

# JDF Interoperability Conformance Specification (ICS) for: Digital Printing (DP) - JDF Process Level 1 and Level 2

~~June 10, 2003~~ October 21, 2003

Version 0.0710

File: icsdp-21-Oct-2003.doc, .pdf

Includes support for JDF 1.2

Revisions are in red with change bars in the right margin.

The open ISSUES are highlighted in green.

## Abstract

*[Write a short abstract that will help the reader figure out whether or not to read the Interoperability Conformance Specification (ICS). The Abstract section SHOULD provide a concise and comprehensive overview of the purpose and contents of the entire document, to give a technically knowledgeable reader a general overview of the function of the document. Start with "This CIP4 JDF Interoperability Conformance Specification (ICS) defines ...". Expand any acronyms in their first occurrence in the Abstract. Don't just repeat the first few paragraphs of your Introduction; be briefer in the Abstract.]*

This CIP4 JDF Conformance Specification defines the interoperability requirements for a subset of JDF defined as Level 1 and Level 2 Digital Printing. Level 1 is useful for low end black and white or color desktop printing systems with no provision for in-line finishing. Level 2 is useful for color and black and white integrated digital printing systems with in-line finishing capabilities. Level 2 encompasses production printing IPP semantics and includes Level 1.



## Copyright Notice

Copyright © 2000-2002, International Cooperation for Integration of Processes in Prepress, Press and Postpress, hereinafter referred to as CIP4. All Rights Reserved

Permission is hereby granted, free of charge, to any person obtaining a copy of the Specification and associated documentation files (the “Specification”) to deal in the Specification, including without limitation the rights to use, copy, publish, distribute, and/or sublicense copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the following conditions. The above copyright notice and this permission notice must be included in all copies or substantial portions of the Specification.

THE SPECIFICATION IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED, OR OTHERWISE, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT WILL CIP4 BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF, OR IN CONNECTION WITH THE SPECIFICATION OR THE USE OR OTHER DEALINGS IN THE SPECIFICATION.

Except as contained in this notice or as allowed by membership in CIP4, the name of CIP4 must not be used in advertising or otherwise to promote the use or other dealings in this Specification without prior written authorization from CIP4.

## Licenses and Trademarks

International Cooperation for Integration of Processes in Prepress, Press and Postpress, CIP4, Job Description Format, JDF and the CIP4 logo are trademarks of CIP4.

Rather than put a trademark symbol in every occurrence of other trademarked names, we state that we are using the names only in an editorial fashion, and to the benefit of the trademark owner, with no intention of infringement of the trademark.

## Table of Contents

<b>Copyright Notice .....</b>	<b>2</b>
<b>Licenses and Trademarks .....</b>	<b>2</b>
<b>1 Introduction .....</b>	<b>7</b>
<b>1.1 Overview of JDF .....</b>	<b>7</b>
1.1.1 What is JDF? .....	7
1.1.2 Elements in a JDF Job Ticket .....	7
1.1.3 Content of a JDF Job Ticket .....	9
<b>1.2 Digital Printing Job Ticket .....</b>	<b>10</b>
<b>1.3 Conformance Interface, Producer, and Consumer .....</b>	<b>11</b>
<b>1.4 Scope .....</b>	<b>12</b>
<b>2 Terminology .....</b>	<b>12</b>
<b>2.1 Conformance Terminology .....</b>	<b>13</b>
<b>2.2 Other Terminology .....</b>	<b>13</b>
<b>3 JDF Supported by ICS for Digital Printing .....</b>	<b>14</b>
<b>3.1 Combined Digital Printing Node .....</b>	<b>15</b>
3.1.1 Settings Policy .....	15
3.1.2 Conformance Levels .....	15
3.1.3 Processes .....	15
3.1.3.1 Required Processes .....	15
3.1.3.2 Optional Finishing Processes .....	16
3.1.3.3 Other Optional Processes .....	16
3.1.3.4 Sequence of Processes .....	16
3.1.3.5 Processes and Exchange Resources .....	16
3.1.4 Combined Digital Printing Node Attributes .....	19
3.1.4.1 Types Attribute .....	19
3.1.5 Elements Included in a Combined Digital Printing Node .....	20
3.1.6 Resources .....	20
3.1.6.1 Resource Attributes .....	20
3.1.6.2 Resources and Processes .....	21
3.1.6.3 Resources and ResourcePool .....	21
3.1.6.4 ResourceLinks and ResourceLinkPool .....	22
3.1.7 Content of a Digital Printing JDF Job Ticket .....	23
3.1.8 Resources and Elements Supported by ICS for Digital Printing .....	25
3.1.8.1 Resource References .....	41
3.1.8.2 Partitionable Resources .....	41
3.1.8.3 Using Resource References and Partitionable Resources .....	47
3.1.8.4 Resource Update .....	49
<b>3.2 Features Supported .....</b>	<b>49</b>
3.2.1 Approval .....	51
3.2.2 Billing Code .....	53
3.2.3 Binding .....	54
3.2.4 Black Overprint .....	55
3.2.5 Collate .....	55
3.2.6 Comment/Description of Job .....	56
3.2.7 Contact Information .....	56
3.2.8 Covers .....	58
3.2.9 Destination or Physical Printer Requested .....	61
3.2.10 Destination or Physical Printer Resolution .....	61

3.2.11	Document Compression .....	62
3.2.12	Document File Format .....	63
3.2.13	Document File Name .....	65
3.2.14	Document Natural Language .....	67
3.2.15	Feed Orientation .....	68
3.2.16	Fit Policy .....	69
3.2.17	Folding .....	70
3.2.18	Force Page .....	72
3.2.19	Hold Job .....	74
3.2.20	ICS Versions .....	74
3.2.21	Image Shift Back Side .....	75
3.2.22	Image Shift Front Side .....	76
3.2.23	Input Tray Name .....	76
3.2.24	Insert Sheet .....	78
3.2.25	Interleave Sheets .....	79
3.2.26	Job Created By .....	80
3.2.27	Job Modified By .....	81
3.2.28	Job Name .....	81
3.2.29	Job Priority .....	82
3.2.30	Job Ticket Created By .....	82
3.2.31	Job Ticket Template Comment or Description .....	83
3.2.32	Job Ticket Template ID .....	83
3.2.33	Job Ticket Template Version .....	84
3.2.34	Jog Offset .....	84
3.2.35	Margins .....	85
3.2.36	Media .....	86
3.2.37	Message to Operator .....	92
3.2.38	Notification .....	92
3.2.39	Number of Copies .....	94
3.2.40	Number Up .....	94
3.2.41	Output Bin Name .....	95
3.2.42	Page Delivery .....	95
3.2.43	Page Distribution .....	96
3.2.44	Page Order .....	97
3.2.45	Presentation Direction .....	97
3.2.46	Print Quality .....	98
3.2.47	Process Color Model .....	99
3.2.48	Proof Print .....	100
3.2.49	Punching and Hole Making .....	102
3.2.50	Range of Pages to Include .....	103
3.2.51	Range of Pages to Output .....	103
3.2.52	Rotate Page .....	104
3.2.53	Screening Family .....	105
3.2.54	Settings Policy .....	105
3.2.55	Sides .....	107
3.2.56	Spot Color .....	107
3.2.57	Stapling and Stitching .....	108
3.2.58	Start, Separator/Slip, End Sheets .....	109
3.2.59	Trapping .....	111
3.2.60	Trimming .....	112
<b>3.3</b>	<b>Conformance Level(s), Modules, and/or Profiles .....</b>	<b>116</b>
<b>4</b>	<b>Conformance Requirements for &lt;fill in your ICS&gt; .....</b>	<b>117</b>
<b>4.1</b>	<b>Conformance Requirements for &lt;fill in Producers of your Interface&gt; .....</b>	<b>117</b>
<b>4.2</b>	<b>Conformance Requirements for &lt;fill-in Consumers of your Interface&gt; .....</b>	<b>117</b>

**5 References..... 118**

**5.1 Normative References ..... 118**

**5.2 Informative References ..... 120**

**6 Registration Considerations..... 121**

**7 Internationalization Considerations..... 121**

**8 Security Considerations..... 121**

**9 Contributors ..... 121**

**10 Acknowledgments..... 121**

**11 Author’s Address ..... 122**

**Annex A <fill-in a title for Annex> (Normative) ..... 123**

**Annex B <fill-in a title for Annex> (Informative) ..... 124**

**Annex C Tips for Good Technical Writing (Normative) ..... 125**

**C.1 Test each sentence for understandability ..... 125**

**C.2 Avoid the passive voice ..... 125**

**C.3 Be careful of the word “or” ..... 125**

**C.4 Distinguish between “support” versus “send” for Producer..... 125**

**C.5 Avoid the use of pronouns..... 125**

**C.6 Include the “why” justifying “what” statements..... 126**

**C.7 Don’t specify conformance in a Notes or Informative Annexes ..... 126**

**C.8 Use lots of examples..... 126**

**C.9 Avoid repeating the same conformance statements..... 127**

**C.10 Use “which” versus “that” properly ..... 127**

**C.11 Avoid the vague word “use” ..... 127**

**C.12 Avoid vague sentences ..... 127**

**C.13 Additional general good writing tips ..... 128**

**Annex D Change Log (Informative) ..... 129**

**D.1 Changes to make version 0.01, January 31, 2003 ..... 129**

**D.2 Changes to make version 0.02, March 2, 2003 ..... 129**

**D.3 Changes to make version 0.03, March 17, 2003 ..... 130**

**D.4 Changes to make version 0.04, March 24, 2003 ..... 130**

**D.5 Changes to make version 0.05, April 16, 2003 ..... 131**

**D.6 Changes to make version 0.06, May 23, 2003 ..... 132**

**D.7 Changes to make version 0.07, June 10, 2003 ..... 133**

**D.8 Changes to make version 0.08, August 22, 2003 ..... 133**

**D.9 Changes to make version 0.09, October 7, 2003..... 134**

**D.10 Changes to make version 0.09, October 14, 2003 ..... 134**

**D.11 Changes to make version 0.10, October 21, 2003 ..... 134**

**Table of Figures**

Figure 1-1 - Job Ticket Conceptual View .....	8
Figure 1-2 - Process and Resource Relationship .....	8
Figure 1-3 - Structure of a JDF Job Ticket.....	10
Figure 1-4 - Example of JDF and JMF workflow interactions .....	11
Figure 3-1 - Possible successor processes of each predecessor process.....	16
Figure 3-2 - Process and Resource Flow .....	18
Figure 3-3 - Combined Digital Printing Node Child Elements and Number of Occurrences .....	20
Figure 3-4 - Supported Processes and their Resources.....	21
Figure 3-5 - Number of Occurrences of Supported Resources .....	21
Figure 3-6 - Content of a Digital Printing Job Ticket .....	23
Figure 3-7 - Elements Supported in a Digital Printing Job Ticket .....	26
Figure 3-8 - Logical Producer - Consumer Diagram.....	114
Figure 3-9 - Configurations of a Controller controlling a Device .....	115
Figure 3-10 - Configurations of a Controller controlling Slave Controllers controlling Devices .....	116

# 1 Introduction

JDF (~~Job Definition Format~~) ~~was designed to~~ describes a job from conception through delivery. ~~Also the~~ job ticket can be used ~~for the to specify the~~ entire workflow. ~~JDF was also designed so that it is valid to use only the JDF that defines or~~ part of the workflow. This Interoperability and Conformance Specification (ICS) for Digital Printing defines subsets of JDF necessary to describe what a JDF producer (JDF Controller/Agent) can direct a JDF consumer (JDF Device) to do, such as printing, hole making, folding, imposition, etc.

~~The CIP4 JDF Specification [JDF] defines all valid JDF and JDF syntax can be found at [www.cip4.org](http://www.cip4.org).~~

JDF is a very ~~comprehensive flexible and extensible data~~ job ticket format, that allows for many ~~different~~ ways to ~~define specify~~ a digital print job. ~~To minimize complexity and to better guarantee interoperability between JDF producers and consumers, this ICS identifies a relatively small subset of JDF for Digital Printing. This specification will define the ICS for Digital Printing preferred method and syntax to define the JDF that is interpreted by ICS for Digital Printing software that uses a JDF job ticket, so the remainder of this specification will only define the JDF and the JDF syntax that is supported by the ICS for Digital Printing. The CIP4 JDF Specification [JDF] defines all valid JDF and JDF syntax can be found at [www.cip4.org](http://www.cip4.org).~~

~~This ICS defines~~ There are two subsets of JDF ~~covered by this specification~~ referred to as Level 1 and Level 2. Level 1 is useful for low end black and white or color desktop printing systems with no provision for in-line finishing. Level 2 is useful for color and black and white integrated digital printing systems with in-line finishing capabilities. ~~#Level 2~~ encompasses production printing IPP semantics and includes Level 1.

## 1.1 Overview of JDF

### 1.1.1 What is JDF?

JDF (~~Job Definition Format~~) is an extensible, XML-based job ticket format ~~that is~~ used by the printing industry. It has the ability to unify the pre-press, press, and post-press aspects of a printing job. ~~It and~~ provides the means to bridge the communication gap between MIS (Management Information Services) and Production Services. MIS oversees the relationship between all of the units in a workflow and is responsible for planning and controlling the job. Production Services is responsible for the operation of the job. ~~So~~ JDF provides a way to describe a job from the time that a customer places the order until the time that the finished product is placed in the customer's hands.

JDF was developed and is maintained by CIP4 (International Cooperation for Integration of Processes in Prepress, Press, and Postpress) which is an international world wide operating standards body whose purpose is to encourage computer based integration of all processes that have to be considered in the graphics arts industry.

JDF is a vendor-independent XML-based data format. ~~#JDF~~ is not a product or application, ~~it is a technology that can be incorporated into a product or application. It is~~ As an extensible format ~~easily extensible so that~~ it can ~~easily~~ be tailored to meet vendor-specific ~~product~~ requirements and ~~product~~ capabilities. JDF does not dictate that a workflow must be constructed in any pre-specified way. ~~Its~~ flexibility allows JDF to ~~be used to~~ model ~~a~~ custom solutions. JDF is equally as effective with a simple system, such as submitting a job to a printer.

