# ISO/IEC 29341-9-12

Edition 1.0 2008-11

# INTERNATIONAL STANDARD

## Information technology – UPnP Device Architecture – Part 9-12: Imaging Device Control Protocol – Print Basic Service

ISO/IEC 29341-9-12:2008(E)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

## About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

## About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

- Catalogue of IEC publications: www.iec.ch/searchpub

The IEC on-line Catalogue enables you to search by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub

Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Electropedia: www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv

If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00

# ISO/IEC 29341-9-12

Edition 1.0 2008-11

# INTERNATIONAL STANDARD

## Information technology – UPnP Device Architecture – Part 9-12: Imaging Device Control Protocol – Print Basic Service

INTERNATIONAL ELECTROTECHNICAL COMMISSION

PRICE CODE R

ICS 35.200

ISBN 2-8318-1011-31

# CONTENTS

## LIST OF TABLES



## LIST OF FIGURES

# INFORMATION TECHNOLOGY – UPNP DEVICE ARCHITECTURE –

## Part 9-12: Imaging Device Control Protocol – Print Basic Service

## FOREWORD

1) ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards. Their preparation is entrusted to technical committees; any ISO and IEC member body interested in the subject dealt with may participate in this preparatory work. International governmental and non-governmental organizations liaising with ISO and IEC also participate in this preparation.

2) In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

3) The formal decisions or agreements of IEC and ISO on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC and ISO member bodies.

4) IEC, ISO and ISO/IEC publications have the form of recommendations for international use and are accepted by IEC and ISO member bodies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC, ISO and ISO/IEC publications is accurate, IEC or ISO cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

5) In order to promote international uniformity, IEC and ISO member bodies undertake to apply IEC, ISO and ISO/IEC publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any ISO/IEC publication and the corresponding national or regional publication should be clearly indicated in the latter.

6) ISO and IEC provide no marking procedure to indicate their approval and cannot be rendered responsible for any equipment declared to be in conformity with an ISO/IEC publication.

7) All users should ensure that they have the latest edition of this publication.

8) No liability shall attach to IEC or ISO or its directors, employees, servants or agents including individual experts and members of their technical committees and IEC or ISO member bodies for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication of, use of, or reliance upon, this ISO/IEC publication or any other IEC, ISO or ISO/IEC publications.

9) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

IEC and ISO draw attention to the fact that it is claimed that compliance with this document may involve the use of patents as indicated below.

ISO and IEC take no position concerning the evidence, validity and scope of the putative patent rights. The holders of the putative patent rights have assured IEC and ISO that they are willing to negotiate free licences or licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of the putative patent rights are registered with IEC and ISO.

Intel Corporation has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Intel Corporation
Standards Licensing Department
5200 NE Elam Young Parkway
MS: JFS-98
USA – Hillsboro, Oregon 97124

Microsoft Corporation has informed IEC and ISO that it has patent applications or granted patents as listed below:

6101499 / US; 6687755 / US; 6910068 / US; 7130895 / US; 6725281 / US; 7089307 / US; 7069312 / US; 10/783 524 /US

Information may be obtained from:

Microsoft Corporation
One Microsoft Way
USA – Redmond WA 98052

Philips International B.V. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Philips International B.V. – IP&S
High Tech campus, building 44 3A21
NL – 5656 Eindhoven

NXP B.V. (NL) has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

NXP B.V. (NL)
High Tech campus 60
NL – 5656 AG Eindhoven

Matsushita Electric Industrial Co. Ltd. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Matsushita Electric Industrial Co. Ltd.
1-3-7 Shiromi, Chuoh-ku
JP – Osaka 540-6139

Hewlett Packard Company has informed IEC and ISO that it has patent applications or granted patents as listed below:

5 956 487 / US; 6 170 007 / US; 6 139 177 / US; 6 529 936 / US; 6 470 339 / US; 6 571 388 / US; 6 205 466 / US

Information may be obtained from:

Hewlett Packard Company
1501 Page Mill Road
USA – Palo Alto, CA 94304

Samsung Electronics Co. Ltd. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Digital Media Business, Samsung Electronics Co. Ltd.
416 Maetan-3 Dong, Yeongtang-Gu,
KR – Suwon City 443-742

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. IEC and ISO shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 29341-9-12 was prepared by UPnP Implementers Corporation and adopted, under the PAS procedure, by joint technical committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

The list of all currently available parts of the ISO/IEC 29341 series, under the general title *Universal plug and play (UPnP) architecture*, can be found on the IEC web site.

This International Standard has been approved by vote of the member bodies, and the voting results may be obtained from the address given on the second title page.

# ORIGINAL UPNP DOCUMENTS
(informative)

Reference may be made in this document to original UPnP documents. These references are retained in order to maintain consistency between the specifications as published by ISO/IEC and by UPnP Implementers Corporation. The following table indicates the original UPnP document titles and the corresponding part of ISO/IEC 29341:

