to "correct" the Printer's inappropriate value, including asking for user
1635   intervention.

1636   Example 1:

1637   Printer A has an optical media type sensor that can determine whether it has plain paper, transparency, matte or
1638   glossy photo paper, etc., loaded. Printer B has no media type sensor, but it has a front panel selector that allows the
1639   user to "tell" the Printer the media type that is loaded. Printer C has no sensor and no front panel selector for media
1640   type. For printers A and B, media type *could be* considered a Critical Attribute. It is vendor choice whether *media-*
1641   *type* is included in the allowedValueList for CriticalAttributesSupported. For printer C, media type is not detectable
1642   and therefore cannot be included in the allowedValueList for CriticalAttributesSupported.

1643 Assume both printers A and B have included *media-type* in their respective CriticalAttributesSupported
1644 allowedValueLists, and the invoking Control Point also includes it in the CriticalAttributesList IN parameter. Upon
1645 receipt of a CreateJobV2 request with an attribute of media type set to glossy photo, printers A and B MUST verify
1646 that glossy photo is available for use (Printer A by sensing and Printer B via its front panel UI); if not available, they
1647 MUST either issue a *media-change-request* and wait for glossy photo to become available, or abort the job. Printer
1648 C SHOULD assume it doesn't know and proceed to print the job. If either printer A or B excludes *media-type* from
1649 CriticalAttributesSupported, or if it is included there but the invoking Control Point excludes *media-type* from
1650 CriticalAttributesList input to CreateJobV2 or CreateURIJob, then the printer behaves the same as printer C.

1651 Example 2:

1652 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
1653 the length of a manually fed sheet of paper. Printer E cannot sense media in the tray, but does detect the trailing
1654 edge of a sheet as the sheet moves toward the print head. Printer E can determine the length of a sheet of paper, but
1655 only after it has printed the contents of the whole physical page. For printer D, page length is detectable when
1656 loading from the tray, but not detectable when feeding from the manual slot. For printer E, page length is
1657 detectable, even though it cannot verify the value until *after* the page content is printed. When the Control Point
1658 includes *media-size* in its CriticalAttributesList for either printer, the printer MUST abort the job as soon as it
1659 detects a mismatch. For printer D, that would be at the time it prepares to load a sheet for printing. For printer E,
1660 that would be at the end of the first page. (While this example is useful to clarify the treatment of Critical
1661 Attributes, printer D would likely not support *media-size* as a CriticalAttribute, since it cannot detect *media-size* on
1662 manual feed.)

1663 The Printer is considered to satisfy the intent of a job when the value of every attribute included in the
1664 CriticalAttributesList matches the job's utilized value for that attribute. Any attributes in the Printer's
1665 allowedValueList for CriticalAttributesSupported but *not* included by the Control Point in the CriticalAttributesList
1666 submitted with the CreateJobV2 or CreateURIJob action are processed on a best-effort basis and MUST NOT cause
1667 the job to be aborted.

1668 This means that a request for plain paper can be considered met when photo paper is loaded if the Printer has no
1669 way of verifying its loaded media type. This allows manufacturers of lower cost printers to still take advantage of
1670 aborting jobs they know they can't meet intent for, without demanding that everything be detectable and included in
1671 CriticalAttributesSupported. For example, even if the Printer can't verify media type, it MAY still be able to verify
1672 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
1673 requested media size of 4" x 6" cannot be met, and therefore SHOULD be aborted.

1674 This implicit acceptance of non-detectable attributes still allows manufacturers to add value by detecting more
1675 attributes and exposing them in CriticalAttributesSupported.

1676 Finally, the Printer is NOT REQUIRED to know the value of Critical Attributes before marking paper. If the job
1677 requests US legal size paper, but the Printer doesn't know it has US letter size paper until it reaches the bottom of
1678 the letter-size sheet, that is acceptable. As soon as the Printer does discover that the sheet is short, though, it
1679 considers the intent unmet and MUST abort the job immediately if it was created by the CreateJobV2 or
1680 CreateURIJob action with *media-size* in the CriticalAttributesList.