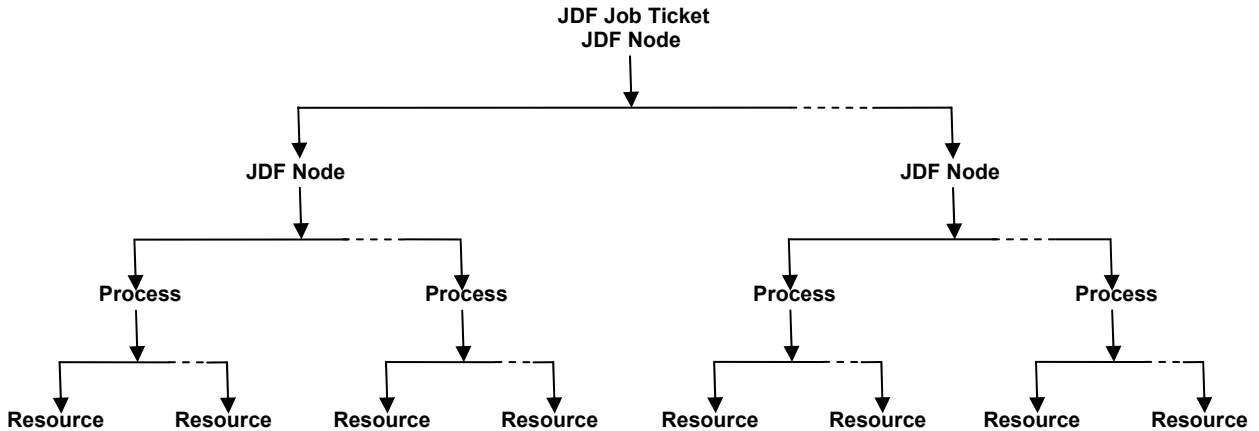
### 1.1.2 Elements in a JDF Job Ticket

There is a conceptual view of a JDF job ticket and an actual view. The conceptual view is the model on which a JDF job ticket is built where a JDF job ticket ~~is comprised of~~ one JDF node that may contain one or more other JDF nodes. ~~e~~ Each of these JDF nodes ~~is may~~ comprised of one or more processes, and resources that are inputs to and

outputs from processes. The actual or physical view of a JDF job ticket is how the document that contains the JDF XML is physically structured and what it contains.

Following is a diagram of the conceptual view of a JDF job ticket:

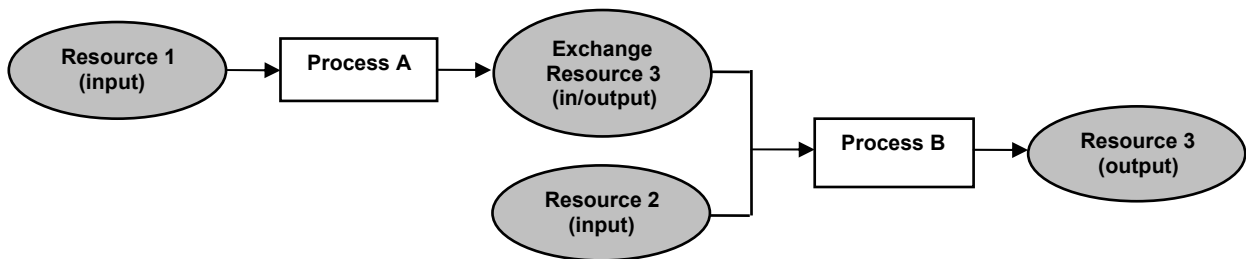
Figure 1-1 - Job Ticket Conceptual View



A JDF node is an XML element that represents an aspect of a job in terms of a process, such as stitching, folding, or trimming; in terms of a product that contributes to the end result, such as a brochure; or in terms of the combination of the two. A process is a unit of work or a step in the job workflow. A JDF node that is comprised of multiple processes is called a combined process JDF node.

Resources are inputs to and outputs from processes that provide details or information to the process or from one process to another process. For example, the `StitchingParams` resource is an input to the `Stitching` process and specifies the number of stitches and the placement of the stitches. The `Component` resource is an output of the `DigitalPrinting` process and specifies how many copies of the job to produce. The order that the processes are specified in a combined process JDF node is pertinent because it defines the order that the processes are to be executed. So if the job is to be stitched before it is folded as in the case of a saddle stitched booklet, then the `Stitching` process must be specified before the `Folding` process.

Figure 1-2 - Process and Resource Relationship



A JDF job ticket is an XML document so it requires that the version of XML and the character set encoding be specified.

```
<?xml version="1.0" encoding="UTF-8"?>
```

The attribute `encoding="UTF-8"` is the default that specifies compressed Unicode which includes ASCII as a subset.



A JDF node is specified as follows:

```
<JDF ID="node ID" Type="pre-defined JDF type or any JDF process" Types="list of JDF processes if
combined process JDF node" Status="Ready" Activation="Active"
xmlns="http://www.cip4.org/JDFSchema_1_2" xmlns:privatens="private XML namespace"
Version="1.2">
```

JDF extensibility is implemented using XML namespaces. The URI for the official JDF XML namespace is [http://www.cip4.org/JDFSchema\\_1\\_2](http://www.cip4.org/JDFSchema_1_2). Private namespaces can be declared and used to extend existing JDF resources by adding private attributes and elements. The namespaces that are to be used by the job ticket are declared in the JDF node and appear as `xmlns:privatens="private namespace"`. Namespaces that contain vendor specific attributes and/or values can be declared.

The version of JDF that is used to generate the job ticket is specified in the outside JDF node by the Version attribute. The version of JDF that is supported by the ICS for Digital Printing is 1.2.

A JDF node contains attributes and elements. Elements contain attributes and subelements. Subelements are XML elements that are contained in another element. A JDF node can contain any or all of the following elements:

- **Comment** - Human readable comment or description of the job ticket. A JDF node can contain 0 or more Comment elements.
- **AncestorPool** - Contains 1 or more Ancestor elements and is only required for a spawned job. Spawning is the process of extracting a JDF sub-node from a job and creating a new, complete JDF document. An Ancestor element reflects the family tree of the spawned job. A JDF node can contain 0 or 1 AncestorPool elements.
- **CustomerInfo** - Information about the customer who ordered the job. A JDF node can contain 0 or 1 CustomerInfo elements.
- **NodeInfo** - Information about planned scheduling and message routing. A JDF node can contain 0 or 1 NodeInfo elements.
- **ResourceLinkPool** - Links the JDF node to the Resources it uses. Also identifies the processes each Resource is an input to or output from and whether the Resource is an input or output. A JDF node can contain 0 or 1 ResourceLinkPool elements.
- **ResourcePool** - Contains 0 or more Resource elements where a Resource element is something that is consumed or produced. There are many possible Resources defined by JDF, many of which will not be supported by a specific implementation of JDF. A JDF node can contain 0 or 1 ResourcePool elements.
- **AuditPool** - Contains post-facto recorded results. A JDF node can contain 0 or 1 Audit Pool elements.
- **StatusPool** - Describes the status of a JDF node that processes partitioned Resources. A partitioned Resource is a Resource that has individual nested parts. A JDF node can contain 0 or 1 StatusPool elements.
- **JDF** - A nested JDF node. A JDF node can contain 0 or more JDF nodes.

### 1.1.3 Content of a JDF Job Ticket

The actual view of a JDF job ticket is the physical structure or content of the job ticket. Following is a sample JDF job ticket document that contains all of the elements described in the previous section:

Figure 1-3 - Structure of a JDF Job Ticket

```

<?xml version="1.0" encoding="UTF-8"?>
<JDF ID="outer node ID" Type="pre-defined JDF type or any JDF process"
  Types="list of JDF processes if combined process JDF node" Status="Ready" Activation="Active"
  xmlns="http://www.cip4.org/JDFSchema_1_2" xmlns:privatens="private XML namespace"
  Version="1.2">
  <Comment ...>
  </Comment>
  <AncestorPool >
    <Ancestor .../>
    <Ancestor .../>
  </AncestorPool>
  <CustomerInfo .../>
  <NodeInfo .../>
  <ResourceLinkPool>
    <Link for 1st Resource and processes that use it appear here />
    <Link for 2nd Resource and processes that use it appear here />
  </ResourceLinkPool>
  <ResourcePool>
    <1st Resource appears here />
    <2nd Resource appears here />
  </ResourcePool>
  <AuditPool>
    <An Audit element appears here />
  </AuditPool>
  <StatusPool>
    <Status="Ready"/>
  </StatusPool>
  <JDF ID="inner node ID" Type="pre-defined JDF type or any JDF process"
    Types="list of JDF processes in combined process JDF node" Status="Ready">
    Elements contained in this nested JDF node appear here. Any of the elements shown above in the
    outside JDF node can appear here.
  </JDF>
</JDF>

```

### Comments in a JDF Job Ticket

There are two types of comments that can appear in a JDF job ticket:

The following notation allows comments to be added inside of the job ticket. These comments aren't available outside of the job ticket. They are just comments to assist anyone reading the JDF in the job ticket.

```
<!-- The following specifies how to print duplex. -->
```

The following is a comment that can be extracted from the JDF job ticket and interpreted by the consumer of the job ticket:

```
<Comment Name="Description" A description of this job. </Comment>
```

## 1.2 Digital Printing Job Ticket

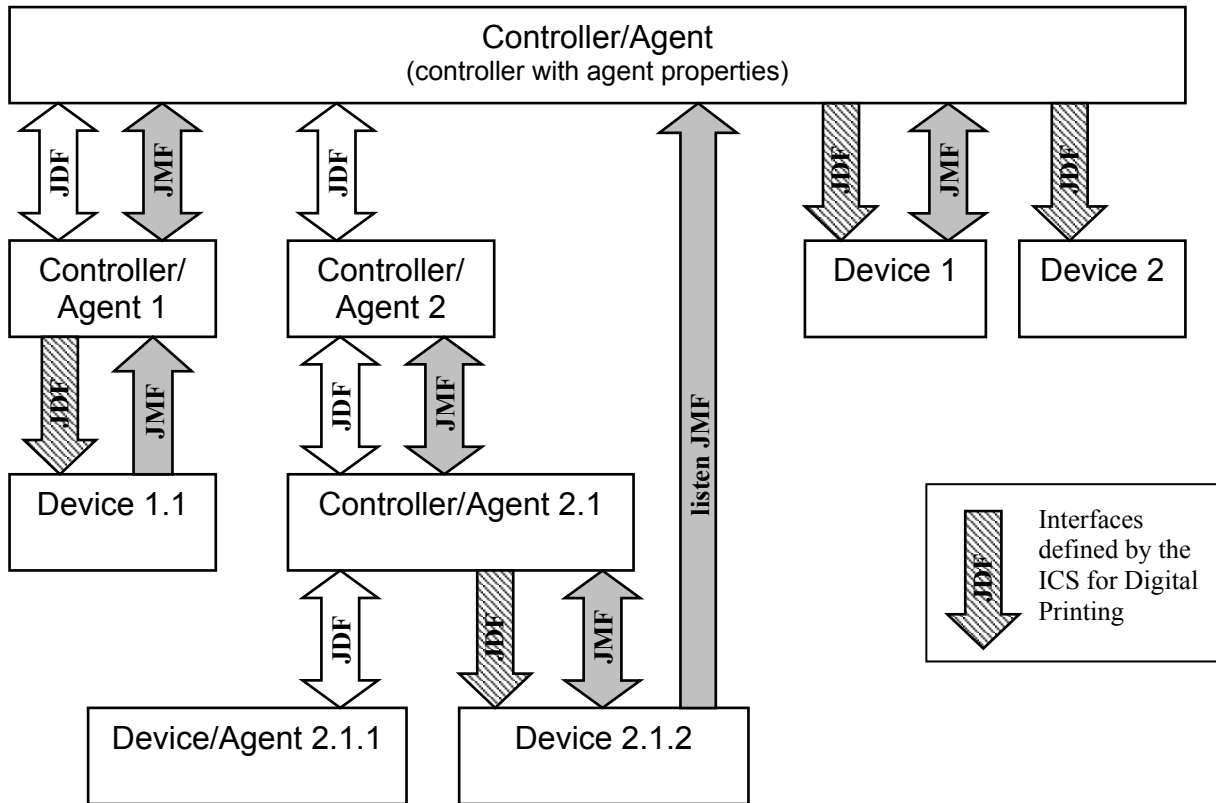
A Digital Printing Job Ticket MUST contain a combined process JDF node (Types="Combined") that contains some of the features described in section 3.2 Features Supported. The job ticket MAY contain JDF that is not described in this ICS. In this case if a Device consuming the job ticket is only interested in the features included in this ICS, then the Device MUST search for the combined process JDF node and process it.

For a Device to be in conformance with the ICS for Digital Printing, it must be able to consume and process all of the JDF that appears in the ICS for Digital Printing for a given conformance level without an error.

### 1.3 Conformance Interface, Producer, and Consumer

The following diagram, derived from the diagram that appears in the JDF Specification [JDF] section 2.1.2.6 System Interaction, shows the hierarchical structure and the interaction of the components in a JDF system.

Figure 1-4 - Example of JDF and JMF workflow interactions



The diagram was modified to show the interfaces that are conformance interfaces for the ICS for Digital Printing. These interfaces are shown by the arrows labeled "JDF" that have been cross-hatched. Note that all of the interfaces that are conformance interfaces for the ICS for Digital Printing are interfaces in which a JDF instance is passed from the output of a Controller/Agent to the input to a Device.

The Conformance Interface defined in this ICS consists of the interface between a Controller/Agent that produces a JDF instance and a Device that consumes a JDF instance and that executes a JDF *Combined* process that includes a DigitalPrinting process as specified by that JDF instance.