| **UPnP Document Title** | **ISO/IEC 29341 Part** |
|---|---|
| UPnP Device Architecture 1.0 | ISO/IEC 29341-1 |
| UPnP Basic:1 Device | ISO/IEC 29341-2 |
| UPnP AV Architecture:1 | ISO/IEC 29341-3-1 |
| UPnP MediaRenderer:1 Device | ISO/IEC 29341-3-2 |
| UPnP MediaServer:1 Device | ISO/IEC 29341-3-3 |
| UPnP AVTransport:1 Service | ISO/IEC 29341-3-10 |
| UPnP ConnectionManager:1 Service | ISO/IEC 29341-3-11 |
| UPnP ContentDirectory:1 Service | ISO/IEC 29341-3-12 |
| UPnP RenderingControl:1 Service | ISO/IEC 29341-3-13 |
| UPnP MediaRenderer:2 Device | ISO/IEC 29341-4-2 |
| UPnP MediaServer:2 Device | ISO/IEC 29341-4-3 |
| UPnP AV Datastructure Template:1 | ISO/IEC 29341-4-4 |
| UPnP AVTransport:2 Service | ISO/IEC 29341-4-10 |
| UPnP ConnectionManager:2 Service | ISO/IEC 29341-4-11 |
| UPnP ContentDirectory:2 Service | ISO/IEC 29341-4-12 |
| UPnP RenderingControl:2 Service | ISO/IEC 29341-4-13 |
| UPnP ScheduledRecording:1 | ISO/IEC 29341-4-14 |
| UPnP DigitalSecurityCamera:1 Device | ISO/IEC 29341-5-1 |
| UPnP DigitalSecurityCameraMotionImage:1 Service | ISO/IEC 29341-5-10 |
| UPnP DigitalSecurityCameraSettings:1 Service | ISO/IEC 29341-5-11 |
| UPnP DigitalSecurityCameraStillImage:1 Service | ISO/IEC 29341-5-12 |
| UPnP HVAC_System:1 Device | ISO/IEC 29341-6-1 |
| UPnP HVAC_ZoneThermostat:1 Device | ISO/IEC 29341-6-2 |
| UPnP ControlValve:1 Service | ISO/IEC 29341-6-10 |
| UPnP HVAC_FanOperatingMode:1 Service | ISO/IEC 29341-6-11 |
| UPnP FanSpeed:1 Service | ISO/IEC 29341-6-12 |
| UPnP HouseStatus:1 Service | ISO/IEC 29341-6-13 |
| UPnP HVAC_SetpointSchedule:1 Service | ISO/IEC 29341-6-14 |
| UPnP TemperatureSensor:1 Service | ISO/IEC 29341-6-15 |
| UPnP TemperatureSetpoint:1 Service | ISO/IEC 29341-6-16 |
| UPnP HVAC_UserOperatingMode:1 Service | ISO/IEC 29341-6-17 |
| UPnP BinaryLight:1 Device | ISO/IEC 29341-7-1 |
| UPnP DimmableLight:1 Device | ISO/IEC 29341-7-2 |
| UPnP Dimming:1 Service | ISO/IEC 29341-7-10 |
| UPnP SwitchPower:1 Service | ISO/IEC 29341-7-11 |
| UPnP InternetGatewayDevice:1 Device | ISO/IEC 29341-8-1 |
| UPnP LANDevice:1 Device | ISO/IEC 29341-8-2 |
| UPnP WANDevice:1 Device | ISO/IEC 29341-8-3 |
| UPnP WANConnectionDevice:1 Device | ISO/IEC 29341-8-4 |
| UPnP WLANAccessPointDevice:1 Device | ISO/IEC 29341-8-5 |
| UPnP LANHostConfigManagement:1 Service | ISO/IEC 29341-8-10 |
| UPnP Layer3Forwarding:1 Service | ISO/IEC 29341-8-11 |
| UPnP LinkAuthentication:1 Service | ISO/IEC 29341-8-12 |
| UPnP RadiusClient:1 Service | ISO/IEC 29341-8-13 |
| UPnP WANCableLinkConfig:1 Service | ISO/IEC 29341-8-14 |
| UPnP WANCommonInterfaceConfig:1 Service | ISO/IEC 29341-8-15 |
| UPnP WANDSLLinkConfig:1 Service | ISO/IEC 29341-8-16 |
| UPnP WANEthernetLinkConfig:1 Service | ISO/IEC 29341-8-17 |
| UPnP WANIPConnection:1 Service | ISO/IEC 29341-8-18 |
| UPnP WANPOTSLinkConfig:1 Service | ISO/IEC 29341-8-19 |
| UPnP WANPPPConnection:1 Service | ISO/IEC 29341-8-20 |
| UPnP WLANConfiguration:1 Service | ISO/IEC 29341-8-21 |
| UPnP Printer:1 Device | ISO/IEC 29341-9-1 |
| UPnP Scanner:1.0 Device | ISO/IEC 29341-9-2 |
| UPnP ExternalActivity:1 Service | ISO/IEC 29341-9-10 |
| UPnP Feeder:1.0 Service | ISO/IEC 29341-9-11 |
| UPnP PrintBasic:1 Service | ISO/IEC 29341-9-12 |
| UPnP Scan:1 Service | ISO/IEC 29341-9-13 |
| UPnP QoS Architecture:1.0 | ISO/IEC 29341-10-1 |
| UPnP QosDevice:1 Service | ISO/IEC 29341-10-10 |
| UPnP QosManager:1 Service | ISO/IEC 29341-10-11 |
| UPnP QosPolicyHolder:1 Service | ISO/IEC 29341-10-12 |
| UPnP QoS Architecture:2 | ISO/IEC 29341-11-1 |
| UPnP QOS v2 Schema Files | ISO/IEC 29341-11-2 |

| UPnP Document Title | ISO/IEC 29341 Part |
|---|---|
| UPnP QosDevice:2 Service | ISO/IEC 29341-11-10 |
| UPnP QosManager:2 Service | ISO/IEC 29341-11-11 |
| UPnP QosPolicyHolder:2 Service | ISO/IEC 29341-11-12 |
| UPnP RemoteUIClientDevice:1 Device | ISO/IEC 29341-12-1 |
| UPnP RemoteUIServerDevice:1 Device | ISO/IEC 29341-12-2 |
| UPnP RemoteUIClient:1 Service | ISO/IEC 29341-12-10 |
| UPnP RemoteUIServer:1 Service | ISO/IEC 29341-12-11 |
| UPnP DeviceSecurity:1 Service | ISO/IEC 29341-13-10 |
| UPnP SecurityConsole:1 Service | ISO/IEC 29341-13-11 |

# 1. Overview and Scope

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

This service-type enables the following functions:

- *Printing*

This service template does not address:

- *Faxing*

# 2. Service Modeling Definitions

## 2.1. ServiceType

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

## 2.2. Terminology

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

### 2.2.1. Conformance Terminology

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

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

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

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

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

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

### 2.2.2. Other terminology

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

a) Print Service (or Printer) - the UPnP entity that accepts actions from UCP (clients), returns responses, and sends events.