1681 ### 2.9.4. Side-by-side Images

1682 Side-by-side images SHOULD be supported as specified in the XHTML-Print data without any reformatting. Side-
1683 by-side images MUST be supported when the images are "included by reference" (see XHTML-Print specification
1684 [XHTML-PRINT] section 4.4). If side-by-side images cannot be printed without reformatting when the job is
1685 created by CreateJobV2 or CreateURIJob, and *'image-layout'* is included in the CriticalAttributesList, the job
1686 MUST be aborted.

2.9.5. Actions

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

- o **Create Job (Deprecated)**. This action is used to submit a job to the Printer. The allocated JobId is returned.

- o **CancelJob**. This can be used to cancel a job using the JobId.

- o **GetPrinterAttributes (Deprecated)**. This action can be used to query some of the Printer attributes.

- o **GetJobAttributes**. This action can be used to query some of the job attributes of a specified job.

- o **CreateJobV2**. This action is used to submit a job to the Printer and the Printer MUST honor all supplied IN parameter values or reject the action. The allocated JobId is returned.

- o **CreateURIJob**. This action is the same as CreateJobV2, except that rather than pushing the print data to the Printer, the Control Point provides a SourceURI from which the Printer pulls the print data.

- o **GetMargins**. This action returns the four widths of the margins between the four edges and the edge of the printable area for the requested combination of MediaType and MediaSize, along with an indication as to whether the Printer supports full-bleed printing for the MediaSize / MediaType combination.

- o **GetMediaList**. This action returns the supported media sizes for a particular media type, or the supported media types for a particular media size, or a matrix of all types and sizes supported.

- o **GetPrinterAttributesV2**. Similar to GetPrinterAttributes, this action extends the set of printer attributes returned to include an indication of whether or not the Printer is currently connected to the internet.

2.9.6. Events

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

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

2.9.7. Security

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

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

2.9.8. Localization

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

1726 correspond to IPP keyword values) to the locale of its user. The Control Point (client) is expected to convert the
1727 enum integer values to human readable string values in the locale of the user.

1728 ## 2.9.9. IPP Data Type mapping to UPnP Data Types

1729 Basic IPP data types are transformed as follows.

1730 **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

1731 The derived types in IPP are mapped onto the following UPnP data types.

1732 **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).

1733

1734 **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.

1735

2.9.10. Improving Output Consistency for XHTML-Print

1737 Since the output of XHTML-Print [XHTML-PRINT] and CSS-Print [CSSPP] onto paged media (e.g., printed-
1738 paper) is different from the output display on screen media, pagination needs to be considered. This, along with the
1739 fact that some ambiguity exists in the interpretation of XHTML and CSS leads to various inconsistent outputs
1740 among output devices (e.g., printers), which is unexpected.

1741 In supporting XHTML-Print [XHTML-PRINT] and CSS-Print [CSSPP] as a document format, compliance to
1742 XHTML-PRINT/CSS Print Profile Guidelines for PrintEnhanced:1 [XPCSSGUIDE] is strongly recommended.
1743 This guideline provides information for both printers and content creators to help achieve an improved level of
1744 output consistency on print media among printers which support XHTML-Print and CSS-Print.

1745 In addition, in order to achieve a high level of consistency, sample templates for simple photo layouts is provided as
1746 a reference in [XPCSSGUIDE] for both printer implementations and content authors.