A Producer of this Interface is a Controller/Agent that uses a JDF instance to specify the execution of a JDF *Combined* process that includes a DigitalPrinting process by a Device.

A Consumer of this Interface is a Device that consumes JDF instances and that executes a JDF *Combined* process that includes a DigitalPrint process as specified by that JDF instance.

## 1.4 Scope

*[You MUST include the scope that is covered in the Conformance Specification, that is summarize what is covered. Sometimes it is helpful to include what is not covered as well, in case some readers might make the wrong assumptions and be disappointed. For example, if you cover Product Intent, but not Process in your ICS, state that your ICS covers Product Intent, but does not cover Process.]*

This ICS includes [the following](#):

- [A combined process node that includes the DigitalPrinting process.](#)

This ICS does not include [the following](#):

- [Intent, JMF, and Capabilities](#) because [they will be included at a later date or covered by another ICS.](#)
- [The Device is not required to write or add to a JDF job ticket \(it is not an Agent\).](#)

## 2 Terminology

- 1) A **job** is a unit of work that can be considered, at least from some point of view, by itself. Thus what is considered as a job from the point of view of a printing system may be a part of a job from the point of view of a print shop. JDF is concerned with specifying the unit of work that comprises a job and how it is to be accomplished via one or more steps. Typically, a job consists of work done on one or more files, using a set of resources, to produce output.
- 2) A **document** is a meaningful unit of information. It may be represented in various ways: for example as a file, as a part of a file, or by printed output from a job. The JDF defined in this specification specifies how input files are processed to produce output.
- 3) A **JDF job ticket** is a data structure that specifies how a job is to be done. A JDF job ticket is comprised of one or more **JDF nodes**. Each JDF node specifies a step in the performance of the job. A JDF node is represented by an XML element, and as such may have attributes and may contain subelements. Subelements are XML elements that are contained in another element.
- 4) A **process** is an entity that performs a unit of work or a step in the job workflow. Stitching processes, Folding processes, Trimming processes, and LayoutPreparation processes, are examples of processes. A JDF node specifies the unit of work done by one or more processes.

In addition to JDF nodes that specify the unit of work done by a single process, JDF also allows nodes that specify a job step that includes work done by two or more distinct processes. In this case the output resource(s) of one processes are the input resource(s) to another process. Such JDF nodes are called combined process nodes.

- 5) **Resources** are objects, such as Media, that are input to or output by processes. Resources that are input to processes may be consumed by the process (e.g., Media for the DigitalPrinting process) or merely used by them (e.g. Media for the LayoutPreparation process).
- 6) An **ICS for Digital Printing job ticket** is a JDF job ticket that conforms to the constraints of this specification. An ICS for Digital Printing job ticket contains a single JDF node, that may be a DigitalPrinting node or a combined process node that combines a DigitalPrinting process with one or more additional processes.

- 7) A **page** is a unit of output that is normally imaged on a single contiguous area on the output medium. The page is normally the unit of output to which medium selection, imposition, and a variety of other output processing options are applied.
- 8) A **sheet** is a unit of a medium, typically a sheet of paper or an area on a continuous roll of roll-fed paper, on which output is printed. Sheets may be cut and/folded after printing. A sheet with no folds typically contains 1 or 2 finished pages; a brochure with one fold inside normally contains 4 finished pages.

## 2.1 Conformance Terminology

The words “**MUST**”, “**MUST NOT**”, “**REQUIRED**”, “**SHOULD**”, “**SHOULD NOT**”, “**RECOMMENDED**”, “**MAY**”, and “**OPTIONAL**” are used in this specification in all caps to define a requirement for the Producer) and the Consumer as defined in [JDF] section 1.4.1 and is not reproduced here. In order to claim conformance to this ICS, an implementation **MUST** observe the semantics of all capitalized conformance terms in this ICS as defined in [JDF] section 1.4.1 and 1.4.2. These capitalized terms apply to conformance to this ICS only and do not apply to conformance to other specifications.

## 2.2 Other Terminology

This Conformance Specification uses the same terminology as [ICS-std], including **Conformance Level**, **Controller/Agent**, **Customer**, **Device**, **Feature**, **Interface**, **Interoperability**, **JDF Interoperability Conformance Specification (ICS)**, **Support**, and **Vendor** which is not reproduced here.

**ISSUE: Fix ICS-std definitions for the above before removing them from below.**

This Conformance Specifications defines the following additional terminology:

**Combined Digital Printing Node** - A JDF *Combined* node (value of the Types attribute is "Combined") that contains a DigitalPrinting process and that conforms to the requirements of this ICS.

**Conformance Level (Level for short)** - a set of **REQUIRED** and **OPTIONAL Features** of an **Interface** for a **Device** to support that provide a certain level of service. If an **Interface** has multiple **Conformance Levels**, each higher **Conformance Level** is a superset of all the lower levels.

**Controller/Agent** as defined by the JDF Specification [JDF] is a JDF Controller/Agent that writes and routes JDF, representing a job, to a **Device**.

**Customer** - a user who purchases a **Consumer** product or a **Controller/Agent** product from a **Vendor**. It is the Customer who is concerned about Interoperability between **Device** and **Controller/Agent** products of a particular **Interface**. Some conformance requirements of a **Controller/Agent** or **Device** include interactions between the **Controller/Agent** and the **Customer** or between the **Device** and the **Customer**. For example, in order for a **Vendor** to claim conformance to an ICS for a **Device**, a **Customer** **MUST** be able to configure a default natural language for the **Device** to use.

**Device** as defined in the JDF Specification [JDF] interprets JDF and executes instructions representing a job.

**Digital Printing ICS Subset** - The subset of JDF that can be parsed, validated, and appropriately processed by a Device that conforms to this ICS.

**Digital Printing Job Ticket** - A JDF instance, representing a job, that contains at least one Combined Digital Printing Node.

**Feature** - the parts of an **Interface** that a **Device** supports and that a **Controller/Agent** invokes in order to achieve some well-defined behavior. The semantics of the **Feature** is defined in the Interface standard with reference to [JDF]. Examples of **Features** of an (1) API, (2) a file format, or (3) a network protocol **Interface** are: (1) functions

in an Application Programming Interface (API), (2) an element in an XML file, and (3) an operation or attribute in a network protocol such as JMF, respectively.

**Interface** - a well-defined boundary between a **Controller/Agent** and a **Device** that communicate with each other with one-directional or bi-directional interaction. The specification of the **Interface** is such that a conforming **Controller/Agent** and a conforming **Device** can be implemented independently by different **Vendors** and be able to interoperate. The **Interface** specification defines the syntax and semantics of **Features**. The **Interface** specification also defines which **Features** a **Device** MUST, SHOULD, and MAY support and which **Features** a **Controller/Agent** MUST, SHOULD, and MAY supply.

**JDF Interoperability Conformance Specification (ICS)** - a CIP4 approved document that defines one or more **Interfaces** and/or **Conformance Levels** between entities in the JDF Specification [JDF] and the conformance requirements for a **Controller/Agent** and a **Device** of that **Interface** such that conforming implementations will interoperate.

**JDF Job Ticket** is used in the ICS for Digital Printing as it is defined in the JDF Specification [for representing a job](#).

**Level 1 Combined Digital Printing Node** conforms to the requirements of the ICS for Digital Printing for Conformance Level 1.

**Level 2 Combined Digital Printing Node** conforms to the requirements of the ICS for Digital Printing for Conformance Level 2.

**Level 1 Controller/Agent** conforms to the requirements of the ICS for Digital Printing for Conformance Level 1.

**Level 2 Controller/Agent** conforms to the requirements of the ICS for Digital Printing for Conformance Level 2.

**Level 1 Device** conforms to the requirements of the ICS for Digital Printing for Conformance Level 1.

**Level 2 Device** conforms to the requirements of the ICS for Digital Printing for Conformance Level 2.

NOTE: **Combined Digital Printing Node**, **Device**, and **Controller/Agent** without the Level 1 or Level 2 qualifier applies to both Level 1 and Level 2.

**Support** - a **Device** implementation **supports** a **Feature**, if the **Device** implementation responds to a request from a **Controller/Agent** for that **Feature** across the **Interface**. Such a response MUST be according to the semantics defined in the JDF and ICS specifications. Note: a **Device** MAY choose to disable a **Feature** in the **Device**, so that the **Device** implementation no longer **supports** the **Feature**, but that is a **Customer** choice, not a **Vendor** choice.

**Vendor** - a company or organization that produces a product that conforms to the **Interface** defined in an ICS as a **Controller/Agent** or as a **Device** of that **Interface**. Some conformance requirements of a **Controller/Agent** or **Device** include interactions between the **Vendor** and the **Customer**. For example, in order for a **Vendor** to claim conformance to this ICS for a **Device**, a **Vendor** MUST state the **Conformance Levels** and **Conformance Modules** supported by the **Device** implementation in any claim of conformance to this ICS.

### 3 JDF Supported by ICS for Digital Printing

This ICS specifies a subset of JDF that can be used to enable interoperability when used by a Controller/Agent to specify the execution of a JDF *Combined* node that includes a DigitalPrinting process by a Device. For Level 1 and for Level 2 this ICS defines:

~~1. a subset of JDF that can be parsed, validated, and appropriately processed by a conforming Device~~

2.1. a subset of JDF that ~~must~~**MUST** be supported by a conforming Device

2. a subset of JDF that **MAY** be supported by a conforming Device

The difference between being able to be “parsed, validated, and appropriately processed” and “supported” is that in order to parse, validate, and appropriately process a JDF element or attribute, the Device is not required to “perform the action defined in [the JDF] specification for the JDF construct when consuming a JDF instance that includes the JDF syntactic construct.” ([JDF]). The subset identified in (2) above is defined by identifying the JDF elements and attributes identified in (1) above which the Device must support.

**ISSUE: Is the above still correct? We decided that for each conformance level we will decide which features **MUST** be supported for the device to conform to that level and which features **MAY** be optionally supported but are not required to be supported. Can we require that even though the optional features **MAY** be supported by the device the device has to be able to at least “parse, validate, and appropriately process” the JDF? Or should #2 above be changed to state that it “**MAY** be supported”?**

## 3.1 Combined Digital Printing Node

A conforming Digital Printing Job Ticket **MUST** contain at least one Combined Digital Printing Node.

### 3.1.1 Settings Policy

A conforming Device **MUST** execute a Combined Digital Printing Node according to the definition for the SettingsPolicy attribute (and the BestEffortExceptions, MustHonorExceptions, and OperatorInterventionExceptions attributes) in the [JDF] when the Combined Digital Printing Node contains attributes that are defined in the [JDF], but have been omitted in the ICS for Digital Printing. This also applies to resources and attributes that are provided by a vendor as defined in a vendor namespace.

### 3.1.2 Conformance Levels

Conformance Levels 1 and 2 as defined by this ICS include support for the features defined in 3.2 Features Supported. For each of the Levels, there are features that **MUST** (Required) be supported by the Controller/Agent and Device and features that **MAY** (Optional) be supported by the Controller/Agent and Device. If a Feature **MUST** be supported, then regardless of the value of the SettingsPolicy attribute (and the BestEffortExceptions, MustHonorExceptions, and OperatorInterventionExceptions attributes), the Device **MUST** be able to consume and process the JDF that the feature contains. If the feature **MAY** be supported by the Device, then the value of the Settings Policy attribute (and the BestEffortExceptions, MustHonorExceptions, and OperatorInterventionExceptions attributes), **MUST** be honored. So, if the feature **MAY** be supported by the Device for a given conformance level and SettingsPolicy is set to BestEffort, it is valid for a conforming Device to ignore the feature.

### 3.1.3 Processes

A Combined Digital Printing Node **MUST** be a JDF *Combined* node as specified in [JDF].

#### 3.1.3.1 Required Processes

A Combined Digital Printing Node **MUST** contain the following processes:

1. LayoutPreparation
2. Imposition
3. Interpreting
4. Rendering
5. DigitalPrinting

### 3.1.3.2 Optional Finishing Processes

A Combined Digital Printing Node MAY contain 0 or more occurrences of each of the following processes:

1. Stitching
2. Folding
3. Trimming
4. HoleMaking
5. CoverApplication
6. SpineTaping

### 3.1.3.3 Other Optional Processes

A Combined Digital Printing Node MAY contain 0 or 1 occurrence of each of the following processes:

1. Trapping
2. Screening

### 3.1.3.4 Sequence of Processes

The following table shows the processes that are possible successors of other processes. The processes listed across the top of the table are the processes that are predecessors of the processes listed along the left side. An X indicates that it is valid for the Successor process to follow the Predecessor process.