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

c) Production Job Attributes - job attributes that are not inherent to the PDL Data Stream and so the UCP MAY override the PDL Data Stream instructions, if any, by supplying corresponding IN parameters when submitting the job (see section 2.4).

d) Layout Job Attributes - job attributes that are inherent to the PDL Data Stream and *cannot* be overridden by supplying corresponding IN parameters when submitting the job (see section 2.4).

e) Comma Separated Value (CSV) - a variable that contains multiple string values separated by the US-ASCII COMMA (‘,’) character (see section 2.5.1.1).

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

g) Tracked Job - a UPnP or non-UPnP job that is visible to a UPnP control point, i.e., has a JobId and appears in the JobIdList, and on which the control point can perform any of the Job operations defined in this document.

h) Untracked Job - a non-UPnP job that is not visible to a UPnP control point, i.e., does not have a JobId and does not appear in the JobIdList, and on which the control point cannot perform any of the Job operations defined in this document.

### 2.2.3. Notation: use of quotation marks

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

## 2.3. References

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

[DEVICE] - UPnP Device Architecture, version 1.0.

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

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

[PWG5101.1] *IEEE-ISTO 5101.1-2001 Media Standardized Names <work in progress>, ftp://ftp.pwg.org/pub/pwg/standards/pwg5101.1.pdf, .doc, .rtf for standardized names*

[UPnP-ENHANCED] - Albright, S., Hastings, T., Zehler, P., and G Shults, "PrintEnhancedLayout:0.10 Service Template For UPnP Version 1.0", work in progress, TBD, 2001.

[XHTML-PRINT] - “XHTML (tm) - Print”, version 0.60, May 11, 2001, <work in progress>, Available at: ftp://ftp.lexmark.com/pub/standards/xhtml-print.pdf

[MULTIPLEXED] - R. Herriot, "The MIME Application/Multiplexed Content-type", June 26, 2001, available at: http://search.ietf.org/internet-drafts/draft-herriot-application-multiplexed-04.txt (Subsequent versions, if any, will be available from the same location with the “04” incremented, and eventually as an information RFC.)

## 2.4. Intent of a Print Job

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

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

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

### 2.4.1. Production vs. Layout Job Attributes

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

Job attributes are partitioned between Production and Layout as follows:

Production Job Attributes (Job Attributes takes precedence):
JobName
JobOriginatingUserName
Copies
Sides
NumberUp
PrintQuality

Layout Job Attributes (data stream takes precedence):
OrientationRequested
MediaSize
MediaType

### 2.4.2. Precedence of Production vs. Layout Job Attributes

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

**Table 1: Precedence of Production and Layout Job Attributes**

| Type of job attribute | IN parameter | PDL Data Stream | Printer SHOULD |
|---|---|---|---|
| Production attribute: | <Distinguished Value> | absent | use <defaultValue> in SCPD |
| | X | absent | use X |
| | <Distinguished Value> | Y | use Y |
| | X | X | use X |
| | X | Y | use X (IN *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) ** |

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

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

## 2.5. State Variables

### 2.5.1. Derived data types

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

#### *2.5.1.1. Comma Separated Value (CSV) Lists*

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

- A list is represented as a UPnP String type.
- Values within a list are separated by commas.
- Only three value types are used as CSV elements in this specification—string, integer and boolean.
- Integer values are represented in CSVs with the same syntax as the int data type specified in [DEVICE] (i.e., optional leading sign, optional leading zeroes)
- Boolean values are represented in CSVs as either ‘0’ for false or ‘1’ for true (which is a subset of the defined boolean data type values specified in [DEVICE]: ‘0’, ‘false’, ‘no’, ‘1’, ‘true’, ‘yes’.
- String values are represented in CSVs with the same syntax as the string data types specified in [DEVICE] (i.e., any Unicode string), with two exceptions that are represented using a backslash escape character:
  - The comma (‘,’) is represented as ‘\,’.
  - The backslash (‘\’) is represented as ‘\\’.

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

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*. |

#### *2.5.1.2. State variables, actions and action parameters*

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

## 2.6. Service State Table

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

### 2.6.1. The Printer's supported and default values

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

Each SST variable definition in this document specifies whether or not vendors in their Service Description MAY subset and/or extend the <allowedValueList> element in their Service Description from those "Allowed Values" values given in this document. The Printer's "current" <allowedValueList> and <defaultValue> values may or may not be the same as the factory supported and default values, respectively, for that parameter, i.e., someone may have changed the settings from the factory-supplied values. Any <allowedValueList> and <defaultValue> element value MAY be changed at any time after Service Discovery. Furthermore, the current

<allowedValueList> and <defaultValue> values for a job parameter could also possibly change between invocations of the action that uses it, for example, if someone reconfigures the Printer's "current" device setting for that parameter. However, the UPnP Device Architecture, version 1.0 [DEVICE], states that any change to the <allowedValueList> or <defaultValue> element requires the printer to issue an "ssdp:byebye" and then re-advertise itself. Each of the values in the <defaultValue> elements is implementation specific, but MUST be one of the values from the Service Description's associated <allowedValueList> element, if present.

