PODi

DPT 3.0

September 2011

Digital Print Ticket —

Printing with PPML and JDF

PODi: the Digital Printing Initiative
1240 Jefferson Road, Rochester, New York 14623, USA
Tel: (585) 239-6014
Internet: http://www.podi.org

PODi the Digital Printing Initiative

Approval of a PODi standard requires acceptance by the members of PODi.

PODi is a not for profit industry consortium formed in 1996. Its charter is to foster the growth of the digital printing industry through market and standards development activities. PODi constantly monitors market and technology trends in the industry, and shares information through seminars, independent research, white papers, articles, and the web. PODi promotes interoperability through the PPML suite of open, XML based standards, test suites and certification.

PODi welcomes feedback on this specification, and offers the following services to support widespread adoption of the specification:

Specification Updates

> The PPML specification is distributed free of charge. Developers who are implementing the PPML standard are invited to subscribe to free PPML updates and the technical note service.

Developer Support web site

> Software and hardware developers interested in supporting PPML are invited to register for the PPML Developers discussion group.

To participate in the PPML initiative, send an email to ppmlinfo@podi.org.

Contents

Foreword

PODi: the Digital Printing Initiative developed this Digital Print Ticket (DPT) specification in response to a need expressed by its members to define the semantics for how the Personalized Print Markup Language (PPML) and the CIP4 Job Definition Format (JDF) interact.

This specification restricts itself to defining the semantics of how PPML and JDF interact. It extends version 2.2 to allow for the use of features from JDF 1.4 and PPML 3.0. Note that the DPT specification can be used with any version of PPML and JDF.

The working group responsible for the current specification had the following membership:

PODi Senior Technologist: Dr. Paul Jones

PODi Director of Technology: James Mekis

Subcommittee members:

Adobe: Dov Isaacs
Canon: Peter Wyatt, Daehan Choi
EFI: Reuven Ackner
Hewlett-Packard: Steve Hiebert, Michael Persons, Don Abel
Kodak Creo: Luci Wahrmann
Kodak: Tim Donahue
Konica Minolta: Darrell Hopp
Printable: Doug Cogan
Punch Graphix: Bart Wynants
QualityLogic: Steve Kang, David Kendal
vdptech: Boris Aronshtam
Xerox: John Czudak
XMPie: Peter Davis

Send suggestions for improving this standard to PODi, 1240 Jefferson Road, Rochester, NY 14623, USA; e-mail: ppmlinfo@podi.org.

Introduction

The PODi Digital Print Ticket (DPT) specification defines a conformance subset of Page Printer Markup Language (PPML) data sets for use with the CIP4 Job Definition Format (JDF)[1]. This specification defines how to use JDF semantics when the PPML data is characterized as a MultiSet content format.

DPT 3.0 extends the Digital Print Ticket specification to take advantage of new features of JDF 1.4 and PPML 3.0. These new features enable control of processing at the page, document, and set level.

A set of application notes will be developed for this specification and when completed will be available at http://www.podi.org. Additional tools and test suites are also at this site for developers.

[1] See www.CIP4.org for the JDF Specification and other supporting documentation on JDF.

Digital Print Ticket — Printing with PPML and JDF

1 Scope

This specification defines the semantics for using the Personalized Print Markup Language (PPML) and the Job Definition Format (JDF) for printing applications. This conformance subset of PPML specifies the semantics required to make content data and meta information described in PPML available to a JDF Consumer as described by the JDF specification. This specification also defines a set of minimal requirements for a JDF Device that drives a PPML Consumer.

This specification is not intended to define nor does it define a conformance subset of JDF for use with PPML.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including amendments) applies.

JDF Specification, version 1.4a. December 2009.
http://www.cip4.org

PODi: the Digital Printing Initiative. *Personalized Print Markup Language Functional Specification* (all versions, 1.5/2.1/2.2/2.3/3.0).

ANSI CGATS.20.2002 (PPML/VDX)

XML Path Language (XPath), version 1.0. November 16, 1999.
http://www.w3c.org/TR/XPath

3 Definitions

For the purposes of this specification, the following definitions apply.

3.1
JDF Device
The component of a JDF workflow part that interprets JDF and executes the instructions.

3.2
PPML Consumer
A device, process, or system that reads and interprets PPML data.

3.3
Job Ticket
The instructions required to produce a set of printed documents.

4 Terms, symbols, notations, and abbreviations

Code excerpts, element names, and attributes are written in a bold sans serif font; for example **DOCUMENT_SET.**

The vertical bar character (|) signifies the logical OR operator. For instance, "**SOURCE | OCCURRENCE_REF**" means "**SOURCE** or **OCCURRENCE_REF**".