Figure 3-1 - Possible successor processes of each predecessor process

		Predecessor Process														
		LayoutPreparation	Imposition	Trapping (vector)	Interpreting	Rendering	Trapping (raster)	Screening	DigitalPrinting	Stitching	Folding	Trimming	HoleMaking	CoverApplication	SpineTaping	Approval
Successor Process	LayoutPreparation															
	Imposition	X		X												
	Trapping (vector)		X		X											
	Interpreting		X	X												
	Rendering			X	X											
	Trapping (raster)						X									
	Screening						X	X								
	DigitalPrinting						X	X	X							
	Stitching									X		X	X	X	X	X
	Folding									X	X	X	X	X	X	X
	Trimming									X	X	X	X	X	X	X
	HoleMaking									X	X	X	X	X	X	X
	CoverApplication									X	X	X	X	X		
	SpineTaping									X	X	X	X	X		
Approval									X	X	X	X	X	X	X	

### 3.1.3.5 Processes and Exchange Resources

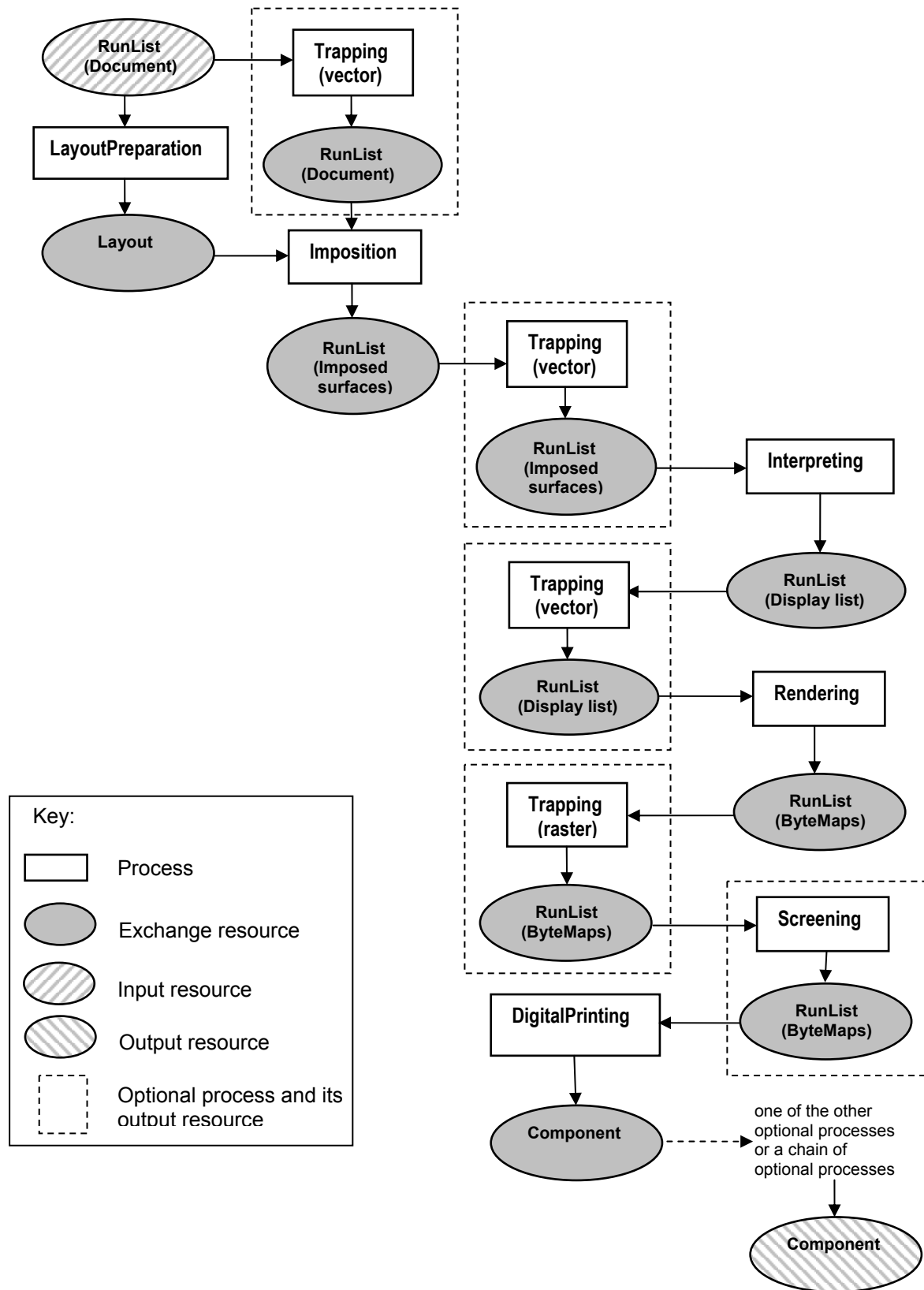
Resources represent the “things” that are produced (output) or consumed (input) by processes. Exchange Resources are resources that are produced by one process and consumed by another process within a *Combined* Node.

In a *Combined* Node a Resource that is an output from one process and an input to another (an Exchange resource) does not appear in the JDF Job Ticket.



The following diagram shows the sequence of the processes in a Combined Digital Printing Node and the flow of the content data (Exchange Resources) from one process to the next.

Figure 3-2 - Process and Resource Flow



### 3.1.4 Combined Digital Printing Node Attributes

The following table lists the attributes of a Combined Digital Printing Node. If the entry labeled “Differences between this ICS and JDF” is blank, the description of the attribute and the action defined when consuming a Combined Digital Printing Node is as specified in [JDF].

A Device that conforms to this ICS MUST support all of the attributes and values specified by this table whether the value of the attribute is specified by this Combined Digital Printing Node itself, is inherited from a predecessor node, or is assigned as a default value as specified by [JDF].

Name	Data Type	Differences between use by this ICS and [JDF]
<i>Activation ?</i>	enumeration	If present, the value MUST be <i>Active</i> or <i>Hold</i>
<i>ID</i>	ID	
<i>SettingsPolicy ?</i>	enumeration	
<i>Status</i>	enumeration	MUST be present, value MUST be <i>Ready</i>
<i>Template ?</i>	boolean	
<i>TemplateID ?</i>	string	
<i>TemplateVersion ?</i>	string	
<i>Type</i>	NMTOKEN	MUST be present, value MUST be <i>Combined</i>
<i>Types ?</i>	NMTOKENS	See 3.1.4.1 Types Attribute below.
<i>Version ?</i>	string	MUST be present, value MUST be 1.2
<i>xmlns ?</i>	URI	MUST be present, value MUST be <a href="http://www.cip4.org/JDFSchema_1_2">http://www.cip4.org/JDFSchema_1_2</a>

#### 3.1.4.1 Types Attribute

The value of the *Types* attribute for the Combined Digital Printing Node consists of the sequence of *Type* values for the included processes.

The *Types* attribute of the Combined Digital Printing Node MUST contain the *Type* values for the required processes in 3.1.3.1 Required Processes. The *Type* values for each of the required processes MUST occur in the value of the *Types* attribute exactly once.

The *Type* values for each of the required processes MUST occur in the value of the *Types* attribute in the order in which the processes are listed in 3.1.3.1 Required Processes.

NOTE: For the Required processes the *Type* values that occur in the value of the *Types* attribute are:

1. "LayoutPreparation"
2. "Imposition"
3. "Interpreting"
4. "Rendering"
5. "DigitalPrinting"

The corresponding *Types* value for the above would be "LayoutPreparation Imposition Interpreting Rendering DigitalPrinting".

If a Trapping process is present, the Trapping process *Type* value MUST occur in the *Types* value.

The Trapping process *Type* value MAY appear in one of four places in the value of the *Types* attribute:

- a. before “Imposition” (for vector-based trapping)
- b. after “Imposition” (for vector-based trapping)
- c. after “Interpreting” (for vector-based trapping)

- d. after “Rendering” (for raster-based trapping)

The Trapping process *Type* value MUST NOT appear at any other place in the value of the *Types* attribute.

If a Screening process is present, the Screening process *Type* value MUST occur in the *Types* value.

For each of the processes listed in 3.1.3.2 Optional Finishing Processes that is present, the corresponding process:

- a. The *Type* value of the process MUST occur in the *Types* value
- b. The *Type* values must occur in the order in which the processes are executed
- c. The *Type* values MUST occur after the *Type* value of the DigitalPrinting process

### 3.1.5 Elements Included in a Combined Digital Printing Node

The following table lists the child elements of a Combined Digital Printing Node as specified in [JDF], and the number of occurrences of each that can occur in a Combined Digital Printing Node.

A Combined Digital Printing Node MUST contain exactly one of the child elements for which the number of occurrences specified is “1” and MAY contain “0 or 1” or “0 or more” of the other child elements, as specified in the following table.

Figure 3-3 - Combined Digital Printing Node Child Elements and Number of Occurrences

JDF node child element	Number of occurrences
AuditPool	0 or 1
Comment	0 or more
CustomerInfo	0 or 1
NodeInfo	0 or 1
ResourcePool	1
ResourceLinkPool	1

### 3.1.6 Resources

The processes to be executed are defined in the *Types* attribute of the Combined Digital Printing Node. Resources are elements that are consumed (input) or produced (output) by processes. The Resources are defined and reside in the ResourcePool. The Resources are linked to their respective processes in the ResourceLinkPool.

#### 3.1.6.1 Resource Attributes

A Resource that resides in the ResourcePool has a set of attributes. The following table lists the attributes that are defined for all resources supported in a Combined Digital Printing Node and any differences between the use of the attribute in this ICS and [JDF].

Table 3-1 Contents of Resource element

Name	Data Type	Differences between use by this ICS and [JDF]
<i>BestEffortExceptions ?</i>	NMTOKENS	
<i>Class</i>	enumeration	The value MUST be one of the following: Consumable Implementation Parameter Quantity
<i>ID</i>	ID	
<i>MustHonorExceptions ?</i>	NMTOKENS	
<i>OperatorInterventionExcpetions ?</i>	NMTOKENS	
<i>rRefs ?</i>	IDREFS	
<i>SettingsPolicy ?</i>	enumeration	

Name	Data Type	Differences between use by this ICS and [JDF]
Status	Data Type	The value for input Resources MUST be <i>Available</i> . The value for output Resources MUST be <i>Unavailable</i> .
UpdateID ?	NMTOKEN	

### 3.1.6.2 Resources and Processes

The following table lists the resources that are required by or optional for each of the processes supported by a Combined Digital Printing Node.

Figure 3-4 - Supported Processes and their Resources

Process	Required Input Resources	Optional Input Resources	Required Output Resources
Approval	ApprovalParams	---	---
CoverApplication	CoverApplicationParams	---	---
DigitalPrinting	---	DigitalPrintingParams ColorantControl Device Media	Component
Folding	FoldingParams	---	---
HoleMaking	HoleMakingParams	---	---
Imposition	RunList (Document)	---	---
Interpreting	---	ColorantControl	---
LayoutPreparation	RunList (Document)	LayoutPreparationParams	---
Rendering	---	RenderingParams	---
Screening	---	ScreeningParams	---
SpineTaping	SpineTapingParams	---	---
Stitching	StitchingParams	---	---
Trapping (raster)	---	ColorantControl TrappingDetails	---
Trapping (vector)	---	ColorantControl TrappingDetails	---
Trimming	TrimmingParams	---	---

Because the LayoutPreparation, Imposition, Interpreting, Rendering, and DigitalPrinting processes MUST always be present in a Combined Digital Printing Node, the RunList and Component resources and their respective resource links MUST always be present as well.

Resources that are inputs to or outputs from processes MUST appear in the ResourcePool. There are also resources that appear in the ResourcePool that are not inputs to or outputs from processes, but are merely used by another resource that is in the ResourcePool. The LayoutElement is an example of such a resource. It is defined in the ResourcePool and referenced by the RunList resource, but is not an input to or output from a process.

NOTE: Because LayoutElement is not an input to or output from a process, it does not have an associated resource link in the ResourceLinkPool.

### 3.1.6.3 Resources and ResourcePool

A Combined Digital Printing Node MUST contain a ResourcePool. Resources MUST be defined and reside in the ResourcePool. Some types of Resources MUST only appear once in the ResourcePool, others MAY appear 0 or multiple times. The following table lists all possible resources and the number of times that they MUST or MAY appear in the ResourcePool.

Figure 3-5 - Number of Occurrences of Supported Resources

Resource or Element	Number of occurrences
ApprovalParams	0, <del>or</del> 1, or 2

ColorantControl	0 or more
Component	1
CoverApplicationParams	0 or 1
Device	0 or 1
DigitalPrintingParams	1
FoldingParams	0 or 1
HoleMakingParams	0 or 1
LayoutElement	0 or more
LayoutPreparationParams	1
Media	0 or more
RenderingParams	0 or 1
RunList	1
ScreeningParams	0 or 1
SpineTapingParams	0 or 1
StitchingParams	0 or 1
TrappingDetails	0 or 1
TrimmingParams	0 or 1

#### 3.1.6.4 ResourceLinks and ResourceLinkPool

Each Combined Digital Printing node MUST contain a ResourceLinkPool element. A ResourceLink element for each Resource that is an input to or an output from a process MUST appear in the ResourceLinkPool. The ResourceLink elements also define whether the resources are inputs (consumed) or outputs (produced) and the processes they are inputs to or outputs from.

Each ResourceLink element in the ResourceLinkPool MUST have a CombinedProcessIndex attribute. The CombinedProcessIndex attribute is an integer that is the zero-based index into the Combined Digital Printing Node's *Types* attribute. It identifies the processes that the resource is an input to or an output from. For example, ash shown below, the CombinedProcessIndex for RunListLink MUST have a value of "0 1", because the RunList resource is an input to both the LayoutPreparation and Imposition processes.

```
...
<RunListLink rRef="RL" Usage="Input" CombinedProcessIndex="0 1" />
...
```

The rRef attribute of each ResourceLink element references a Resource that is defined in the ResourcePool.

In the following example, the ComponentLink element is located in the ResourceLinkPool and uses the rRef attribute to reference the Component resource that is defined in the ResourcePool and has an ID of "Amount".

```
...
<ResourceLinkPool>
  <ComponentLink rRef="Amount" Usage="Output" CombinedProcessIndex="5" Amount="5" />
  ...
</ResourceLinkPool>
<ResourcePool>
  <Component ID="Amount" Class="Quantity" Status="Unavailable"
    ComponentType="FinalProduct" />
  ...
</ResourcePool>
...
```

Only those Resources that appear in the ResourceLinkPool MUST be used when the job is processed. So if there are Resources defined in the ResourcePool that should be inputs to or outputs from the processes specified in the Types attribute for the JDF node, these Resources must each have a ResourceLink in the ResourceLinkPool or they will not be used.

In Figure 3-6 - Content of a Digital Printing Job Ticket there is a list of all possible ResourceLinks that can exist in the ResourceLinkPool supported by the ICS for Digital Printing. It is possible that some of the processes supported by ICS for Digital Printing may not be included in the job ticket because the function is not necessary or are not supported by the product, so the Resources required by that process will not be defined in the ResourcePool, and the ResourceLink for that Resource will not appear in the ResourceLinkPool. For example, if the job is not to be stitched, the Stitching process is not needed. “Stitching” will not appear in the Types attribute of the JDF node, there will not be a StitchingParams resource, and a StitchingParamsLink is not needed.