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

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

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

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

### 2.6.3. Purposes of the SST State Variables

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

**Table 2: State Variables**

| Variable Name | Req. or Opt.[1] | Data Type | Allowed Value | Default Value | Eng. Units |
|---|---|---|---|---|---|
| *Printer Attributes* | | | | | |
| *VariableName* | *R* | *string* | *TBD* | *DEFAULT* | *TBD* |
| ***PrinterName*** | *R* | *string* | | *<implementation specific>* | *N/A* |
| ***PrinterLocation*** | *R* | *string* | | *<implementation specific>* | *N/A* |
| ***DeviceId*** | *R* | *string* | | *<implementation specific>* | *N/A* |
| ***PrinterState*** | *R* | *string* | *See section 2.6.7* | *idle* | *N/A* |
| ***PrinterStateReasons*** | *R* | *string* | *See section 2.6.8* | *none* | *N/A* |
| ***XHTMLImageSupported*** | *R* | *string* | *See section 2.6.9* | *image/jpeg* | *N/A* |
| ***ColorSupported*** | *R* | *boolean* | *See section 2.6.10* | *<implementation specific>* | *N/A* |
| ***JobIdList*** | *R* | *string (CSV i4)* | *See section 2.6.11* | *<empty String>* | *N/A* |
| ***JobId*** | *R* | *i4* | *Range: 0 to $2^{31}$-1* | *0* | *N/A* |
| ***JobEndState*** | *R* | *string (CSV i4,string, string,i4, string)* | *See section 2.6.13* | *<empty String>* | *N/A* |

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

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

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

### 2.6.4. *PrinterName*

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

*(See [MODEL] section 4.4.4)*

### 2.6.5. *PrinterLocation*

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

(See [MODEL] section 4.4.4)

### 2.6.6. *DeviceId*

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

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

```
key: value {,value} repeated for each key
```

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

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

```
MANUFACTURER:ACME Manufacturing;

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

MODEL:LaserBeam 9;

COMMENT:Anything you like;

ACTIVE COMMAND SET:PCL;
```

*(See IEEE 1284-2000 clause 7.6)*

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

### 2.6.7. *PrinterState*

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

> ***idle*** *- new jobs can start processing immediately without waiting.*
>
> ***processing*** *- jobs (Tracked or Untracked) are processing; new jobs will wait before processing, i.e., are said to be pending.*
>
> ***stopped*** *- no jobs can be processed and intervention is needed.*

*(See [MODEL] section 4.4.11)*

*Vendors MUST NOT subset or extend allowed values.*

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

| Value | Req. or Opt. |
|---|---|
| *idle* | *R* |
| *processing* | *R* |
| *stopped* | *R* |

## 2.6.8. *PrinterStateReasons*

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**Table *2.2:* 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* |
| *media-change-request* | *O* |
| *Vendor-defined* | *O* |

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

### 2.6.9. *XHTMLImageSupported*

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

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

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

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

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

| Value | Req. or Opt. |
|---|---|
| *image/jpeg* | *R* |
| *Vendor-defined* | *O* |

### 2.6.10. *ColorSupported*

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

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

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

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

**Table 2.4: allowedValueList for ColorSupported**

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

## 2.6.11. JobIdList

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

*The list is a sequence of Comma Separated i4 Values (CSV i4 - see section 2.5.1.1). Each value is a JobId of a job on 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.*

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

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

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

## 2.6.12. JobId

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

(See [MODEL] section 4.3.2)

## 2.6.13. 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 via polling.*

*JobEndState is a heterogeneous CSV list of five items: JobId, JobName, JobOriginatingUserName, JobMediaSheetsCompleted, 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 2.6.12)*

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

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

***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 2.6.25.*

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

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

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

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

### *2.6.14. JobName*

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

### *2.6.15. JobOriginatingUserName*

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

### 2.6.16. *DocumentFormat*

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

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

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

*(See [MODEL] section 4.1.9)*

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

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

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

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

### 2.6.17. *Copies*

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

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

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

*Vendors MUST NOT extend the allowed values.*

### 2.6.18. *Sides*

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

***one-sided***
***two-sided-long-edge***
***two-sided-short-edge***
***device-setting***

*(See [MODEL] section 4.2.8)*

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

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

*Vendors MUST NOT extend allowed values.*

**Table *2.6:* allowedValueList for *Sides***

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

### 2.6.19. *NumberUp*

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

***1** - One page per side.*
***2** - Two pages per side.*
***4** - Four pages per side.*
***device-setting***

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

*(See [MODEL] section 4.2.9)*

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

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

**Table 2.7: allowedValueList for *NumberUp***

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

## 2.6.20. *OrientationRequested*

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

***portrait***
***landscape***
***reverse-landscape***
***reverse-portrait***
***device-setting***

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

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

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

*Vendors MUST NOT extend allowed values.*

**Table 2.8: allowedValueList for *OrientationRequested***

| Value | Req. or Opt. |
|---|---|
| *device-setting* | *R* |
| *portrait* | *R* |
| *landscape* | *O* |
| *reverse-landscape* | *O* |
| *reverse-portrait* | *O* |

## 2.6.21. *MediaSize*

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

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

*custom_index-card_3.5x5in*

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

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

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

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

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

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

**Table 2.9: allowedValueList for MediaSize**

| Value[3] | Req. or Opt. |
|---|---|
| *device-setting* | *R* |
| *na_letter_8.5x11in* | *O* |
| *na_legal_8.5x14in* | *O* |
| *iso_a4_210x297mm* | *O* |
| *iso_c5_162x229mm* | *O* |
| *iso_dl_110x220mm* | *O* |
| *jis_b4_257x364mm* | *O* |
| *custom_xxx_IIIxJJJmm* | *O* |
| *custom_xxx_IIIxJJJin* | *O* |
| *custom_min_IIIxJJJmm* | *O* |
| *custom_max_IIIxJJJin* | *O* |
| *Vendor-defined (see [PWG5101.1]* | *O* |

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

## 2.6.22. MediaType

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

| | |
|---|---|
| ***stationery*** | *Separately cut sheets of an opaque material* |
| ***transparency*** | *Separately cut sheets of a transparent material* |
| ***envelope*** | *Envelopes that can be used for conventional mailing purposes* |
| ***labels*** | *Label stock [For example, a sheet of peel-off labels].* |
| ***photographic*** | *Separately cut sheets of an opaque material to produce photographic quality images* |
| ***cardstock*** | *Separately cut sheets of an opaque material that is heavier and stiffer than stationery.* |