1747 3.   XML Service Description

1748 The following SCPD is intended as an example and vendors should adjust values based on their product specific
1749 implementation.   An exception to this is for Certification Tool testing where the SCPD provided below must match
1750 the AllowedValues listed below.

```
1751 <?xml version="1.0"?>
1752 <scpd xmlns="urn:schemas-upnp-org:service-1-0">
1753   <specVersion>
1754     <major>1</major>
1755     <minor>0</minor>
1756   </specVersion>
1757   <actionList>
1758     <action>
1759     <name>CancelJob</name>
1760       <argumentList>
1761         <argument>
1762           <name>JobId</name>
1763           <direction>in</direction>
1764           <relatedStateVariable>JobId</relatedStateVariable>
1765         </argument>
1766       </argumentList>
1767     </action>
1768     <action>
1769     <name>CreateJob</name>
1770       <argumentList>
1771         <argument>
1772           <name>JobName</name>
1773           <direction>in</direction>
1774           <relatedStateVariable>JobName</relatedStateVariable>
1775         </argument>
1776         <argument>
1777           <name>JobOriginatingUserName</name>
1778           <direction>in</direction>
1779           <relatedStateVariable>JobOriginatingUserName</relatedStateVariable>
1780         </argument>
1781         <argument>
1782           <name>DocumentFormat</name>
1783           <direction>in</direction>
1784           <relatedStateVariable>DocumentFormat</relatedStateVariable>
1785         </argument>
1786         <argument>
1787           <name>Copies</name>
1788           <direction>in</direction>
1789           <relatedStateVariable>Copies</relatedStateVariable>
1790         </argument>
1791         <argument>
1792           <name>Sides</name>
1793           <direction>in</direction>
1794           <relatedStateVariable>Sides</relatedStateVariable>
1795         </argument>
1796         <argument>
1797           <name>NumberUp</name>
1798           <direction>in</direction>
1799           <relatedStateVariable>NumberUp</relatedStateVariable>
```

```
1800            </argument>
1801            <argument>
1802              <name>OrientationRequested</name>
1803              <direction>in</direction>
1804              <relatedStateVariable>OrientationRequested</relatedStateVariable>
1805            </argument>
1806            <argument>
1807              <name>MediaSize</name>
1808              <direction>in</direction>
1809              <relatedStateVariable>MediaSize</relatedStateVariable>
1810            </argument>
1811            <argument>
1812              <name>MediaType</name>
1813              <direction>in</direction>
1814              <relatedStateVariable>MediaType</relatedStateVariable>
1815            </argument>
1816            <argument>
1817              <name>PrintQuality</name>
1818              <direction>in</direction>
1819              <relatedStateVariable>PrintQuality</relatedStateVariable>
1820            </argument>
1821            <argument>
1822              <name>JobId</name>
1823              <direction>out</direction>
1824              <relatedStateVariable>JobId</relatedStateVariable>
1825            </argument>
1826            <argument>
1827              <name>DataSink</name>
1828              <direction>out</direction>
1829              <relatedStateVariable>DataSink</relatedStateVariable>
1830            </argument>
1831          </argumentList>
1832        </action>
1833        <action>
1834        <name>CreateJobV2</name>
1835          <argumentList>
1836            <argument>
1837              <name>JobName</name>
1838              <direction>in</direction>
1839              <relatedStateVariable>JobName</relatedStateVariable>
1840            </argument>
1841            <argument>
1842              <name>JobOriginatingUserName</name>
1843              <direction>in</direction>
1844              <relatedStateVariable>JobOriginatingUserName</relatedStateVariable>
1845            </argument>
1846            <argument>
1847              <name>DocumentFormat</name>
1848              <direction>in</direction>
1849              <relatedStateVariable>DocumentFormat</relatedStateVariable>
1850            </argument>
1851            <argument>
1852              <name>Copies</name>
1853              <direction>in</direction>
1854              <relatedStateVariable>Copies</relatedStateVariable>
```

```
1855              </argument>
1856              <argument>
1857                <name>Sides</name>
1858                <direction>in</direction>
1859                <relatedStateVariable>Sides</relatedStateVariable>
1860              </argument>
1861              <argument>
1862                <name>NumberUp</name>
1863                <direction>in</direction>