Many PPML element names are common English words; therefore it is often convenient and accurate to use them conversationally. In this specification, when an element name appears in text *not* in a bold san serif font, but with Initial Capitals, it is specifically referring to the PPML item that bears that name. When it appears with no capitalization, the word is being used with no special PPML significance.

For example:

The **SOURCE** element contains one or more component files.
In an **OBJECT** element, the Source may contain data in any of several formats.
Customers may submit image data that was gathered from a number of different sources.

In tables of XML attributes, when the data type is Number or Integer, a multiplication sign indicates a string of numbers separated by spaces. For instance, “Number × 4” indicates that the value of the attribute should be four numbers, such as ”1.234 2.0 3 4.567.”

Table 1 — Terms used DPT 3.0

Term	Reference
PPML	Personalized Print Markup Language
PPML Specification	*Personalized Print Markup Language Functional Specification*
JDF	Job Definition Format
URL	Uniform Resource Locator defined in the RFC 2396: Uniform Resource Identifiers (URI)
XML	The data format defined by the *Extensible Markup Language (XML) 1.0* specification

5 Conformance

A conforming PPML dataset shall adhere to the technical requirements set forth in 6.1 Requirements for PPML datasets.

A conforming JDF job ticket shall adhere to the technical requirements set forth in 6.2 Requirements for JDF Job Tickets.

NOTE In accordance with the Base Conformance ICS, the JDF job ticket must always be external to the PPML file. To use a DPT 2.0 compliant job ticket embedded in the PPML data, that JDF job ticket must be externalized before submission to the JDF Device. Externalizing a JDF job ticket may require rewriting URLs in the JDF job ticket, in particular self-referencing URLs.

A conforming JDF Device shall adhere to the technical requirements set forth in 6.3 Requirements for JDF Devices, shall conform to the Base ICS Level 0, and may support additional levels.

A conforming PPML Consumer shall adhere to the technical requirements set forth in 6.4 Requirements for PPML Consumers.

.

A conforming JDF Consumer shall adhere to the technical requirements set forth in 6.5 Requirements for accessing PPML content data from JDF.

6 Technical Requirements

This specification identifies requirements for PPML datasets, PPML Consumers, JDF job tickets and JDF Devices driving PPML Consumers.

6.1 Requirements for PPML datasets

6.1.1 Conformance

The PPML element shall include a **CONFORMANCE** element indicating conformance to this specification. That **CONFORMANCE** element shall have a *SubSet* attribute with a value of "DPT" and a Level attribute with a value of "3.0".

6.1.2 Label

The value of the *Label* attribute of a **PPML PAGE**, **DOCUMENT**, **DOCUMENT_SET** and **JOB** elements must conform to the *NMTOKEN* production as defined in the XML specification.

NOTE: The values of a Label attribute need not be unique. It may however be useful to have unique labels, to allow content data to be correlated with the original database of customer information.

6.2 Requirements for JDF Job Tickets

The *ElementType* attribute of a JDF **LAYOUTELEMENT** element shall be present and have the value "MultiSet" if the containing JDF **FILESPEC** element has a *MimeType* attribute with a value of "application/vnd.podi-ppml+xml".

The *Context* attribute on a JDF **METADATAMAP** element shall not contain a value other than "Set", "Document", "Page" or "PagePool".

The *DataType* attribute on a JDF **METADATAMAP** element shall contain the value "PartIDKeys".

The *Name* attribute on a JDF **METADATAMAP** element shall not contain a value other than "Metadata0", "Metadata1", "Metadata2", "Metadata3", "Metadata4", "Metadata5", "Metadata6", "Metadata6", "Metadata7", "Metadata8", "Metadata9", "RunTags", "SetTags", "DocTags", "PageTags" or "Side".

The *Path* attribute of a JDF **EXPR** element shall have a value that conforms to the XPath 1.0 specification.

The *Path* attribute of a JDF element that derives from the abstract JDF **EVALUATION** element shall have a value that conforms to the XPath 1.0 specification.

XPath expressions shall only reference **METADATA, DATUM, PPML, DOCUMENT_SET, JOB, DOCUMENT, PAGE** and **PAGE_DESIGN** elements from the PPML namespace. An XPath expression may reference elements that are contained within a **DATUM** element that are from another XML namespace.

XPath expressions shall return a Boolean, a number, a string or a node-set containing at most one node.