***device-setting*** *Indicates that the control point wants the Printer to use its <defaultValue> value for MediaType.*

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

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

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

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

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

| Value[3] | Req. or Opt. |
|---|---|
| *device-setting* | *R* |
| *stationery* | *O* |
| *stationery-inkjet* | *O* |
| *transparency* | *O* |
| *envelope* | *O* |
| *labels* | *O* |
| *photographic* | *O* |
| *cardstock* | *O* |
| *Vendor-defined (see [PWG5101.1]* | *O* |

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

## 2.6.23. *PrintQuality*

*Specifies the print quality requested for the job. Values:*

***draft***
***normal***
***high***
***device-setting***

*(See [MODEL] section 4.2.13)*

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

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

*Vendors MUST NOT extend allowed values.*

**Table *2.11:* allowedValueList for *PrintQuality***

| Value | Req. or Opt. |
|---|---|
| *device-setting* | *R* |
| *draft* | *O* |
| *normal* | *R* |
| *high* | *O* |

## 2.6.24. *DataSink*

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

## 2.6.25. *JobMediaSheetsCompleted*

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

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

## 2.7. Eventing and Moderation

**Table 3: 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* |

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

### 2.7.1. Event Model

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

### 2.7.2. Synchronization of Evented Variables

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

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

For purposes of this document, atomically means:

1. From the viewpoint of any UCP observer external to the Print Service, all of the values change at the same time. To achieve this, it is RECOMMENDED that all evented variables changed by this collected set of actions appear in a single event message.

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

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

**Table 4: Synchronization of Evented Variables**

| State | Transition events (and conditions) | Transition actions |
|---|---|---|
| ? | Initialize PrintBasic service | I, R0, J0, M0, E0 |
| idle | CreateJob or CreateEnhancedJob or create non-UPnP tracked job | P, J1 |
| | Create untracked job — action invoked by non-UPnP entity | P |
| | <printer error> | S, R1 |
| processing | CreateJob or CreateEnhancedJob or create non-UPnP tracked job | P, J2 |
| | Create untracked job — action invoked by non-UPnP entity | *invisible* |
| | Terminate active job that was tracked.<br>Its termination condition, *T*, is one of 'successful', 'canceled' or 'aborted'. | |
| | No more jobs. | I, J0, M0, E1(*T*) |
| | Next job is tracked. | J3, M0, E1(*T*) |
| | Next job is untracked, and there are no more tracked jobs. | J0, M0, E1(*T*) |
| | Next job is untracked, but there are still tracked jobs in the queue. | J4, M0, E1(*T*) |
| | Terminate tracked job that was not active.<br>Its termination condition, *T*, is one of 'canceled' or 'aborted'. | J5, E2(*T*) |
| | Terminate active job that was untracked. | |
| | No more jobs. | I, M0 |
| | Next job is tracked. | J6, M0 |
| | Next job is untracked. | *invisible* |
| | Terminate inactive job that was untracked. | *invisible* |
| | Drop a sheet into the output tray that is not the last sheet of the job. | |
| | Job is tracked. | M1 |
| | 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) |
| | The active job was lost. It was tracked; the next job is untracked. | S, R1, J4, M0, E1(aborted) |
| | The active job was lost. It was untracked; the next job is tracked. | S, R1, J6, M0 |
| | The active job was lost. It was untracked; the next job is untracked. | S, R1 |
| stopped | All problems corrected. | |
| | No jobs are queued. | I, R0 |
| | Jobs are queued. | P, R0 |
| | The reported problem is fixed, but another problem still exists. | R2 |
| | CreateJob or CreateEnhancedJob or create non-UPnP tracked job | |
| | JobIdList is empty. | J1, M0 |
| | JobIdList is not empty. | J2 |
| | Create untracked job. | *invisible* |

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

| Variable(s) affected | Label | New variable value(s) | Action Descriptions |
|---|---|---|---|
| Printer State | I | idle | Printer enters idle state. |
| | P | processing | Printer enters processing state. |
| | S | stopped | Printer enters stopped state. |
| PrinterState Reasons | R0 | none | Printer is operating normally, there are no problems to report. |
| | R1 | <reason> | Old value was 'none'. New value is the reason the printer is in the current PrinterState (§ 2.6.7) |
| | R2 | <new reason> | Old value was something other than 'none'. New value is still not 'none', but is different from old value. |
| JobIdList, JobId | J0 | JobIdList ← {}<br>JobId ← 0 | New list value is empty. |
| | J1 | JobIdList ← {$id_1$}<br>JobId ← $id_1$ | New list contains single job |
| | J2 | JobIdList ← {…,$id_n$,$id_{n+1}$} *OR* {…,$id_i$,$id_{n+1}$,$id_{i+1}$,… ,$id_n$ }<br>*<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 ← {$id_2$,…}<br>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 ← {$id_2$,…}<br>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 ← {…,$id_{i-1}$,$id_{i+1}$,…}<br>*<no change to JobId>* | Old list had at least two jobs. New list has same contents as old *except* the $i^{th}$ job, where i > 1, has been removed. |
| | J6 | *<no change to JobIdList>*<br>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'. |

## 2.8. Actions

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

**Table 6: Actions**

| Name | Req. or Opt. [1] |
|---|---|
| *CreateJob* | *R* |
| *CancelJob* | *R* |
| *GetPrinterAttributes* | *R* |
| *GetJobAttributes* | *R* |
| *Non-standard actions implemented by a UPnP vendor go here.* | X |

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

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

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

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

### 2.8.1. CreateJob

This action is the first step in submitting a job to the printer. The Printer returns a unique JobId to identify the job for this service. The Printer generates the JobId in an implementation-defined manner. The Printer MUST return values in the range 1 to $2^{31}$-1; 0 and negative values are invalid values to be returned as a result of a CreateJob action. Furthermore, the Printer SHOULD NOT re-use values recently assigned, since UCPs would 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 2.3). 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 (UCP) supplies input parameters that are unsupported or their values are unsupported (except DocumentFormat), the Printer (1) MUST accept the CreateJob request, (2) MUST ignore or substitute supported values, respectively, and (3) MUST print the job. This behavior corresponds to the 'false' or omitted value of the IPP "ipp-attribute-fidelity" operation attribute. However, unlike IPP, the Printer does not return any indication that attributes are being ignored or that values are being substituted.
3. If a client (UCP) supplies a conflicting combination of MediaSize and MediaType (or any other set of IN parameters), the Printer MUST accept the CreateJob request, (2) MUST ignore or substitute the conflicting values, and (3) MUST print the job. Whether or not a Printer can detect combinations of different parameter values that are not supported, such as combinations of MediaType and MediaSize values that are not supported, is IMPLEMENTATION-DEPENDENT. If an implementation does detect combinations that are not supported, it substitutes values for one or more parameters to give a combination that is supported.