1864                <relatedStateVariable>NumberUp</relatedStateVariable>
1865              </argument>
1866              <argument>
1867                <name>OrientationRequested</name>
1868                <direction>in</direction>
1869                <relatedStateVariable>OrientationRequested</relatedStateVariable>
1870              </argument>
1871              <argument>
1872                <name>MediaSize</name>
1873                <direction>in</direction>
1874                <relatedStateVariable>MediaSize</relatedStateVariable>
1875              </argument>
1876              <argument>
1877                <name>MediaType</name>
1878                <direction>in</direction>
1879                <relatedStateVariable>MediaType</relatedStateVariable>
1880              </argument>
1881              <argument>
1882                <name>PrintQuality</name>
1883                <direction>in</direction>
1884                <relatedStateVariable>PrintQuality</relatedStateVariable>
1885              </argument>
1886              <argument>
1887                <name>CriticalAttributesList</name>
1888                <direction>in</direction>
1889    <relatedStateVariable>A_ARG_TYPE_CriticalAttribList</relatedStateVariable>
1890              </argument>
1891              <argument>
1892                <name>JobId</name>
1893                <direction>out</direction>
1894                <relatedStateVariable>JobId</relatedStateVariable>
1895              </argument>
1896              <argument>
1897                <name>DataSink</name>
1898                <direction>out</direction>
1899                <relatedStateVariable>DataSink</relatedStateVariable>
1900              </argument>
1901            </argumentList>
1902          </action>
1903          <action>
1904          <name>CreateURIJob</name>
1905            <argumentList>
1906              <argument>
1907                <name>JobName</name>
1908                <direction>in</direction>
1909                <relatedStateVariable>JobName</relatedStateVariable>
```

```
1910          </argument>
1911          <argument>
1912            <name>JobOriginatingUserName</name>
1913            <direction>in</direction>
1914            <relatedStateVariable>JobOriginatingUserName</relatedStateVariable>
1915          </argument>
1916          <argument>
1917            <name>DocumentFormat</name>
1918            <direction>in</direction>
1919            <relatedStateVariable>DocumentFormat</relatedStateVariable>
1920          </argument>
1921          <argument>
1922            <name>Copies</name>
1923            <direction>in</direction>
1924            <relatedStateVariable>Copies</relatedStateVariable>
1925          </argument>
1926          <argument>
1927            <name>Sides</name>
1928            <direction>in</direction>
1929            <relatedStateVariable>Sides</relatedStateVariable>
1930          </argument>
1931          <argument>
1932            <name>NumberUp</name>
1933            <direction>in</direction>
1934            <relatedStateVariable>NumberUp</relatedStateVariable>
1935          </argument>
1936          <argument>
1937            <name>OrientationRequested</name>
1938            <direction>in</direction>
1939            <relatedStateVariable>OrientationRequested</relatedStateVariable>
1940          </argument>
1941          <argument>
1942            <name>MediaSize</name>
1943            <direction>in</direction>
1944            <relatedStateVariable>MediaSize</relatedStateVariable>
1945          </argument>
1946          <argument>
1947            <name>MediaType</name>
1948            <direction>in</direction>
1949            <relatedStateVariable>MediaType</relatedStateVariable>
1950          </argument>
1951          <argument>
1952            <name>PrintQuality</name>
1953            <direction>in</direction>
1954            <relatedStateVariable>PrintQuality</relatedStateVariable>
1955          </argument>
1956          <argument>
1957            <name>CriticalAttributesList</name>
1958            <direction>in</direction>
1959    <relatedStateVariable>A_ARG_TYPE_CriticalAttribList</relatedStateVariable>
1960          </argument>
1961            <argument>
1962            <name>SourceURI</name>
1963            <direction>in</direction>
1964            <relatedStateVariable>SourceURI</relatedStateVariable>
```

```
1965              </argument>
1966          <argument>
1967                <name>JobId</name>
1968                <direction>out</direction>
1969                <relatedStateVariable>JobId</relatedStateVariable>
1970              </argument>
1971            </argumentList>
1972          </action>
1973          <action>
1974          <name>GetJobAttributes</name>
1975            <argumentList>
1976              <argument>
1977                <name>JobId</name>