### 3.1.7 Content of a Digital Printing JDF Job Ticket

The structure and the content of a Digital Printing Job Ticket is shown as follows. The specific order of the elements belonging to a Combined Digital Printing Nnode, the resources listed in the ResourcePool, and the resource links in the ResourceLinkPool is not required by JDF. Only a subset of JDF processes and JDF resources as defined in this ICS are supported in a Combined Digital Printing Node, all of which are listed below. Although all of the processes and resources that are supported by the Combined Digital Printing Node are listed below, it is unlikely that any one job ticket will contain all of them.

Figure 3-6 - Content of a Digital Printing Job Ticket

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<JDF ID="J1" Type="Combined" Types="LayoutPreparation Imposition Trapping Interpreting Rendering
  Screening DigitalPrinting Stitching Folding Trimming HoleMaking CoverApplication
  SpineTaping Approval" Status="Ready" Activation="Active"
  xmlns="http://www.cip4.org/JDFSchema_1_2" xmlns:privatens="private XML namespace"
  Version="1.2">
  <Comment ...>
  </Comment>
  <CustomerInfo ... />
  <NodeInfo ... />
  <ResourceLinkPool>
    <RunListLink rRef="ID of RunList resource" ...
      Usage="Input"
      CombinedProcessIndex="index into Types for LayoutPreparation and
        Imposition" />
    <LayoutPreparationParamsLink rRef="ID of LayoutPreparationParams resource" ...
      Usage="Input"
      CombinedProcessIndex="index into Types for LayoutPreparation" />
    <TrappingDetailsLink rRef="ID of TrappingDetails resource" ...
      Usage="Input"
      CombinedProcessIndex="index into Types for Trapping" />
    <ColorantControlLink rRef="ID of ColorantControl resource" ...
      Usage="Input"
      CombinedProcessIndex="index into Types for Trapping, Interpreting, and/or
        DigitalPrinting" />
    ... additional ColorantControlLink elements
    <RenderingParamsLink rRef="ID of RenderingParams resource" ...
      Usage="Input"
      CombinedProcessIndex="index into Types for Rendering" />
    <ScreeningParamsLink rRef="ID of ScreeningParams resource" ...
      Usage="Input"
      CombinedProcessIndex="index into Types for Screening" />
    <ComponentLink rRef="ID of Component resource" ...
      Usage="Output"
      CombinedProcessIndex="index into Types for DigitalPrinting process or last
        finishing process" />
    <DeviceLink rRef="ID of Device resource" ...
```

```

        Usage="Input"
        CombinedProcessIndex="index into Types for DigitalPrinting" />
    <MediaLink rRef="ID of Media resource" ...
        Usage="Input"
        CombinedProcessIndex="index into Types for DigitalPrinting" />
    ... additional MediaLink elements
    <DigitalPrintingParamsLink rRef="ID of DigitalPrintingParams resource" ...
        Usage="Input"
        CombinedProcessIndex="index into Types for DigitalPrinting"/>
    <StitchingParamsLink rRef="ID of StitchingParams resource" ...
        Usage="Input"
        CombinedProcessIndex="index into Types for Stitching" />
    <FoldingParamsLink rRef="ID of FoldingParams resource" ...
        Usage="Input"
        CombinedProcessIndex="index into Types for Folding" />
    <TrimmingParamsLink rRef="ID of TrimmingParams resource" ...
        Usage="Input"
        CombinedProcessIndex="index into Types for Trimming" />
    <HoleMakingParamsLink rRef="ID of HoleMakingParams resource" ...
        Usage="Input"
        CombinedProcessIndex="index into Types for HoleMaking" />
    <CoverApplicationParamsLink rRef="ID of CoverApplicationParams resource" ...
        Usage="Input"
        CombinedProcessIndex="index into Types for CoverApplication"/>
    <SpineTapingParamsLink rRef="ID of SpineTapingParams resource" ...
        Usage="Input"
        CombinedProcessIndex="index into Types for SpineTapping"/>
    <ApprovalParamsLink rRef="ID of ApprovalParams resource" ...
        Usage="Input"
        CombinedProcessIndex="index into Types for Approval" />
</ResourceLinkPool>
<ResourcePool>
    <LayoutElement ... />
    ... additional LayoutElement resources
    <RunList ... >
        <LayoutElementRef ... />
    </RunList>
    <LayoutPreparationParams ... >
        <MediaRef ... />
    </LayoutPreparationParams>
    <TrappingDetails ... />
    <ColorantControl ... />
    ... additional ColorantControl resources
    <RenderingParams ... />
    <ScreeningParams ... />
    <Component ... />
    <Device ... />
    <Media ... />
    ... additional Media resources
    <DigitalPrintingParams ... >
        <MediaRef ... />
    </DigitalPrintingParams>
    <StitchingParams ... />
    <FoldingParams ... />
    <TrimmingParams ... />
    <HoleMakingParams ... />

```