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

#### *2.8.1.1. Arguments*

**Table 7: 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* |

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

#### *2.8.1.2. Errors*

| errorCode | errorDescription | Description |
|---|---|---|
| Codes 401, 402, 403, 501, 600-99 from the table Common Action Error Codes (below) | See the table Common Action Error Codes (below) | See the table Common Action Error Codes (below) |
| *720* | *ClientErrorFormatNotSupported* | *The supplied DocumentFormat parameter value is not supported by the Printer object.*<br>*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.* |

## 2.8.2. CancelJob

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

### 2.8.2.1. Arguments

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

| Argument | Direction | relatedStateVariable |
|---|---|---|
| *JobId* | *IN* | *JobId* |

### 2.8.2.2. Errors

| errorCode | errorDescription | Description |
|---|---|---|
| Codes 401, 402, 403, 501, 600-99 from the table Common Action Error Codes (below) | See the table Common Action Error Codes (below) | See the table Common Action Error Codes (below) |
| *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.* |

### 2.8.2.3. Effect on State

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

## 2.8.3. GetPrinterAttributes

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

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

#### *2.8.3.1. Arguments*

**Table 9: Arguments for *GetPrinterAttributes***

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

#### *2.8.3.2. Errors*

| errorCode | errorDescription | Description |
|---|---|---|
| Codes 401, 402, 403, 501, 600-99 from the table Common Action Error Codes (below) | See the table Common Action Error Codes (below) | See the table Common Action Error Codes (below) |
| *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.* |

### *2.8.4. GetJobAttributes*

The GetJobAttributes action allows a client (UCP) to determine some of the values of job-related variables 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 are maintained in the JobIdList variable. These variables allow end users to identify their job (i.e., “JobName”, “JobOriginatingUserName”). Other information can be derived from the GetJobAttributes action.

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

#### *2.8.4.1. Arguments*

**Table 10: Arguments for *GetJobAttributes***

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

#### *2.8.4.2. Errors*

| errorCode | errorDescription | Description |
|---|---|---|
| Codes 401, 402, 403, 501, 600-99 from the table Common Action Error Codes (below) | See the table Common Action Error Codes (below) | See the table Common Action Error Codes (below) |
| *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.* |

### 2.8.5. HTTP Post

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

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

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

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

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

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

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

### 2.8.7. Common Error Codes

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

**Table 11: Common Error Codes**

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

## 2.9. Theory of Operation

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

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 (UCP) can determine which CreateJob action parameter values a Print Service implementation supports using the values returned in the <allowedValueList> element of the Service Description.

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

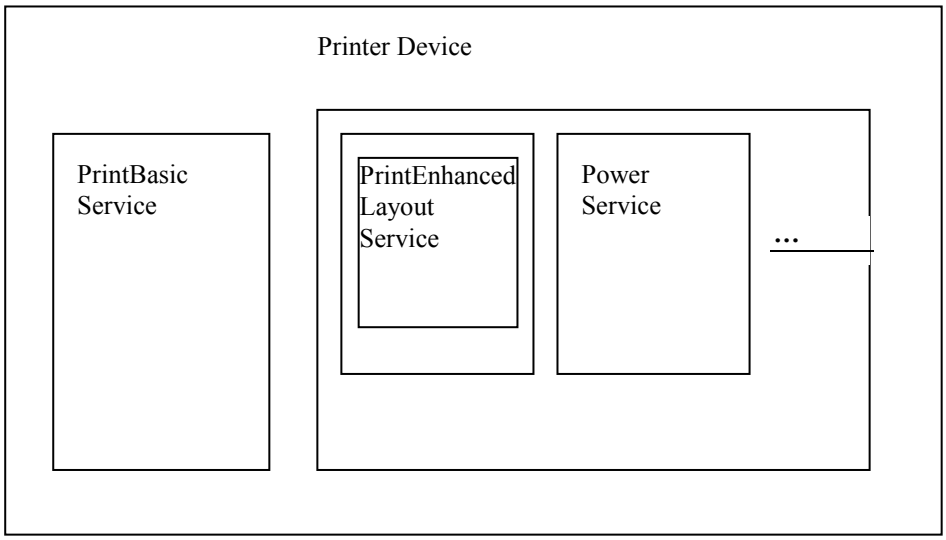


**Figure 1 – Printer Device and Services**

### 2.9.1. 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 allocated by the device. The [MODEL] describes the rules for JobId production (1 to 2**31-1). The JobId is returned by the CreateJob action.

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

- The complete list of known jobs is made available as a state variable represented as CSV list (see section 2.5.1.1) called **JobIdList**.
- All waiting jobs appear in the **JobIdList** variable even those that the device has decided not to print for some reason (they are in the IPP ‘pending’ or ‘pending-held’ job state).
- The order of jobs in the **JobIdList** variable indicates the order in which the jobs will be initiated.
- The job that is actually printing at the moment (or for which the Print Service is stopped) is called the current job. If the current job is “Tracked” (see 2.2.2g), its job identifier is stored in the **JobId** Print Service state variableand that same JobId value is also the first **JobId** in the **JobIdList**. If there is no current job, i.e., there are no jobs, or all jobs are pending or held, or an Untracked Job (2.2.2h) is printing, the **JobId** is 0.
- Once a job has been printed (or cancelled or aborted) it no longer appears in the **JobIdList**, whether or not the Printer has any other jobs to print.