A DPT compliant JDF Job Ticket shall not contain partitioning of resources by partition keys other than: "Metadata0", "Metadata1", "Metadata2", "Metadata3", "Metadata4", "Metadata5", "Metadata6", “Metadata6”, “Metadata7”, “Metadata8”, “Metadata9”, “RunTags”, “SetTags”, “DocTags”, “PageTags”, “RunIndex”, “SetIndex”, “DocIndex”, “PageIndex”, “SetRunIndex”, “DocRunIndex”, “SetDocIndex”, “SheetName”, “SheetIndex”, “Separation”.

6.3 Requirements for JDF Devices

A DPT compliant JDF Device shall model a PPML Consumer as a JDF process node that has a **RunList** that refers to PPML data as an input resource. That process shall model a DPT 3.0 compliant PPML Consumer.

NOTE If the *IgnorePDLImposition* attribute of that JDF **LAYOUTELEMENT** has the value of “true” or is unspecified, the PPML Consumer must process the PPML data as a non-imposing PPML Consumer as defined by the PPML specification.

A DPT compliant JDF Device shall support the use of JDF **METADATAMAP** elements that are permitted for a DPT compliant JDF job ticket and that are embedded in a RunList that refers to PPML data as an input resource

A DPT compliant JDF Device shall support the use of XPath expressions that are permitted for a DPT compliant JDF job ticket. The value of an XPath expression of type node-set that contains a node shall be converted to text by taking the string-value of that node as defined in the XPath specification.

A DPT compliant JDF Device shall support partitioning of resources referenced from a process that models a DPT 3.0 compliant PPML Consumer.

A DPT compliant JDF Device shall support partitioning of resources by the following partition keys “Metadata0”, “Metadata1”, “Metadata2”, “Metadata3”, “Metadata4”, “Metadata5”, “Metadata6”, “Metadata6”, “Metadata7”, “Metadata8”, “Metadata9”, “RunTags”, “SetTags”, “DocTags”, “PageTags”, “RunIndex”, “SetIndex”, “DocIndex”, “PageIndex”, “SetRunIndex”, “DocRunIndex”, “SetDocIndex”, “SheetName”, “SheetIndex”, “Separation”, “Side”.

6.4 Requirements for PPML Consumers

6.4.1 URL

A PPML Consumer shall support at a minimum the same URL resolving capabilities of the JDF Device driving that Consumer to ensure that files referenced from PPML via relative URL's are supported.

NOTE As the JDF Device driving a PPML Consumer is required to support the JDF Base ICS Level 0, the above only implies support for URL's based on the file protocol.

6.4.2 PPML TICKET

The PPML **TICKET_REF** and **TICKET** elements present in the PPML data shall be ignored.

NOTE Support for **TICKET_REF** is strictly for reasons of backwards compatibility with the ANSI CGATS.20.2002 (PPML/VDX) standard. In VDX, the **TICKET_REF** elements are used in defining product intent only. The **TICKET_REF** element is kept in this specification to allow the PPML data to be used unmodified after the intent to process mapping. The use of partitioning product intent resources, rather than **TICKET_REF**, is strongly recommended.

6.5 Requirements for accessing PPML content data from JDF

A PPML file organizes page content into a hierarchical structure whose levels correspond to a content file format as characterized by the *LayoutElement/@ElementType* attribute value of “MultiSet”. The following clauses define the input order of the hierarchical elements as seen by a JDF **RunList** resource definition.

6.5.1 Document_Set

Each PPML **DOCUMENT_SET** or **JOB** element shall define a JDF document set as specified by the definition of the *Sets* attribute of the JDF **RunList** resource in the JDF specification.

6.5.1.2 Input Order

The order in which the PPML **DOCUMENT_SET** or **JOB** elements occur in the PPML element shall define the input order of the document sets to the JDF **RunList** resource.

6.5.1.3 Label

The value of the *Label* attribute of a PPML **DOCUMENT_SET** or **JOB** element shall define the name of the set as used by the JDF *SetNames* attribute of a **RunList** resource.

NOTE The value of the *SetNames* attribute of the JDF **RunList** element identifies for selection those PPML **JOB** or **DOCUMENT_SET** elements whose Label attribute value is included in the *SetNames* attribute of the **RunList** element.

6.5.1.4 Class

The value of the *Class* attribute of a PPML **DOCUMENT_SET** or **JOB** element shall define for the set in JDF defined by that element the value for an implicit label as defined in the description of the *SetTags* partition key in the JDF specification.

NOTE If the value of the Class attribute of a PPML **DOCUMENT_SET** or **JOB** element matches an entry in the value of a *SetTags* attribute of a JDF **Part** element or a JDF Resource partition, then the documents in the identified PPML **DOCUMENT_SET** or **JOB** element will be processed in accordance with the JDF specification.