1978                <direction>in</direction>
1979                <relatedStateVariable>JobId</relatedStateVariable>
1980              </argument>
1981              <argument>
1982                <name>JobName</name>
1983                <direction>out</direction>
1984                <relatedStateVariable>JobName</relatedStateVariable>
1985              </argument>
1986              <argument>
1987                <name>JobOriginatingUserName</name>
1988                <direction>out</direction>
1989                <relatedStateVariable>JobOriginatingUserName</relatedStateVariable>
1990              </argument>
1991              <argument>
1992                <name>JobMediaSheetsCompleted</name>
1993                <direction>out</direction>
1994                <relatedStateVariable>JobMediaSheetsCompleted</relatedStateVariable>
1995              </argument>
1996            </argumentList>
1997          </action>
1998          <action>
1999          <name>GetMargins</name>
2000            <argumentList>
2001              <argument>
2002                <name>MediaSize</name>
2003                <direction>in</direction>
2004                <relatedStateVariable>MediaSize</relatedStateVariable>
2005              </argument>
2006              <argument>
2007                <name>MediaType</name>
2008                <direction>in</direction>
2009                <relatedStateVariable>MediaType</relatedStateVariable>
2010              </argument>
2011              <argument>
2012                <name>PageMargins</name>
2013                <direction>out</direction>
2014                <relatedStateVariable>PageMargins</relatedStateVariable>
2015              </argument>
2016              <argument>
2017                <name>FullBleedSupported</name>
2018                <direction>out</direction>
2019                <relatedStateVariable>FullBleedSupported</relatedStateVariable>
```

```
2020              </argument>
2021            </argumentList>
2022          </action>
2023          <action>
2024          <name>GetMediaList</name>
2025            <argumentList>
2026              <argument>
2027                <name>MediaSize</name>
2028                <direction>in</direction>
2029                <relatedStateVariable>MediaSize</relatedStateVariable>
2030              </argument>
2031              <argument>
2032                <name>MediaType</name>
2033                <direction>in</direction>
2034                <relatedStateVariable>MediaType</relatedStateVariable>
2035              </argument>
2036              <argument>
2037                <name>MediaList</name>
2038                <direction>out</direction>
2039                <relatedStateVariable>A_ARG_TYPE_MediaList</relatedStateVariable>
2040              </argument>
2041            </argumentList>
2042          </action>
2043          <action>
2044          <name>GetPrinterAttributes</name>
2045            <argumentList>
2046              <argument>
2047                <name>PrinterState</name>
2048                <direction>out</direction>
2049                <relatedStateVariable>PrinterState</relatedStateVariable>
2050              </argument>
2051              <argument>
2052                <name>PrinterStateReasons</name>
2053                <direction>out</direction>
2054                <relatedStateVariable>PrinterStateReasons</relatedStateVariable>
2055              </argument>
2056              <argument>
2057                <name>JobIdList</name>
2058                <direction>out</direction>
2059                <relatedStateVariable>JobIdList</relatedStateVariable>
2060              </argument>
2061              <argument>
2062                <name>JobId</name>
2063                <direction>out</direction>
2064                <relatedStateVariable>JobId</relatedStateVariable>
2065              </argument>
2066            </argumentList>
2067          </action>
2068          <action>
2069          <name>GetPrinterAttributesV2</name>
2070            <argumentList>
2071              <argument>
2072                <name>PrinterState</name>
2073                <direction>out</direction>
2074                <relatedStateVariable>PrinterState</relatedStateVariable>
```

```
2075              </argument>
2076              <argument>
2077                <name>PrinterStateReasons</name>
2078                <direction>out</direction>
2079                <relatedStateVariable>PrinterStateReasons</relatedStateVariable>
2080              </argument>
2081              <argument>
2082                <name>JobIdList</name>
2083                <direction>out</direction>
2084                <relatedStateVariable>JobIdList</relatedStateVariable>
2085              </argument>
2086              <argument>
2087                <name>JobId</name>