- When the Print Service has no tracked jobs to print, the **JobIdList** state variable is an empty string.

### 2.9.2. Actions

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

- CreateJob. This action is used to submit a job to the printer. The allocated JobId is returned.
- CancelJob. This can be used to cancel a job using the JobId.
- GetPrinterAttributes. This action can be used to query some of the printer attributes.
- GetJobAttributes. This action can be used to query some of the job attributes of a specified job.

### 2.9.3. 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 five evented state variables, JobIdList, JobEndState, PrinterState, PrinterStateReasons, and JobMediaSheetsCompleted 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 UCP 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 CreateJob action. Similarly a UCP can determine that a job has completed, whether successful of not, by matching the JobId for that job with the first element of the evented JobEndState variable.

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

### 2.9.4. 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 adopt the model that IPP uses – which is in fact fairly simple.

### 2.9.5. 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 UCP (client) is expected to localize the well-known string values (that correspond to IPP keyword values) to the locale of its user. The UCP (client) is expected to convert the enum integer values to human readable string values in the locale of the user.

## 2.9.6. IPP Data Type mapping to UPnP Data Types

Basic IPP data types are transformed as follows.

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

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

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

**Table 13: Derived data type mappings**

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

**Table 14: Structured Data Type mapping**

| IPP Type | UPnP equivalent |
|---|---|
| resolution | This is represented as a pair of integers <Attribute Name>X and <attribute Name>Y |
| 1setOf X | See the earlier discussion on arrays in section 2.5.1.1. |

# 3. XML Service Description

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <actionList>
    <action>
    <name>CreateJob</name>
      <argumentList>
        <argument>
          <name>JobName</name>
          <direction>in</direction>
          <relatedStateVariable>JobName</relatedStateVariable>
        </argument>
        <argument>
          <name>JobOriginatingUserName</name>
          <direction>in</direction>

<relatedStateVariable>JobOriginatingUserName</relatedStateVariable>
        </argument>
        <argument>
          <name>DocumentFormat</name>
          <direction>in</direction>
          <relatedStateVariable>DocumentFormat</relatedStateVariable>
        </argument>
        <argument>
          <name>Copies</name>
          <direction>in</direction>
          <relatedStateVariable>Copies</relatedStateVariable>
        </argument>
        <argument>
          <name>Sides</name>
          <direction>in</direction>
          <relatedStateVariable>Sides</relatedStateVariable>
        </argument>
        <argument>
          <name>NumberUp</name>
          <direction>in</direction>
          <relatedStateVariable>NumberUp</relatedStateVariable>
        </argument>
        <argument>
          <name>OrientationRequested</name>
          <direction>in</direction>
          <relatedStateVariable>OrientationRequested</relatedStateVariable>
        </argument>
        <argument>
          <name>MediaSize</name>
          <direction>in</direction>
          <relatedStateVariable>MediaSize</relatedStateVariable>
        </argument>
        <argument>
          <name>MediaType</name>
          <direction>in</direction>
          <relatedStateVariable>MediaType</relatedStateVariable>
        </argument>
        <argument>
          <name>PrintQuality</name>
          <direction>in</direction>
          <relatedStateVariable>PrintQuality</relatedStateVariable>
        </argument>
```

```
      <argument>
        <name>JobId</name>
        <direction>out</direction>
        <relatedStateVariable>JobId</relatedStateVariable>
      </argument>
      <argument>
        <name>DataSink</name>
        <direction>out</direction>
        <relatedStateVariable>DataSink</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
  <action>
  <name>CancelJob</name>
    <argumentList>
      <argument>
        <name>JobId</name>
        <direction>in</direction>
        <relatedStateVariable>JobId</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
  <action>
  <name>GetPrinterAttributes</name>
    <argumentList>
      <argument>
        <name>PrinterState</name>
        <direction>out</direction>
        <relatedStateVariable>PrinterState</relatedStateVariable>
      </argument>
      <argument>
        <name>PrinterStateReasons</name>
        <direction>out</direction>
        <relatedStateVariable>PrinterStateReasons</relatedStateVariable>
      </argument>
      <argument>
        <name>JobIdList</name>
        <direction>out</direction>
        <relatedStateVariable>JobIdList</relatedStateVariable>
      </argument>
      <argument>
        <name>JobId</name>
        <direction>out</direction>
        <relatedStateVariable>JobId</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
  <action>
  <name>GetJobAttributes</name>
    <argumentList>
      <argument>
        <name>JobId</name>
        <direction>in</direction>
        <relatedStateVariable>JobId</relatedStateVariable>
      </argument>
      <argument>
        <name>JobName</name>
        <direction>out</direction>
        <relatedStateVariable>JobName</relatedStateVariable>
      </argument>
      <argument>
        <name>JobOriginatingUserName</name>
        <direction>out</direction>
```

```
<relatedStateVariable>JobOriginatingUserName</relatedStateVariable>
        </argument>
        <argument>
          <name>JobMediaSheetsCompleted</name>
          <direction>out</direction>