6.5.2 DOCUMENT

Each PPML **DOCUMENT** element shall define a JDF document as specified by the definition of the **Docs** attribute of the JDF **RunList** resource in the JDF specification.

6.5.2.1 Input Order

The order in which the PPML **DOCUMENT** elements occur in a PPML **DOCUMENT_SET** or **JOB** element shall define the input order of the documents within a set to the JDF **RunList** resource.

6.5.2.2 Label

The value of the *Label* attribute of a PPML **DOCUMENT** element shall define the name of the document as used by the JDF *DocNames* attribute of a **RunList** resource.

The value of the *Label* attribute of a PPML **DOCUMENT** element shall define for the document in JDF defined by that **DOCUMENT** element the value for an implicit label as defined in the description of the *RunTag* attribute of the **RunList** element in the JDF specification.

NOTE The value of the *DocNames* attribute of the JDF **RunList** element identifies for selection those PPML **DOCUMENT** elements whose *Label* attribute value is included in the *DocNames* attribute of the **RunList** element.

NOTE If the value of the *Label* attribute of a PPML **DOCUMENT** element matches an entry in the value of a *RunTags* attribute of a JDF **PART** element or a JDF Resource partition, then the pages in the identified PPML **DOCUMENT** element will be processed in accordance with the JDF specification.

6.5.2.3 Class

The value of the *Class* attribute of a PPML **DOCUMENT** element shall define for the document in JDF defined by that **DOCUMENT** element the value for an implicit label as defined in the description of the **DocTags** partition key in the JDF specification.

NOTE If the value of the *Class* attribute of a PPML **DOCUMENT** element matches an entry in the value of a *DocTags* attribute of a JDF **PART** element or a JDF Resource partition, then the pages in the identified PPML **DOCUMENT** element will be processed in accordance with the JDF specification.

6.5.3 PAGE

Each PPML **PAGE** element shall define a JDF page as specified by the definition of the *Pages* attribute of the JDF **RunList** resource in the JDF specification.

6.5.3.1 Input Order

The order in which the PPML **PAGE** elements occur in a PPML **DOCUMENT** element shall define the input order of the pages within a document to the JDF **RunList** resource.

6.5.3.2 TrimBox

In PPML, each **PAGE** element shall have a defined trim box that takes its value from the *TrimBox* attribute of the PPML **PAGE_DESIGN** or **PRINT_LAYOUT** element in effect.

The reference coordinate for placing a PPML page into the JDF coordinate space shall be the lower-left corner of the region defined by the trim box of that PPML page.

NOTE PPML requires a **PAGE_DESIGN** or **PRINT_LAYOUT** to be in effect for each page, therefore the trim box is always defined.

6.5.3.3 Label

The value of the *Label* attribute of a PPML **PAGE** element shall define the name of the page as used by the JDF *PageNames* attribute of a **RunList** resource.

NOTE The value of the *PageNames* attribute of the JDF **RunList** element identifies for selection those PPML **PAGE** elements whose *Label* attribute value is included in the *PageNames* attribute of the **RunList** element.

6.5.3.4 Class

The value of the *Class* attribute of a PPML **PAGE** element shall define for the page in JDF defined by that **PAGE** element the value for an implicit label as defined in the description of the **PageTags** partition key in the JDF specification.

NOTE If the value of the *Class* attribute of a PPML **PAGE** element matches an entry in the value of a *PageTags* attribute of a JDF **Part** element or a JDF Resource partition, then the page identified by the PPML **PAGE** element will be processed in accordance with the JDF specification.

6.6 METADATA

Each PPML **METADATA** element should contain a single **DATUM** element encoded in accordance with the *Common Metadata for Document Production Workflow ICS*.

Example:

```
<METADATA Creator="ACME">
  <DATUM Key="CIP4:Root">
    <Metadata xmlns="urn:cip4.org:CommonMetadata:CIP4"
xmlns:ACME="urn:cip4.org:CommonMetadata:ACME">
       <Conformance>base</Conformance>
         <Creator>ACME</Creator>
         <ModificationDate>20090723T111423+01:00</ModificationDate>
         <ACME:ProductVersion>1.1.2</ACME:ProductVersion>
         <ACME:Status>Softproof</ACME:Status>
    </Metadata>
  </DATUM>
</METADATA>
```

Annex A
(informative)

Overview of the mapping between PPML and JDF

A.1 PPML Element mapping

Table 1 shows the mapping between PPML elements and the corresponding JDF concept. Note that the order of occurrence of the elements in PPML defines the order of pages, documents and sets in JDF.