2088                <direction>out</direction>
2089                <relatedStateVariable>JobId</relatedStateVariable>
2090              </argument>
2091              <argument>
2092                <name>InternetConnectState</name>
2093                <direction>out</direction>
2094                <relatedStateVariable>InternetConnectState</relatedStateVariable>
2095              </argument>
2096          </argumentList>
2097        </action>
2098      </actionList>
2099      <serviceStateTable>
2100        <stateVariable sendEvents="no">
2101          <name>A_ARG_TYPE_CriticalAttribList</name>
2102          <dataType>string</dataType>
2103          <defaultValue></defaultValue>
2104        </stateVariable>
2105        <stateVariable sendEvents="no">
2106          <name>A_ARG_TYPE_MediaList</name>
2107          <dataType>string</dataType>
2108          <defaultValue></defaultValue>
2109        </stateVariable>
2110        <stateVariable sendEvents="no">
2111          <name>A_ARG_TYPE_PrinterAbortReason</name>
2112          <dataType>string</dataType>
2113          <defaultValue></defaultValue>
2114          <allowedValueList>
2115            <allowedValue>hardware-error</allowedValue>
2116            <allowedValue>external-access-uri-not-found</allowedValue>
2117            <allowedValue>external-access-object-failure</allowedValue>
2118            <allowedValue>external-access-doc-format-err</allowedValue>
2119            <allowedValue>external-access-http-error</allowedValue>
2120          </allowedValueList>
2121        </stateVariable>
2122        <stateVariable sendEvents="no">
2123          <name>CharRepSupported</name>
2124          <dataType>string</dataType>
2125          <defaultValue></defaultValue>
2126            <allowedValue>iana_iso_8859-1</allowedValue>
2127            <allowedValue>iana_Shift_JIS</allowedValue>
2128            <allowedValue>unicode_katakana</allowedValue>
2129        </stateVariable>
```

```
2130        <stateVariable sendEvents="no">
2131          <name>ColorSupported</name>
2132          <dataType>boolean</dataType>
2133          <defaultValue></defaultValue>
2134        </stateVariable>
2135         <stateVariable sendEvents="yes">
2136          <name>ContentCompleteList</name>
2137          <dataType>string</dataType>
2138          <defaultValue></defaultValue>
2139        </stateVariable>
2140        <stateVariable sendEvents="no">
2141          <name>Copies</name>
2142          <dataType>i4</dataType>
2143          <defaultValue>1</defaultValue>
2144          <allowedValueRange>
2145            <minimum>0</minimum>
2146            <maximum>2147483647</maximum>
2147            <step>1</step>
2148          </allowedValueRange>
2149        </stateVariable>
2150        <stateVariable sendEvents="no">
2151          <name>CriticalAttributesSupported</name>
2152          <dataType>string</dataType>
2153          <defaultValue></defaultValue>
2154          <allowedValueList>
2155            <allowedValue>none</allowedValue>
2156            <allowedValue>copies</allowedValue>
2157            <allowedValue>sides</allowedValue>
2158            <allowedValue>number-up</allowedValue>
2159            <allowedValue>orientation-requested</allowedValue>
2160            <allowedValue>media-size</allowedValue>
2161            <allowedValue>media-type</allowedValue>
2162            <allowedValue>print-quality</allowedValue>
2163            <allowedValue>text-layout</allowedValue>
2164            <allowedValue>image-layout</allowedValue>
2165            <allowedValue>image-orientation</allowedValue>
2166            <allowedValue>pdl-fidelity</allowedValue>
2167            <allowedValue>font-family</allowedValue>
2168            <allowedValue>font-size</allowedValue>
2169          </allowedValueList>
2170          </stateVariable>
2171        <stateVariable sendEvents="no">
2172          <name>DataSink</name>
2173          <dataType>uri</dataType>
2174          <defaultValue></defaultValue>
2175        </stateVariable>
2176        <stateVariable sendEvents="no">
2177          <name>DeviceId</name>
2178          <dataType>string</dataType>
2179          <defaultValue></defaultValue>
2180        </stateVariable>
2181        <stateVariable sendEvents="no">
2182          <name>DocumentFormat</name>
2183          <dataType>string</dataType>
2184          <defaultValue></defaultValue>
```

```
2185        <allowedValueList>
2186          <allowedValue>unknown</allowedValue>
2187          <allowedValue>application/xhtml-print</allowedValue>
2188          <allowedValue>application/xhtml-print-e</allowedValue>
2189          <allowedValue>application/octet-stream</allowedValue>
2190          <allowedValue>text/plain</allowedValue>
2191          <allowedValue>text/plain;charset=utf-8</allowedValue>