```

        <CoverApplicationParams ... />
        <SpineTapingParams ... />
        <ApprovalParams ... />
    </ResourcePool>
    <AuditPool>
        <Created ... />
        <Modified ... />
        ... additional Modified elements
    </AuditPool>
</JDF>

```

The processes listed above as the values for the JDF node's Types attribute are the processes that MUST be executed. If the work performed by a process is not to be performed, the process MUST not appear as a value for the Types attribute and resources MUST not be defined in the ResourcePool or ResourceLinkPool for that process. For example, if the job is not to be stitched, the Stitching process MUST not appear as a value for Types, and StitchingParams MUST not appear in the ResourcePool or StitchingParamsLink in the ResourceLinkPool.

The order of the processes listed in the JDF node's Types attribute is significant because it is the order that the processes MUST be executed, which is the order that the work will be performed (for example, Stitching MUST occur before Folding as shown above). The order is not significant for commutative processes. Commutative processes when executed produce the same result regardless of the order in which the processes are executed. For example, if the Types attribute contains HoleMaking and Stitching and the result is the same as if Stitching is listed before HoleMaking, then the Device can execute the Stitching process before the HoleMaking process.

Vector Trapping is an example of a process that can be executed in one of multiple places. If the Device is only able to perform the vector Trapping process in one place and the result is the same as if the Trapping process were specified in another place, then the device is free to perform vector trapping in an order other than what is specified in the Types attribute.

In a Combined Digital Printing Node the output resource of one process can be the input resource to another process; this type of resource is called an exchange resource. It is important to note that when a resource is an exchange resource, a corresponding resource link MUST not appear in the ResourcePool and ResourceLinkPool.

A resource and its corresponding resource link MUST appear in the ResourcePool and ResourceLinkPool if the processes that it is an input to or output from are listed in the JDF node's Types attribute.

### 3.1.8 Resources and Elements Supported by ICS for Digital Printing

The following table summarizes the JDF resources and elements that are supported by the ICS for Digital Printing.

The columns in the following table are:

- **Element:** The element that appears in a Digital Printing Job Ticket. The Element may be a JDF Resource.
- **XPath:** In shortened XPath notation, describes the hierarchy of the Element This is a shortened form of XPath notation since it does not show the complete path when partitioning is included.
- **Element Attributes:** Contains the attributes for the Element that may be present in a Digital Printing Job Ticket. "No attributes" signifies that the Element does not have any attributes that may be present.
- **Process(s):** Contains the process(s) that the Element, if it is a resource, is an input to or output from. If the Element is a descendant of a resource that is an input to or output from a process, the Process(s) column contains those processes. *N/A* means that the Element is not a resource or a descendant of a resource so it is not applicable that a process be specified in the column..
- **Features:** Contains the features as described in 3.2 Features Supported that use the Element.
- **Feature Attributes and Child Elements:** Contains the attributes that are used by the Feature and if the Element contains a child element, it is listed here. For example, "see Media" means that the Element in the Element column contains the Media element, so see the Media element for a list of its attributes.
- **Feature Attribute Values:** Contains the values for the Feature Attribute that are included in the Digital Printing Job Ticket. *N/A* signifies that previous column contains a child element and not an attribute.

- **Levels:** Specifies the conformance level(s) that the Feature is supported by.

~~6.3 - Alphabetize the Element Attributes column following table~~

ISSUE: Should the Attributes column be moved to the right of the Element Attributes column ?

*Figure 3-7 - Elements Supported in a Digital Printing Job Ticket*

Element	XPath	Element Attributes (M=Must be present)	Process(s)	Features	Feature Attributes and Child Elements (M=Must be present)	Feature Attribute Values (D=Default value)	Levels
Any element	<i>XPath of element</i>	BestEffortExceptions MustHonorExceptions OperatorInterventionExceptions	<i>N/A</i>	Settings Policy	BestEffortExceptions	NMTOKEN where values are attribute names	1 and 2
					MustHonorExceptions	NMTOKEN where values are attribute names	
					OperatorInterventionExceptions	NMTOKEN where values are attribute names	
Any resource	<i>XPath of resource</i>	BestEffortExceptions MustHonorExceptions OperatorInterventionExceptions SettingsPolicy	<i>process for resource</i>	Settings Policy	SettingsPolicy	Enumeration BestEffort (D) MustHonor OperatorIntervention	1 and 2
					BestEffortExceptions	NMTOKEN where values are attribute names	
					MustHonorExceptions	NMTOKEN where values are attribute names	
					OperatorInterventionExceptions	NMTOKEN where values are attribute names	
Address	JDF/ CustomerInfo/ Contact/ Address/	City Country ExtendedAddress PostalCode PostBox Region Street	<i>N/A</i>	<u>Approval</u> Contact Information Proof Print	City	string	2
					Country	string	
					ExtendedAddress	string	
					PostBox	string	
					PostalCode	string	
					Region	string	
					Street	string	
ApprovalParams	JDF/ ResourcePool/ ApprovalParams/	No attributes	Approval	<u>Approval</u> Proof Print	see ApprovalPerson	<i>N/A</i>	2
ApprovalPerson	JDF/ ResourcePool/ ApprovalParams/ ApprovalPerson/	No attributes	Approval	<u>Approval</u> Proof Print	see Contact (M)	<i>N/A</i>	2
AuditPool	JDF/ AuditPool/	No attributes	<i>N/A</i>	Job Created By	see Created	<i>N/A</i>	1 and 2
				Job Modified By	see Modified	<i>N/A</i>	?
				Job Ticket Created By	see Created	<i>N/A</i>	2
AutomatedOverprintParams	JDF/ ResourcePool/ RenderingParams/ AutomatedOverprintParams/	OverPrintBlackLineArt OverPrintBlackText	Rendering	<b>ISSUE: The Coordinate Systems working group decided to replace all ReferenceEdge attributes with Orientation attribute in a new Intermediate ComponentLink.</b> Black Overprint	OverPrintBlackLineArt	false (D) true	2
					OverPrintBlackText	false (D) true	
ColorantControl	JDF/ ResourcePool/ ColorantControl/	ProcessColorModel	DigitalPrinting	Process Color Model	ProcessColorModel	NMTOKEN DeviceCMYK DeviceGray DeviceN DeviceRGB SystemSpecified (D)	2
			Interpreting	Spot Color	see ColorantParams	<i>N/A</i>	2

Element	XPath	Element Attributes (M=Must be present)	Process(s)	Features	Feature Attributes and Child Elements (M=Must be present)	Feature Attribute Values (D=Default value)	Levels
ColorantParams	JDF/ ResourcePool/ ColorantControl/ ColorantParams/	No attributes	Interpreting	Spot Color	see SeparationParams	N/A	2
ComChannel	JDF/ CustomerInfo/ Contact/ ComChannel/	ChannelType (M) Locator (M)	N/A	Approval Contact Information Proof Print	ChannelType (M)	Enumeration Phone Email Fax WWW	2
					Locator (M)	string	
Comment	JDF/ ResourcePool/ Media/ Comment/	Name="Description" Name="OperatorText" Name= "TemplateDescription" Language	DigitalPrinting	Comment/Description of Job Media	Name="Description"	string	1 and 2
	JDF/ Comment/		N/A	Job Ticket Template Comment or Description	Name= "TemplateDescription"	string	2
	JDF/ Comment/		N/A	Message to Operator	Name="OperatorText"	string	2
	JDF/ NodeInfo/ NotificationFilter/ Comment/		N/A	Notification	Language Name="Description"	string string	?
Company	JDF/ CustomerInfo/ Contact/ Company/	OrganizationName (M)	N/A	Approval Contact Information Proof Print	OrganizationName (M)	string	2
ComponentLink	JDF/ ResourceLinkPool/ ComponentLink/	Amount	N/A	Number of Copies	Amount	integer	1 and 2
Contact	JDF/ CustomerInfo/ Contact/	ContactTypes (M)	N/A	Approval Contact Information Proof Print	ContactTypes (M)	NMTOKEN Accounting Approver Customer Delivery Owner Pickup Sender Supplier SurplusReturn ArtReturn	2
					see Address	N/A	
					see ComChannel	N/A	
					see Company	N/A	
					see Person	N/A	
CoverApplicationParams	JDF/ ResourcePool/ CoverApplicationParams/	NoOp ReferenceEdge	CoverApplication	Binding	NoOp	false (D) true	2
					ReferenceEdge	Enumeration Bottom Left	

Element	XPath	Element Attributes (M=Must be present)	Process(s)	Features	Feature Attributes and Child Elements (M=Must be present)	Feature Attribute Values (D=Default value)	Levels
						Right Top SystemSpecified (D)	
Created	JDF/ Audit/Pool/ Created/	AgentName AgentVersion <u>Author</u> <u>TimeStamp</u>	N/A	Job Ticket Created By	AgentName	string	2
					AgentVersion	string	
				Job Created By	<u>Author</u>	string	1 and 2
					<u>TimeStamp</u>	dateTime	
CustomerInfo	JDF/ CustomerInfo/	BillingCode CustomerJobName	N/A	Billing Code	BillingCode	string	2
			N/A	Contact Information	see Contact	N/A	2
			N/A	Job Name	CustomerJobName	string	1 and 2
Device	JDF/ ResourcePool/ Device/	DeviceID (M) JMFURL Manufacturer ModelNumber	DigitalPrinting	Destination or Physical Printer Requested	DeviceID (M)	string	1 and 2
					JMFURL	URL	
					Manufacturer	string	
					ModelNumber	string	
DigitalPrintingParams	JDF/ ResourcePool/ DigitalPrintingParams/	Collate <u>DirectProofAmount</u> ManualFeed NonPrintableMargin <u>Bottom</u> <u>NonPrintableMarginLeft</u> <u>NonPrintableMarginRight</u> <u>NonPrintableMarginTop</u> OutputBin PageDelivery	DigitalPrinting	Collate	Collate	Enumeration None Sheet SheetSetAndJob SystemSpecified (D)	1 and 2
			<u>DigitalPrinting</u>	Proof Print	<u>DirectProofAmount</u>	integer	2
			DigitalPrinting	Input Tray Name	ManualFeed	false true	1 and 2
					see Media	N/A	
			DigitalPrinting	Jog Offset	see Disjointing	N/A	?
			DigitalPrinting	Margins	<u>NonPrintableMarginBottom</u>	4-doubles that are values for left, bottom, right, top	?
					<u>NonPrintableMarginLeft</u>	double	
					<u>NonPrintableMarginRight</u>	double	
					<u>NonPrintableMarginTop</u>	double	
			DigitalPrinting	Media	see Media	N/A	1 and 2
DigitalPrinting	Output Bin Name	OutputBin	NMTOKEN Booklet Bottom Center FaceDown FaceUp FitMedia LargeCapacity Left Mailbox-n Middle MyMailbox Rear Right Side Stacker-n	1 and 2			

Element	XPath	Element Attributes (M=Must be present)	Process(s)	Features	Feature Attributes and Child Elements (M=Must be present)	Feature Attribute Values (D=Default value)	Levels
						SystemSpecified (D) Top Tray- <i>n</i>	
			DigitalPrinting	Page Delivery	PageDelivery	Enumeration Fanfold SameOrderFaceUp SameOrderFaceDown ReverseOrderFaceUp ReverseOrderFaceDown SystemSpecified (D)	?
Disjointing	JDF/ ResourcePool/ DigitalPrintingParams/ Disjointing/	OffsetDirection	DigitalPrinting	Jog Offset	OffsetDirection	Enumeration Alternate Left None (D) Right Straight SystemSpecified	?
FileSpec	JDF/ ResourcePool/ LayoutElement/ FileSpec/	Compression DocumentNaturalLang MimeType MimeTypeVersion URL	Imposition LayoutPreparation	Covers Force Page Insert Sheet Range of Pages to Include	URL	URL	2
				Document Compression	Compression	Enumeration Compress Deflate Gzip None (D)	2
				Document File Format	MimeType	<i>string</i>	1 and 2
					MimeTypeVersion	<i>string</i>	
				Document File Name	URL	<i>string</i>	1 and 2
Document Natural Language	DocumentNaturalLang	<i>string</i>	1 and 2				
FitPolicy	JDF/ ResourcePool/ LayoutPreparationParams/ FitPolicy/	RotatePolicy SizePolicy	LayoutPreparation	Fit Policy	SizePolicy	Enumeration Abort (D) ClipToMaxPage FitToPage ReduceToFit Tile	2
					RotatePolicy	Enumeration NoRotate (D) RotateOrthogonal RotateClockwise RotateCounterClockwise	
FoldingParams	JDF/ ResourcePool/ FoldingParams/	DescriptionType (M) FoldCatalog (M) NoOp ReferenceEdgeA	Folding	Folding	DescriptionType (M)	Enumeration FoldCatalog	2
					FoldCatalog (M)	<i>string</i> F4-1 F6-2 F6-4 F8-2	
					NoOp	false (D) true	
					ReferenceEdgeA	Enumeration Bottom	

Element	XPath	Element Attributes (M=Must be present)	Process(s)	Features	Feature Attributes and Child Elements (M=Must be present)	Feature Attribute Values (D=Default value)	Levels
						Left Right Top SystemSpecified (D)	
HoleMakingParams	JDF/ ResourcePool/ HoleMakingParams/	NoOp HoleReferenceEdge HoleType	HoleMaking	Punching and Hole Making	HoleType	Enumeration S1-generic R2-generic R3-generic R4-generic R5-generic R6-generic R7-generic SystemSpecified (D)	?
					HoleReferenceEdge	Enumeration Bottom Left (D) Right Top	
					NoOp	false (D) true	
ImageShift	JDF/ ResourcePool/ LayoutPreparationParams/ ImageShift/	PositionX PositionY ShiftBack ShiftFront	LayoutPreparation	Fit Policy	PositionX	Enumeration Center Left Right None (D)	2
					PositionY	Enumeration Bottom Center Top None (D)	
				Image Shift Back Side	ShiftBack	2 doubles that are values for x and y	2
				Image Shift Front Side	ShiftFront	2 doubles that are values for x and y	
InsertSheet	JDF/ ResourcePool/ RunList/ InsertSheet/	IsWaste SheetFormat SheetType SheetUsage	Imposition LayoutPreparation	Covers	IsWaste	false	2
					SheetFormat	Blank	
					SheetType	FillSheet	
					SheetUsage	FillForceBack	
				Force Page	SheetFormat	Blank	2
					SheetType	FillSheet	
					SheetUsage	Enumeration FillForceBack FillForceFront	
				Insert Sheet	IsWaste	false	2
					SheetType	InsertSheet	
SheetUsage	Enumeration Header Trailer						
see Sheet	N/A						
see RunList	N/A						

Element	XPath	Element Attributes (M=Must be present)	Process(s)	Features	Feature Attributes and Child Elements (M=Must be present)	Feature Attribute Values (D=Default value)	Levels
	JDF/ ResourcePool/ LayoutPreparationParams/ InsertSheet/		LayoutPreparation	Interleave Sheets	SheetFormat	Enumeration Blank Duplicate	2
					SheetType	Separator	
					SheetUsage	Enumeration Interleaved InterleavedBefore	
					see Sheet	N/A	
				Start, Separator/Slip, End Sheets	IsWaste	false true (D)	2
					SheetFormat	NMTOKEN Blank Brief Full Standard SystemSpecified (D)	
					SheetType	Enumeration JobSheet SeparatorSheet	
					SheetUsage	Enumeration Header Trailer Slip SlipCopy	
					see Sheet	N/A	
				InterpretingParams	JDF/ ResourcePool/ InterpretingParams/	PrintQuality	Interpreting
JDF	JDF/	Activation <a href="#">ICSVersions</a> SettingsPolicy Template TemplateID TemplateVersion	N/A	Hold Job	Activation	Enumeration Active Hold	2
				<a href="#">ICS Versions</a>	<a href="#">ICSVersions</a>	<a href="#">NMTOKENS</a>	
				Job Ticket Template Comment or Description	Template	false (D) true	
				see Comment		N/A	
				Job Ticket Template ID	Template	false (D) true	
				TemplateID		string	
				Job Ticket Template Version	Template	false (D) true	
TemplateVersion		string					



Element	XPath	Element Attributes (M=Must be present)	Process(s)	Features	Feature Attributes and Child Elements (M=Must be present)	Feature Attribute Values (D=Default value)	Levels	
				Settings Policy	SettingsPolicy	Enumeration BestEffort (D) MustHonor OperatorIntervention	1 and 2	
JobField	JDF/ ResourcePool/ LayoutPreparationParams/ InsertSheet/ Sheet/ Surface/ MarkObject/ JobField/	ShowList UserText	LayoutPreparation	Start, Separator/Slip, End Sheets	ShowList	UserText	2	
					UserText	<i>string</i>		
LayoutElement	JDF/ ResourcePool/ LayoutElement/	No attributes	Imposition LayoutPreparation	Covers Force Page Insert Sheet Range of Pages to Include	see FileSpec	N/A	2	
				Document Compression	see FileSpec			2
				Document File Format	see FileSpec			1 and 2
				Document File Name	see FileSpec			1 and 2
				Document Natural Language	see FileSpec			1 and 2
LayoutPreparationParams	JDF/ ResourcePool/ LayoutPreparationParams/	DocIndex NumberUp PageDistributionScheme PageOrder PresentationDirection Rotate RunIndex Sides	LayoutPreparation	Covers	DocIndex	<i>integer</i>	1 and 2	
					RunIndex	<i>integer</i>		
					Sides	Enumeration OneSidedBackFlipX OneSidedBakcFlipY OneSidedFront TwoSidedBackFlipX TwoSidedBakcFlipY		
					see Media	N/A		
				Fit Policy	see FitPolicy	N/A	2	
					see ImageShift			
				Image Shift Back Side Image Shift Front Side	see ImageShift	N/A	2	
				Input Tray Name Media	see Media	N/A	1 and 2	
				Interleave Sheets Start, Separator/Slip, End Sheets	see InsertSheet	N/A	2	
				Number Up	NumberUp	<i>2 integers that are x and y</i>	1 and 2	
				Page Distribution	PageDistributionScheme	Enumeration Perfect Saddle Sequential (D)	2	
				Page Order	PageOrder	Enumeration Booklet Reader (D)	2	
				Presentation Direction	PresentationDirection	Enumeration	2	

Element	XPath	Element Attributes (M=Must be present)	Process(s)	Features	Feature Attributes and Child Elements (M=Must be present)	Feature Attribute Values (D=Default value)	Levels
						FoldCatalog SystemSpecified (D) XYZ permutations of XYZ and xyz	
				Rotate Page	Rotate	Enumeration Rotate0 (D) Rotate90 Rotate180 Rotate270	1 and 2
				Sides	DocIndex	<i>integer</i>	1 and 2
					RunIndex	<i>integer</i>	
					Sides	Enumeration OneSidedBackFlipX OneSidedBacKFlipY OneSidedFront (D) TwoSidedBackFlipX TwoSidedBacKFlipY	
Location	JDF/ Media/ Location/	LocationName	<i>N/A</i>	Input Tray Name	LocationName	<i>string</i>	1 and 2
MarkObject	JDF/ ResourcePool/ LayoutPreparationParams/ InsertSheet/ Sheet/ Surface/ MarkObject/	No attributes	LayoutPreparation	Start, Separator/Slip, End Sheets	see JobField	<i>N/A</i>	2
Media	JDF/ ResourcePool/ Media/	BackCoating Brand Dimension DimensionName FrontCoating HoleType ImagableSide MediaColorMeasurement MediaColorName MediaColorNameDetails MediaSetCount MediaType MediaTypeDetails <a href="#">MediaUnit</a> Opacity Preprinted Recycled RecycledPercentage StockType Weight	DigitalPrinting	Covers	see Media attributes below	see Media values below	2
			DigitalPrinting	Input Tray Name	see Location	<i>N/A</i>	1 and 2
			DigitalPrinting	Media	BackCoating	Enumeration Coated Glossy HighGloss InkJet Matte Satin Semigloss None (D)	1 and 2
				Brand	<i>string</i>		
				Dimension	<i>2 doubles that are x and y</i>		
				DimensionName	NMTOKEN <i>see [IEEE-ISTO 5101.1]</i>		
				FrontCoating	<i>same as values for BackCoating</i>		
				HoleType	Enumeration None (D) <a href="#">S1-generic</a> R2-generic R3-generic R4-generic R5-generic R6-generic R7-generic		

Element	XPath	Element Attributes (M=Must be present)	Process(s)	Features	Feature Attributes and Child Elements (M=Must be present)	Feature Attribute Values (D=Default value)	Levels
						SystemSpecified	
					ImagableSide	Enumeration Back Both (D) Front Neither	
					MediaColorMeasurement	<i>LabColor</i>	
					MediaColorName	Enumeration Black Blue Brown Buff Cyan Gold Goldenrod Gray Green Ivory Magenta MultiColor Mustard NoColor Orange Pink Red Silver Turquoise Violet White Yellow All of the with a "Clear", "Dark", or "Light" prefix SystemSpecified (D)	
					MediaColorNameDetails	<i>string</i>	
					MediaSetCount	<i>integer</i>	
					MediaType	Enumeration <del>Disc</del> <del>Other</del> Paper Transparency Unknown (D)	
					MediaTypeDetails	NMTOKEN CardBoard <del>Continuous</del> ContinuousLong ContinuousShort Envelope EnvelopePlain EnvelopeWindow FullCutTabs Labels Letterhead MultiLayer	

Element	XPath	Element Attributes (M=Must be present)	Process(s)	Features	Feature Attributes and Child Elements (M=Must be present)	Feature Attribute Values (D=Default value)	Levels
						MultiPartForm <del>Paper</del> Photographic PreCutTabs Stationery TabStock <del>Tractor</del> <del>Transparency</del> <del>Unknown (D)</del>	
					<u>MediaUnit</u>	<u>Enumeration</u> <u>Continuous</u> <u>Roll</u> <u>Sheet (D)</u>	
					Opacity	Enumeration Opaque (D) Transpaerent Translucent	
					Preprinted	false (D) true	
					Recycled	false (D) true	
					RecycledPercentage	<i>double</i>	
					StockType	NMTOKEN Bristol Bond Cover Indes Newsprint Offset Tag Text <del>Unknown (D)</del>	
					Weight	<i>double</i>	
					see Comment	<i>N/A</i>	
						JDF/ ResourcePool/ RunList/ InsertSheet/ Sheet/ Media/	
	JDF/ ResourcePool/ LayoutPreparationParams/ InsertSheet/ Sheet/ Media/		LayoutPreparation	Start, Separator/Slip, End Sheets	see Media attributes above	see Media values above	2
MediaLink	JDF/ ResourceLinkPool/	Orientation	DigitalPrinting	Feed Orientation	Orientation	Enmeration Rotate0 Rotate90 Rotate180 Rotate270 Flip0 Flip90	?

Element	XPath	Element Attributes (M=Must be present)	Process(s)	Features	Feature Attributes and Child Elements (M=Must be present)	Feature Attribute Values (D=Default value)	Levels
						Flip180 Flip270	
Modified	JDF/ AuditPool/ Modified/	Author <u>TimeStamp</u>	N/A	Job Modified By	Author <u>TimeStamp</u>	string <u>dateTime</u>	?
NodeInfo	JDF/ NodeInfo/	JobPriority	N/A	Job Priority	JobPriority	integer	2
			N/A	Notification	see NotificationFilter	N/A	?
NotificationFilter	JDF/ NodeInfo/ NotificationFilter/	ChannelType Classes Locator Types	N/A	Notification	Classes ChannelType	Event Enumeration Email Fax File JMF Phone WWW None (D)	?
					Locator	string	
					Types	NMTOKEN	
					see Comment	N/A	
ObjectResolution	JDF/ RenderingParams/ ObjectResolution/	Resolution	Rendering	Destination or Physical Printer Resolution	Resolution	2 integers that are x and y	1 and 2
Person	JDF/ CustomerInfo/ Contact/ Person/	FamilyName FirstName	N/A	<u>Approval</u> Contact Information Proof Print	FirstName FamilyName	string string	2
RenderingParams	JDF/ ResourcePool/ RenderingParams/	No attributes	Rendering	<u>ISSUE: The Coordinate Systems working group decided to replace all ReferenceEdge attributes with Orientation attribute in a new Intermediate Component Ink Black Overprint</u> <u>Approval</u>	see AutomatedOverprint Params	N/A	2
				Destination or Physical Printer Resolution Proof Print	see ObjectResolution	N/A	?
RunList	JDF/ ResourcePool/ RunList/	EndOfDocument PageCopies Pages Run	LayoutPreparation	Covers Force Page Insert Sheet	Pages Run see InsertSheet see LayoutElement	IntegerRangeList string N/A N/A	2
				Document Compression	Run see LayoutElement	string N/A	2
				Document File Format Document Natural	Run see LayoutElement	string N/A	1 and 2
				Document File Name	Run	string	1 and 2

Element	XPath	Element Attributes (M=Must be present)	Process(s)	Features	Feature Attributes and Child Elements (M=Must be present)	Feature Attribute Values (D=Default value)	Levels
					EndOfDocument	false (D)	
					see LayoutElement	N/A	
				Insert Sheet	PageCopies	integer	2
				Range of Pages to Include	Pages	IntegerRangeList	2
					Run	string	
					see LayoutElement	N/A	
RunListLink	JDF/ ResourceLinkPool/ RunListLink/	DocIndex DocRunIndex RunIndex	LayoutPreparation	Range of Pages to Output	RunIndex	integer	1 and 2
					DocIndex	integer	
					DocRunIndex	integer	
ScreeningParams	JDF/ ResourcePool/ ScreeningParams/	No attributes	Screening	Screening Family	see ScreenSelector	N/A	2
ScreenSelector	JDF/ ResourcePool/ ScreeningParams/ ScreenSelector/	ScreeningFamily	Screening	Screening Family	ScreeningFamily	string where possible values are: Adobe Accuate Agfa Balanced Rational Tangent Soft-IS	2
SeparationSpec	JDF/ ResourcePool/ ColorantControl/ ColorantParams/ SeparationParams/	Name	Interpreting	Spot Color	Name	string	2
Sheet	JDF/ ResourcePool/ RunList/ InsertSheet/ Sheet/	No attributes	Imposition LayoutPreparation	Insert Sheet	see Media	N/A	2
	JDF/ ResourcePool/ LayoutPreparationsParams/ InsertSheet/ Sheet/		LayoutPreparation	Interleave Sheets	see Media		
				Start, Separator/Slip, End Sheets	see Media see Surface		2
SpineTapingParams	JDF/ ResourcePool/ SpineTapingParams/	NoOp ReferenceEdge	SpineTaping	Binding	NoOp	false (D) true	2
					ReferenceEdge	Enumeration Bottom Left Right Top SytemSpecified (D)	
StitchingParams	JDF/ ResourcePool/ StitchingParams/	Angle NoOp NumberOfStitches ReferenceEdge StitchType	Stitching	Stapling and Stitching	Angle	double	?
					NoOp	false (D) true	
					NumberOfStitches	integer	
					ReferenceEdge	Enumeration Bottom Left Right Top SytemSpecified (D)	

Element	XPath	Element Attributes (M=Must be present)	Process(s)	Features	Feature Attributes and Child Elements (M=Must be present)	Feature Attribute Values (D=Default value)	Levels
					StitchType	Enumeration Corner Saddle Side SystemSpecifed (D)	
Surface	JDF/ ResourcePool/ LayoutPreparationsParams/ InsertSheet/ Sheet/ Surface/	Side	LayoutPreparation	Start, Separator/Slip, End Sheets	Side	Enumeration Back Front	2
					see MarkObject	N/A	
TrappingDetails	JDF/ ResourcePool/ TrappingDetails/	NoOp Trapping	Trapping	Trapping	NoOp	false (D) true	2
					Trapping	false true	
TrimmingParams	JDF/ ResourcePool/ TrimmingParams/	NoOp <del>TrimmingType</del>	Trimming	Trimming	NoOp	false (D) true	2
					<del>TrimmingType</del>	<del>SystemSpecified</del>	





### 3.1.8.1 Resource References

In some cases, it is possible that more than one Resource references (uses or contains) another Resource. Instead of defining the referenced Resource twice and risking that when updating one definition, the other definition is forgotten and not updated, a Resource reference (Inter-Resource Linking) can be used. Using a Resource reference is also a way to isolate a Resource definition from the Resources that use or contain it.

The ICS for Digital Printing uses resource References instead of imbedding resources inside of other Resources.

**ISSUE: Does a conforming Device have to support both imbedded Resources and Resource references?**

In the following example the Media Resource is declared in the ResourcePool and the DigitalPrintingParams and LayoutPreparationParams Resources each use a reference to refer to it. Note the addition of the "Ref" suffix when the DigitalPrintingParams and LayoutPreparationParams Resources reference the Media Resource and the use of the rRef attribute to identify the Media Resource by its ID attribute. It is clear that the same Media resource is being referenced by both DigitalPrintingParams and LayoutPreparationParams, and any updates to the Media Resource will only have to be made in one place.