<relatedStateVariable>JobMediaSheetsCompleted</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
  </actionList>
  <serviceStateTable>
    <stateVariable sendEvents="no">
      <name>PrinterName</name>
      <dataType>string</dataType>
      <defaultValue></defaultValue>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>PrinterLocation</name>
      <dataType>string</dataType>
      <defaultValue></defaultValue>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>DeviceId</name>
      <dataType>string</dataType>
      <defaultValue></defaultValue>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>PrinterState</name>
      <dataType>string</dataType>
      <defaultValue>idle</defaultValue>
      <allowedValueList>
        <allowedValue>idle</allowedValue>
        <allowedValue>processing</allowedValue>
        <allowedValue>stopped</allowedValue>
      </allowedValueList>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>PrinterStateReasons</name>
      <dataType>string</dataType>
      <defaultValue>none</defaultValue>
      <allowedValueList>
        <allowedValue>none</allowedValue>
        <allowedValue>attention-required</allowedValue>
        <allowedValue>media-jam</allowedValue>
        <allowedValue>paused</allowedValue>
        <allowedValue>door-open</allowedValue>
        <allowedValue>media-low</allowedValue>
        <allowedValue>media-empty</allowedValue>
        <allowedValue>output-area-almost-full</allowedValue>
        <allowedValue>output-area-full</allowedValue>
        <allowedValue>marker-supply-low</allowedValue>
        <allowedValue>marker-supply-empty</allowedValue>
        <allowedValue>marker-failure</allowedValue>
        <allowedValue>media-change-request</allowedValue>
      </allowedValueList>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>XHTMLImageSupported</name>
      <dataType>string</dataType>
      <defaultValue>image/jpeg</defaultValue>
      <allowedValueList>
        <allowedValue>image/jpeg</allowedValue>
```

```
    </allowedValueList>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>ColorSupported</name>
    <dataType>boolean</dataType>
    <defaultValue></defaultValue>
  </stateVariable>
  <stateVariable sendEvents="yes">
    <name>JobIdList</name>
    <dataType>string</dataType>
    <defaultValue></defaultValue>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>JobId</name>
    <dataType>i4</dataType>
    <defaultValue>0</defaultValue>
    <allowedValueRange>
      <minimum>0</minimum>
      <maximum>2147483647</maximum>
      <step>1</step>
    </allowedValueRange>
  </stateVariable>
  <stateVariable sendEvents="yes">
    <name>JobEndState</name>
    <dataType>string</dataType>
    <defaultValue></defaultValue>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>JobName</name>
    <dataType>string</dataType>
    <defaultValue></defaultValue>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>JobOriginatingUserName</name>
    <dataType>string</dataType>
    <defaultValue></defaultValue>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>DocumentFormat</name>
    <dataType>string</dataType>
    <defaultValue>unknown</defaultValue>
    <allowedValueList>
      <allowedValue>unknown</allowedValue>
      <allowedValue>application/vnd.pwg-xhtml-print</allowedValue>
      <allowedValue>application/octet-stream</allowedValue>
      <allowedValue>text/plain</allowedValue>
      <allowedValue>text/plain;charset=utf-8</allowedValue>
      <allowedValue>application/postscript</allowedValue>
      <allowedValue>application/vnd.hp-PCL</allowedValue>
    </allowedValueList>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>Copies</name>
    <dataType>i4</dataType>
    <defaultValue>1</defaultValue>
    <allowedValueRange>
      <minimum>0</minimum>
      <maximum>2147483647</maximum>
      <step>1</step>
    </allowedValueRange>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>Sides</name>
```

```
      <dataType>string</dataType>
      <defaultValue>one-sided</defaultValue>
      <allowedValueList>
        <allowedValue>one-sided</allowedValue>
        <allowedValue>two-sided-long-edge</allowedValue>
        <allowedValue>two-sided-short-edge</allowedValue>
        <allowedValue>device-setting</allowedValue>
      </allowedValueList>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>NumberUp</name>
      <dataType>string</dataType>
      <defaultValue>1</defaultValue>
      <allowedValueList>
        <allowedValue>1</allowedValue>
        <allowedValue>2</allowedValue>
        <allowedValue>4</allowedValue>
        <allowedValue>device-setting</allowedValue>
      </allowedValueList>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>OrientationRequested</name>
      <dataType>string</dataType>
      <defaultValue>portrait</defaultValue>
      <allowedValueList>
        <allowedValue>portrait</allowedValue>
        <allowedValue>landscape</allowedValue>
        <allowedValue>reverse-landscape</allowedValue>
        <allowedValue>reverse-portrait</allowedValue>
        <allowedValue>device-setting</allowedValue>
      </allowedValueList>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>MediaSize</name>
      <dataType>string</dataType>
      <defaultValue></defaultValue>
      <allowedValueList>
        <allowedValue>na_letter_8.5x11in</allowedValue>
        <allowedValue>na_legal_8.5x14in</allowedValue>
        <allowedValue>iso_a4_210x297mm</allowedValue>
        <allowedValue>iso_c5_162x229mm</allowedValue>
        <allowedValue>iso_dl_110x220mm</allowedValue>
        <allowedValue>jis_b4_257x364mm</allowedValue>
        <allowedValue>device-setting</allowedValue>
      </allowedValueList>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>MediaType</name>
      <dataType>string</dataType>
      <defaultValue></defaultValue>
      <allowedValueList>
        <allowedValue>stationery</allowedValue>
        <allowedValue>transparency</allowedValue>
        <allowedValue>envelope</allowedValue>
        <allowedValue>labels</allowedValue>
        <allowedValue>photographic</allowedValue>
        <allowedValue>cardstock</allowedValue>
        <allowedValue>device-setting</allowedValue>
      </allowedValueList>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>PrintQuality</name>
      <dataType>string</dataType>
```

```
      <defaultValue>normal</defaultValue>
      <allowedValueList>
        <allowedValue>draft</allowedValue>
        <allowedValue>normal</allowedValue>
        <allowedValue>high</allowedValue>
        <allowedValue>device-setting</allowedValue>
      </allowedValueList>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>DataSink</name>
      <dataType>uri</dataType>
      <defaultValue></defaultValue>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>JobMediaSheetsCompleted</name>
      <dataType>i4</dataType>
      <defaultValue>0</defaultValue>
      <allowedValueRange>
        <minimum>-1</minimum>
        <maximum>2147483647</maximum>
        <step>1</step>
      </allowedValueRange>
    </stateVariable>
  </serviceStateTable>
</scpd>
```

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

Tel: + 41 22 919 02 11
Fax: + 41 22 919 03 00
info@iec.ch
www.iec.ch