2192          <allowedValue>application/postscript</allowedValue>
2193          <allowedValue>application/vnd.hp-PCL</allowedValue>
2194        </allowedValueList>
2195      </stateVariable>
2196      <stateVariable sendEvents="no">
2197        <name>DocumentUTF16Supported</name>
2198        <dataType>string</dataType>
2199        <defaultValue></defaultValue>
2200        <allowedValueList>
2201          <allowedValue>none</allowedValue>
2202          <allowedValue>all</allowedValue>
2203          <allowedValue>application/xhtml-print</allowedValue>
2204          <allowedValue>application/xhtml-print-e</allowedValue>
2205          <allowedValue>application/octet-stream</allowedValue>
2206          <allowedValue>text/plain</allowedValue>
2207          <allowedValue>text/plain;charset=utf-8</allowedValue>
2208          <allowedValue>application/postscript</allowedValue>
2209          <allowedValue>application/vnd.hp-PCL</allowedValue>
2210        </allowedValueList>
2211      </stateVariable>
2212      <stateVariable sendEvents="no">
2213        <name>FullBleedSupported</name>
2214        <dataType>boolean</dataType>
2215        <defaultValue></defaultValue>
2216      </stateVariable>
2217      <stateVariable sendEvents="no">
2218        <name>InternetConnectState</name>
2219        <dataType>string</dataType>
2220        <defaultValue></defaultValue>
2221        <allowedValueList>
2222          <allowedValue>unknown</allowedValue>
2223          <allowedValue>connected</allowedValue>
2224          <allowedValue>not-connected</allowedValue>
2225        </allowedValueList>
2226      </stateVariable>
2227      <stateVariable sendEvents="yes">
2228        <name>JobAbortState</name>
2229        <dataType>string</dataType>
2230        <defaultValue></defaultValue>
2231      </stateVariable>
2232      <stateVariable sendEvents="yes">
2233        <name>JobEndState</name>
2234        <dataType>string</dataType>
2235        <defaultValue></defaultValue>
2236      </stateVariable>
2237      <stateVariable sendEvents="no">
2238        <name>JobId</name>
2239        <dataType>i4</dataType>
```

```
2240          <defaultValue>0</defaultValue>
2241          <allowedValueRange>
2242            <minimum>0</minimum>
2243            <maximum>2147483647</maximum>
2244            <step>1</step>
2245          </allowedValueRange>
2246        </stateVariable>
2247        <stateVariable sendEvents="yes">
2248          <name>JobIdList</name>
2249          <dataType>string</dataType>
2250          <defaultValue></defaultValue>
2251        </stateVariable>
2252        <stateVariable sendEvents="yes">
2253          <name>JobMediaSheetsCompleted</name>
2254          <dataType>i4</dataType>
2255          <defaultValue>0</defaultValue>
2256          <allowedValueRange>
2257            <minimum>-1</minimum>
2258            <maximum>2147483647</maximum>
2259            <step>1</step>
2260          </allowedValueRange>
2261        </stateVariable>
2262      <stateVariable sendEvents="no">
2263          <name>JobName</name>
2264          <dataType>string</dataType>
2265          <defaultValue></defaultValue>
2266        </stateVariable>
2267        <stateVariable sendEvents="no">
2268          <name>JobOriginatingUserName</name>
2269          <dataType>string</dataType>
2270          <defaultValue></defaultValue>
2271        </stateVariable>
2272        <stateVariable sendEvents="no">
2273          <name>MediaSize</name>
2274          <dataType>string</dataType>
2275          <defaultValue></defaultValue>
2276          <allowedValueList>
2277            <allowedValue>none</allowedValue>
2278            <allowedValue>om_small-photo_100x150mm</allowedValue>
2279            <allowedValue>na_letter_8.5x11in</allowedValue>
2280            <allowedValue>na_legal_8.5x14in</allowedValue>
2281            <allowedValue>iso_a4_210x297mm</allowedValue>
2282            <allowedValue>iso_c5_162x229mm</allowedValue>
2283            <allowedValue>iso_dl_110x220mm</allowedValue>
2284            <allowedValue>jis_b4_257x364mm</allowedValue>
2285            <allowedValue>device-setting</allowedValue>
2286          </allowedValueList>
2287        </stateVariable>
2288        <stateVariable sendEvents="no">
2289          <name>MediaType</name>
2290          <dataType>string</dataType>
2291          <defaultValue></defaultValue>
2292          <allowedValueList>
2293            <allowedValue>none</allowedValue>
2294            <allowedValue>stationery</allowedValue>
```

```
2295          <allowedValue>stationery-inkjet</allowedValue>
2296          <allowedValue>transparency</allowedValue>
2297          <allowedValue>envelope</allowedValue>
2298          <allowedValue>labels</allowedValue>
2299          <allowedValue>photographic</allowedValue>