```

.....
<ResourcePool>
    ...
    <Media ID="TopTray" Class="ParameterConsumable" Status="Available" >
        <Location LocationName="top"/>
    </Media>
    <LayoutPreparationParams ID="LPP" Class="Parameter" Status="Available" >
        <MediaRef rRef="TopTray" />
    </LayoutPreparationParams>
    <DigitalPrintingParams ID="DPP" Class="Parameter" Status="Available" >
        <MediaRef rRef="TopTray" />
    </DigitalPrintingParams>
    ...
</ResourcePool>
...

```

### 3.1.8.2 Partitionable Resources

Part of a Resource may have to be defined or referenced separately from another part of the Resource. The ICS for Digital Printing uses partitioning for two purposes. One, to define the documents (files) or portions of a document that belong to the job and to which the job ticket will refer. Two, when a document in a job or certain pages of the job (or a document in the job) are to be treated differently than the rest of the job (or document). For example, specific pages of a job are to be printed on the front side of the sheet and the rest of the job is to be printed on both sides of the sheet. A partitionable Resource contains nested child Resource elements, each with the same name as the Resource.

Following is a more detailed description of the two purposes for which the ICS for Digital Printing uses the partitioning:

- 1) Partitioning **RunList** to define the documents (files) or a portion of the document in a job:
  - a) When the aggregate of the documents are to be treated as a single entity:
 

**PartIDKeys="Run"** is specified in the parent **RunList** resource and **Run="unique document identifier"** is specified in each nested child **RunList**.
  - b) When the pages of each document are to be referenced with respect to the beginning of each document:
 

The same as above and in addition **EndOfDocument="true"** is specified for each nested **RunList** child resource.

Note: If EndOfDocument="true" is not present for the last nested child RunList, even though the default for EndOfDocument is "false", the behavior is such that the last nested child RunList does include EndOfDocument set to "true". As a result, it is treated as the last document and can be referenced using a Run partition.

- 2) Partitioning RunList to define portions of a job or document as bundle items so that the individual bundle items can be referenced separately in other resources by partitioning using BundleItemIndex. Partitioning RunList so that individual bundle item are defined is necessary when insert sheets are included in the RunList and the insert sheets are to be finished with all of part of the document or job:

PartIDKeys="Run" is specified in the parent RunList resource. Run="unique bundle item identifier" and EndOfBundleItem="true" are specified in each nested child RunList.

There can also be a EndOfDocument="true" specified for each nested RunList child resource if it is necessary for other resources to partition by DocIndex.

Note: If EndOfBundleItem="true" is not present for the last nested child RunList, even though the default for EndOfBundleItem is "false", the behavior is such that the last nested child RunList does include EndOfBundleItem set to "true". As a result, it is treated as the last bundle item and can be referenced using a BundleItemIndex partition.

- 2.3) Partitioning a resource other than RunList to specify that certain pages are to be treated differently:

- a) When specific pages of the job are to be treated differently than the rest of the job:  
**PartIDKeys="RunIndex"** is specified in the parent resource and **RunIndex="list of individual pages and/or page ranges"** is specified in each nested child resource element.
- b) When specific BundleItems of the job are to be treated differently than the rest of the job:  
**PartIDKeys="BundleItemIndex"** is specified in the parent resource and **BundleItemIndex="list of individual BundleItems and/or BundleItem ranges"** is specified in each nested child resource element.
- c) When a specific document (file) or portion of a document in the job is to be treated differently than the rest of the job:  
**PartIDKeys="DocIndex"** is specified in the parent resource and **DocIndex="index of document"** is specified in each nested child resource element.
- d) When specific pages of a document (file) or portion of a document in the job are to be treated differently than the rest of the document or job:  
**PartIDKeys="DocIndex DocRunIndex"** is specified in the parent resource, **DocIndex="index of document"** is specified in each nested child resource, and **DocRunIndex="list of individual pages and/or page ranges"** is specified in each resource element that is a nested child of a nested child resource.
- e) When specific BundleItems of a document (file) or portion of a document in the job are to be treated differently than the rest of the document or job:  
**PartIDKeys="DocIndex BundleItemIndex"** is specified in the parent resource, **DocIndex="index of document"** is specified in each nested child resource, and **BundleItemIndex="list of individual BundleItems and/or BundleItem ranges"** is specified in each resource element that is a nested child of a nested child resource.  
ISSUE: It doesn't seem like it is even possible to partition by both DocIndex and BundleItemIndex because BundleItems are independent of Documents, right? You refer to a document as a whole by using DocIndex, AND refer to individual bundles using BundleItemIndex, but not both. Does this sound correct?

ISSUE: Would it be helpful if we had a table that listed each of the resources that can be partitioned and what partition key they can be partitioned by? Maybe even list the attributes that can appear in the partition

An example of 1a) and 23a) would be if specific pages and page ranges are to be printed on the front side of the sheet and the rest of the pages are to be printed on both sides of the sheet.

The RunList Resource is partitioned by Run so that two documents (files) will be included in the job. Note that EndOfDocument is not specified for the nested RunList child elements, so the pages of the job are an aggregate of the pages of all documents.

The LayoutPreparationParams Resource is the Resource that is partitioned. Note that the parent LayoutPreparationParams includes PartIDKeys="RunIndex" which specifies that partitioning is being done based on the RunIndex attribute and that each child LayoutPreparationParams includes page numbers and page ranges as values for its RunIndex attribute. Note that the pages referenced by RunIndex are the aggregate of the pages in both documents, so the page numbers of the second file starts where the page numbers of the first file ends.

```

...
<ResourcePool>
  ...
  <RunList ID="RunList" Class="Parameter" Status="Available" PartIDKeys="Run" >
    <RunList Run="1" >
      <LayoutElement ID="InputFile1" Class="Parameter" Status="Available" >
        <FileSpec URL="file:///d:/printFile1.afpps" />
      </LayoutElement>
    </RunList>
    <RunList Run="2" >
      <LayoutElement ID="InputFile2" Class="Parameter" Status="Available" >
        <FileSpec URL="file:///d:/printFile2.afpps" />
      </LayoutElement>
    </RunList>
  </RunList>
  <LayoutPreparationParams ID="LPP" Class="Parameter" Status="Available"
  PartIDKeys="RunIndex" >
    <LayoutPreparationParams RunIndex="5 6 8~12 20~1" Sides="OneSidedFront" />
    <LayoutPreparationParams RunIndex="0~4 7 13~19" Sides="TwoSidedFlipY" />
  </LayoutPreparationParams>
  ...
</ResourcePool>
...

```

An example of 1b) and 2) would be if:

- 5 pages are input to the job
- pages 0 through 2 are punched and rotated 90 degrees
- pages 0 through 1 are output to the Top bin
- pages 2 through 4 are stapled and output to the Bottom bin

The RunList Resource is partitioned by Run so that specific pages in the document (file) are bundled together and referenced as a single entity. Note that EndOfBundleItem is specified for the nested RunList child elements, so that certain pages of the document will be included in each of the bundles. Each BundleItem can be referenced by a 0-based index, where 0 is the first BundleItem, 1 is the second BundleItem, etc.

The LayoutPreparationParams Resource is also partitioned. Note that the parent LayoutPreparationParams includes PartIDKeys="RunIndex" which specifies that partitioning is being done based on the RunIndex attribute and that each child LayoutPreparationParams includes page numbers and page ranges as values for its RunIndex attribute.

The DigitalPrintingParams, HoleMakingParams, and StitchingParams Resources are partitioned by BundleItemIndex. Partitioning by BundleItemIndex allows each BundleItem to be referenced individually so that attributes can be applied to a BundleItem or to multiple BundleItems.

ISSUE: Can we make a general statement that BundleItems are only used in conjunction with ~~Finishinganything~~ when insert sheets are to be included since this is when you would want to "bundle" pages together to perform ~~finishinganything~~ on ~~thean~~ entire group of pages ~~that includes the~~ insert sheets?