Table 1 — PPML elements mapping to JDF

PPML Element	JDF
PAGE	Reader page
DOCUMENT	Document
DOCUMENT_SET	Set
JOB	Set

A.2 PPML attribute mapping

Table 2 shows the mapping between PPML attributes and the JDF partition keys that may be used to select a subset of the PPML content data for processing. If the value of the PPML attribute matches a value in the list of values in the corresponding JDF partition key then the content data described by that PPML element to which that attribute belongs is selected.

Table 2 — PPML attributes mapping to JDF

PPML Element/attribute	JDF
PAGE/@*Label*	RunList/@PageNames
DOCUMENT /@*Label*	RunList/@DocNames, RunTags
DOCUMENT_SET/@*Label*	RunList/@SetNames
JOB/@*Label*	RunList/@SetNames
PAGE */@Class*	PageTags
DOCUMENT */@Class*	DocTags
DOCUMENT_SET */@Class*	SetTags
JOB */@Class*	SetTags

A.3 Requirements for PPML data

Table 3 lists the requirements for PPML elements and attributes in DPT 2.2 compliant PPML data.

Table 3 — Requirements on PPML elements and attributes

PPML element/attribute	Requirement
PAGE/@*Label*	NMTOKEN value
DOCUMENT /@*Label*	NMTOKEN value
DOCUMENT_SET@*Label*	NMTOKEN value
JOB/@*Label*	NMTOKEN value
CONFORMANCE	At least one required with the required Subset/Level attributes
CONFORMANCE/@*SubSet*	Required value "DPT"
CONFORMANCE/@*Level*	Required value "3.0"
TICKET	Ignored
TICKET_REF	Ignored
SHEET_LAYOUT	Ignored if LayoutElement/@IgnorePDLImposition equals "true"
EXTERNAL_DATA/@*Src*	Required support for file protocol.
EXTERNAL_DATA_ARRAY /@*Src*	Required support for file protocol.
SEGMENT_ARRAY/@*Src*	Required support for file protocol.
SUPPLIED_RESOURCE/@ *Src* **(deprecated attribute)**	Required support for file protocol.

A.4 Requirements on JDF job tickets

Table 4 lists the requirements for JDF elements and attributes in a DPT 2.2 compliant JDF job ticket.

Table 4 — Requirements for JDF elements and attributes

JDF Element/attribute	Requirement
LayoutElement/@*ElementType*	Value must be "`MultiSet`" if attribute *MimeType* of containing **FILESPEC** element equals "`application/vnd.podi-ppml+xml`"

JDF Element/attribute	Requirement
MetaDataMap/@*Context*	Limited to “Set”, “Document”, “Page” or “PagePool”
MetaDataMap/@*DataType*	Must be “PartIDKeys”
MetaDataMap/@*Name*	Limited to “Metadata0”, “Metadata1”, “Metadata2”, “Metadata3”, “Metadata4”, “Metadata5”, “Metadata6”, “Metadata6”, “Metadata7”, “Metadata8”, “Metadata9”, “RunTags”, “SetTags”, “DocTags” or “PageTags”.
Expr/@*Path*	XPath 1.0 expression. Only allowed to reference **METADATA, DATUM, PPML, DOCUMENT_SET, JOB, DOCUMENT, PAGE** and **PAGE_DESIGN** as well as non-PPML elements contained in a **DATUM** element.
***Evaluation**/@*Path*	See **Expr**/@*Path*
Any resource	Partition keys are limited to: “Metadata0”, “Metadata1”, “Metadata2”, “Metadata3”, “Metadata4”, “Metadata5”, “Metadata6”, “Metadata6”, “Metadata7”, “Metadata8”, “Metadata9”, “RunTags”, “SetTags”, “DocTags”, “PageTags”, “RunIndex”, “SetIndex”, “DocIndex”, “PageIndex”, “SetRunIndex”, “DocRunIndex”, “SetDocIndex”, “SheetName”, “SheetIndex”, “Separation”, “Side”.

Annex B
(informative)
Revision history

B.1 DPT 3.0 (September 2011)

- Add support for JDF 1.4
- Require support for partitioning with a well-defined set of partition keys
- Require support for **MetaDataMap** with a well-defined set of partition keys
- Require support for evaluation using XPath 1.0 to a well-defined set of elements
- Update Forward to note that this specification extends version 2.2 to allow for the use of features from JDF 1.4 and PPML 3.0.

B.2 DPT 2.2 (September 5th, 2005)

- Map Class attributes in PPML 2.2 to the new partition keys in JDF 1.3.

B.3 DPT 2.0 (May 9th, 2005)

- Revoke DPT 1.0.
- Restrict the scope of the specification to define only the interaction between PPML and JDF.

B.4 DPT 1.0 (2002)

- Initial revision