2300          <allowedValue>photographic-glossy</allowedValue>
2301          <allowedValue>photographic-matte</allowedValue>
2302          <allowedValue>cardstock</allowedValue>
2303          <allowedValue>device-setting</allowedValue>
2304        </allowedValueList>
2305      </stateVariable>
2306      <stateVariable sendEvents="no">
2307        <name>NumberUp</name>
2308        <dataType>string</dataType>
2309        <defaultValue>1</defaultValue>
2310        <allowedValueList>
2311          <allowedValue>1</allowedValue>
2312          <allowedValue>2</allowedValue>
2313          <allowedValue>4</allowedValue>
2314          <allowedValue>device-setting</allowedValue>
2315        </allowedValueList>
2316      </stateVariable>
2317      <stateVariable sendEvents="no">
2318        <name>OrientationRequested</name>
2319        <dataType>string</dataType>
2320        <defaultValue>portrait</defaultValue>
2321        <allowedValueList>
2322          <allowedValue>portrait</allowedValue>
2323          <allowedValue>landscape</allowedValue>
2324          <allowedValue>reverse-landscape</allowedValue>
2325          <allowedValue>reverse-portrait</allowedValue>
2326          <allowedValue>device-setting</allowedValue>
2327        </allowedValueList>
2328      </stateVariable>
2329       <stateVariable sendEvents="no">
2330        <name>PageMargins</name>
2331        <dataType>string</dataType>
2332        <defaultValue></defaultValue>
2333      </stateVariable>
2334      <stateVariable sendEvents="no">
2335        <name>PrinterLocation</name>
2336        <dataType>string</dataType>
2337        <defaultValue></defaultValue>
2338      </stateVariable>
2339      <stateVariable sendEvents="no">
2340        <name>PrinterName</name>
2341        <dataType>string</dataType>
2342        <defaultValue></defaultValue>
2343      </stateVariable>
2344      <stateVariable sendEvents="no">
2345        <name>PrintQuality</name>
2346        <dataType>string</dataType>
2347        <defaultValue>normal</defaultValue>
2348        <allowedValueList>
2349          <allowedValue>draft</allowedValue>
```

```
2350          <allowedValue>normal</allowedValue>
2351          <allowedValue>high</allowedValue>
2352          <allowedValue>device-setting</allowedValue>
2353        </allowedValueList>
2354      </stateVariable>
2355      <stateVariable sendEvents="yes">
2356        <name>PrinterState</name>
2357        <dataType>string</dataType>
2358        <defaultValue>idle</defaultValue>
2359        <allowedValueList>
2360          <allowedValue>idle</allowedValue>
2361          <allowedValue>processing</allowedValue>
2362          <allowedValue>stopped</allowedValue>
2363        </allowedValueList>
2364      </stateVariable>
2365      <stateVariable sendEvents="yes">
2366        <name>PrinterStateReasons</name>
2367        <dataType>string</dataType>
2368        <defaultValue>none</defaultValue>
2369        <allowedValueList>
2370          <allowedValue>none</allowedValue>
2371          <allowedValue>attention-required</allowedValue>
2372          <allowedValue>media-jam</allowedValue>
2373          <allowedValue>paused</allowedValue>
2374          <allowedValue>door-open</allowedValue>
2375          <allowedValue>media-low</allowedValue>
2376          <allowedValue>media-empty</allowedValue>
2377          <allowedValue>output-area-almost-full</allowedValue>
2378          <allowedValue>output-area-full</allowedValue>
2379          <allowedValue>marker-supply-low</allowedValue>
2380          <allowedValue>marker-supply-empty</allowedValue>
2381          <allowedValue>marker-failure</allowedValue>
2382          <allowedValue>media-change-request</allowedValue>
2383        </allowedValueList>
2384      </stateVariable>
2385      <stateVariable sendEvents="no">
2386        <name>Sides</name>
2387        <dataType>string</dataType>
2388        <defaultValue>one-sided</defaultValue>
2389        <allowedValueList>
2390          <allowedValue>one-sided</allowedValue>
2391          <allowedValue>two-sided-long-edge</allowedValue>
2392          <allowedValue>two-sided-short-edge</allowedValue>
2393          <allowedValue>device-setting</allowedValue>
2394        </allowedValueList>
2395      </stateVariable>
2396      <stateVariable sendEvents="no">
2397        <name>SourceURI</name>
2398        <dataType>uri</dataType>
2399        <defaultValue></defaultValue>
2400      </stateVariable>
2401      <stateVariable sendEvents="no">
2402        <name>XHTMLImageSupported</name>
2403        <dataType>string</dataType>
2404        <defaultValue>image/jpeg</defaultValue>
```

```
2405            <allowedValueList>
2406              <allowedValue>image/jpeg</allowedValue>
2407            </allowedValueList>
2408          </stateVariable>
2409        </serviceStateTable>
2410    </scpd>
```