```

...
<ResourcePool>
  ...
  <RunList PartIDKeys="Run">
    <RunList Run="0" Pages="0~1" EndOfBundleItem="true" >
      <LayoutElement ID="InputFile1" Class="Parameter" Status="Available" >
        <FileSpec URL="file:///d:/printFile1.afpps" />
      </LayoutElement>
    </RunList>
    <RunList Run="1" Pages="2" EndOfBundleItem="true" >
      <LayoutElement ID="InputFile1" Class="Parameter" Status="Available" >
        <FileSpec URL="file:///d:/printFile1.afpps" />
      </LayoutElement>
    </RunList>
    <RunList Run="2" Pages="3~4" EndOfBundleItem="true" >
      <LayoutElement ID="InputFile1" Class="Parameter" Status="Available" >
        <FileSpec URL="file:///d:/printFile1.afpps" />
      </LayoutElement>
    </RunList>
  </RunList>

  <LayoutPreparationParams ID="LPP" Class="Parameter" Status="Available" PartIDKeys="RunIndex"
    Rotate="Rotate0" >
    <LayoutPreparationParams RunIndex="0~2" Rotate="Rotate90" />
  </LayoutPreparationParams>

  <DigitalPrintingParams ID="DPP" Class="Parameter" Status="Available"
    PartIDKeys="BundleItemIndex" OutputBin="Top" >
    <DigitalPrintingParams BundleItemIndex="1~2" OutputBin="Bottom" >
  </DigitalPrintingParams >

  <HoleMakingParams ID="HMP" Class="Parameter" Status="Available" PartIDKeys="BundleItemIndex"
    NoOp="true" >
    <HoleMakingParams BundleItemIndex="0~1" HoleType="R3-generic"/>
  </HoleMakingParams>

  <StitchingParams ID="SP" Class="Parameter" Status="Available" PartIDKeys="BundleItemIndex"
    NoOp="true" >
    <StitchingParams BundleItemIndex="1~2" StitchType="Corner"/>
  </StitchingParams>
  ...
</ResourcePool>

```

An example of 1a) and 32c) would be if a job contains multiple documents that are to be printed where the first document is to be printed on the front side of the sheet and the second document on both sides of the sheet.

The RunList Resource is partitioned by Run so that two documents (files) can be included in the job. Note that EndOfDocument is not specified for the nested RunList child elements, so the pages of the job are an aggregate of the pages of all documents.

The LayoutPreparationParams Resources is partitioned by DocIndex, so that Sides can be specified for each document. Note that the value of DocIndex refers to the relative position of the document in the RunList.

...

```

<ResourcePool>
...
  <RunList ID="RunList" Class="Parameter" Status="Available" PartIDKeys="Run" >
    <RunList Run="1" >
      <LayoutElement ID="InputFile1" Class="Parameter" Status="Available" >
        <FileSpec URL="file:///d:/printFile1.afpps" />
      </LayoutElement>
    </RunList>
    <RunList Run="2" >
      <LayoutElement ID="InputFile2" Class="Parameter" Status="Available" >
        <FileSpec URL="file:///d:/printFile2.afpps" />
      </LayoutElement>
    </RunList>
  </RunList>
  <LayoutPreparationParams ID="LPP" Class="Parameter" Status="Available"
    PartIDKeys="DocIndex" >
    <LayoutPreparationParams DocIndex="0" Sides="OneSidedFront" />
    <LayoutPreparationParams DocIndex="1" Sides="TwoSidedFlipY" />
  </LayoutPreparationParams>
...
</ResourcePool>
...

```

An example of 1b) and 2d) would be if a job contains multiple documents that are to be printed where the first document is to be printed on the front side of the sheet and the second document on both sides of the sheet, except pages 4 and 5 of the second document are to be printed on the back side of the sheet.

The RunList Resource is partitioned by Run so that two documents (files) can be included in the job. Note that EndOfDocument="true" is specified for the first nested RunList child element, so the pages for each document can be referenced with respect to the beginning of the document. In this example page 4 and 5 for the second document are the 4th and 5th pages of the second document regardless of the number of pages in the first document.

The LayoutPreparationParams resources is partitioned by DocIndex, so that Sides can be specified for each document. Note that the value of DocIndex refers to the relative position of the document in the RunList.

```

...
<ResourcePool>
...
  <RunList ID="RunList" Class="Parameter" Status="Available" PartIDKeys="Run" >
    <RunList Run="1" EndOfDocument="true" >
      <LayoutElement ID="InputFile1" Class="Parameter" Status="Available" >
        <FileSpec URL="file:///d:/printFile1.afpps" />
      </LayoutElement>
    </RunList>
    <RunList Run="2" >
      <LayoutElement ID="InputFile2" Class="Parameter" Status="Available" >
        <FileSpec URL="file:///d:/printFile2.afpps" />
      </LayoutElement>
    </RunList>
  </RunList>
  <LayoutPreparationParams ID="LPP" Class="Parameter" Status="Available"
    PartIDKeys="DocIndex DocRunIndex" >
    <LayoutPreparationParams DocIndex="0" Sides="OneSidedFront" />
    <LayoutPreparationParams DocIndex="1" Sides="TwoSidedFlipY" />
    <LayoutPreparationParams DocRunIndex="4 5"
      Sides="OneSidedBackFlipY" />
  </LayoutPreparationParams>
...

```

```

</LayoutPreparationParams>
...
</ResourcePool>
...

```

ISSUE: Add an example for 2e) using DocIndex and BundleItemIndex here. Is it even possible to partition by DocIndex and BundleItemIndex?

An example of 1e) and 2 would be if:

- job contains 2 documents
- first document is rotated 90 degrees
- pages 0 through 2 of the first document are punched
- an insert sheet is placed before page 2 of the second document
- the insert sheet and pages 2 through 4 of the second document are stapled

The RunList Resource is partitioned by Run so that specific pages in the documents (files) are bundled together and referenced as a single entity. Note that EndOfBundleItem is specified for the nested RunList child elements, so that certain pages of the document will be included in each of the bundles. Each BundleItem can be referenced by a 0-based index, where 0 is the first BundleItem, 1 is the second BundleItem, etc.

The LayoutPreparationParams Resource is also partitioned. Note that the parent LayoutPreparationParams includes PartIDKeys="DocIndex" which specifies that partitioning is being done based on the DocIndex attribute and that each child LayoutPreparationParams includes the index of the document as a value for its DocIndex attribute.

The HoleMakingParams and StitchingParams Resources are partitioned by BundleItemIndex. Partitioning by BundleItemIndex allows each BundleItem to be referenced individually so that attributes can be applied to a BundleItem or to multiple BundleItems.

ISSUE: Can we make a general statement that BundleItems are only used in conjunction with Finishing when insert sheets are to be included since this is when you would want to "bundle" pages together to perform finishing on the entire group of pages including the insert sheets?

```

...
</ResourcePool>

...
<RunList PartIDKeys="Run">
  <RunList Run="0" Pages="0~2" EndOfBundleItem="true" >
    <LayoutElement ID="InputFile1" Class="Parameter" Status="Available" >
      <FileSpec URL="file:///d:/printFile1.ps" />
    </LayoutElement>
  </RunList>
  <RunList Run="1" Pages="3~1" EndOfDocument="true" EndOfBundleItem="true" >
    <LayoutElement ID="InputFile1" Class="Parameter" Status="Available" >
      <FileSpec URL="file:///d:/printFile1.ps" />
    </LayoutElement>
  </RunList>
  <RunList Run="2" Pages="0 1" EndOfBundleItem="true" >
    <LayoutElement ID="InputFile2" Class="Parameter" Status="Available" >
      <FileSpec URL="file:///d:/printFile2.ps" />
    </LayoutElement>
    <InsertSheet SheetType="InsertSheet" SheetUsage="Header" IsWaste="false" />
  </RunList>
  <RunList Run="3" Pages="2~4" EndOfBundleItem="true" >
    <LayoutElement ID="InputFile2" Class="Parameter" Status="Available" >
      <FileSpec URL="file:///d:/printFile2.ps" />
    </LayoutElement>
    <InsertSheet SheetType="InsertSheet" SheetUsage="Header" IsWaste="false" />

```

```

</RunList>
<RunList Run="4" Pages="5~-1" EndOfDocument="true" EndOfBundleItem="true" >
  <LayoutElement ID="InputFile2" Class="Parameter" Status="Available" >
    <FileSpec URL="file:///d:/printFile2.ps" />
  </LayoutElement>
</RunList>
</RunList>

<LayoutPreparationParams ID="LPP" Class="Parameter" Status="Available" PartIDKeys="DocIndex"
  Rotate="Rotate0" >
  <LayoutPreparationParams DocIndex="0" Rotate="Rotate90" />
</LayoutPreparationParams>

<HoleMakingParams ID="HMP" Class="Parameter" Status="Available"
  PartIDKeys="BundleItemIndex" >
  <HoleMakingParams BundleItemIndex="0" HoleType="R3-generic"/>
</HoleMakingParams>

<StitchingParams ID="SP" Class="Parameter" Status="Available" PartIDKeys="BundleItemIndex" >
  <StitchingParams BundleItemIndex="3" StitchType="Corner"/>
</StitchingParams>
...
</ResourcePool>
...

```

### 3.1.8.3 Using Resource References and Partitionable Resources

Following is an example of using both Resource references and partitioned Resources.

Resource references are used to refer to the four LayoutElement resources that identify the files that are part of the job. There are four RunList partitions, each of which contains a reference to a LayoutElement that resides in the ResourcePool.

There are three Media Resources that are referenced by a DigitalPrintingParams resource. The DigitalPrintingParams resource has been partitioned so that the four input files can be printed on media pulled from different input trays.

- The first file is printed using the media in the Top tray.
- The second file is printed using the media in the Middle tray.
- The third file is printed using the media in the Bottom tray.
- The first and last pages of the fourth file are printed using the Top tray.
- The rest of the pages of the fourth file are printed using the Middle tray.

**ISSUE:** Add use of EndOfBundleItem and BundleItemIndex to the following example. Don't really need to create BundleItems in the following example, so why do so?

```

.....
<ResourcePool>
...
  <LayoutElement ID="InputFile1" Class="Parameter" Status="Available" >
    <FileSpec URL="file:///d:/printFile1.afp" />
  </LayoutElement>
  <LayoutElement ID="InputFile2" Class="Parameter" Status="Available" >
    <FileSpec URL="file:///d:/printFile2.afp" />
  </LayoutElement>
  <LayoutElement ID="InputFile3" Class="Parameter" Status="Available" >
    <FileSpec URL="file:///d:/printFile3.afp" />

```

```

</LayoutElement>
<LayoutElement ID="InputFile4" Class="Parameter" Status="Available" >
  <FileSpec URL="file:///d:/printFile4.afp" />
</LayoutElement>
<RunList ID="RunList" Class="Parameter" Status="Available" PartIDKeys="Run" >
  <RunList Run="1" EndOfDocument="true" >
    <LayoutElementRef rRef="InputFile1" />
  </RunList>
  <RunList Run="2" EndOfDocument="true">
    <LayoutElementRef rRef="InputFile2" />
  </RunList>
  <RunList Run="3" EndOfDocument="true" >
    <LayoutElementRef rRef="InputFile3" />
  </RunList>
  <!-- Note that including EndOfDocument="true" for the following RunList
        is not required since this is the last document in the RunList. -->
  <RunList Run="4" EndOfDocument="true">
    <LayoutElementRef rRef="InputFile4" />
  </RunList>
</RunList>
<Media ID="TopTray" Class="Parameter" Status="Available">
  <Location LocationName="top" />
</Media>
<Media ID="MiddleTray" Class="Parameter" Status="Available" >
  <Location LocationName="middle" />
</Media>
<Media ID="BottomTray" Class="Parameter" Status="Available" >
  <Location LocationName="bottom" />
</Media>
<!-- Partition DigitalPrintingParams by both DocIndex and DocRunIndex so that each file in the
job and the pages in each file can be individually referenced -->
<DigitalPrintingParams ID="DPP" Class="Parameter" Status="Available"
  PartIDKeys="DocIndex DocRunIndex">
  <!-- The top tray is the default tray to be used, so it is specified in the parent
DigitalPrintingParams. It will be used by the 1st file in the job since there is not a
partition for the 1st file. -->
  <MediaRef rRef="TopTray" />
  <!-- The middle tray will be used to print the file identified by DocIndex="1", which is
the 2nd file. -->
  <DigitalPrintingParams DocIndex="1">
    <MediaRef rRef="MiddleTray" />
  </DigitalPrintingParams>
  <!-- The bottom tray will be used to print the file identified by DocIndex="2", which is
the 3rd file. -->
  <DigitalPrintingParams DocIndex="2">
    <MediaRef rRef="BottomTray" />
  </DigitalPrintingParams>
  <!--The top tray will be used to print the first and last pages of the file identified by
DocIndex="3" and the middle tray will be used to print the rest of the pages in the file.
-->
  <DigitalPrintingParams DocIndex="3">
    <DigitalPrintingParams DocRunIndex="0 -1" >
      <MediaRef rRef="TopTray" />
    </DigitalPrintingParams>
    <DigitalPrintingParams DocRunIndex="1~-2" >
      <MediaRef rRef="MiddleTray" />
    </DigitalPrintingParams>
  </DigitalPrintingParams>

```



```

        </DigitalPrintingParams>
      </DigitalPrintingParams>
    </DigitalPrintingParams>
  </ResourcePool>
  ...

```

### 3.1.8.4 Resource Update

ResourceUpdate is an abstract element that optionally contains any of the attributes and elements valid for the Resource that it resides in. Mandatory attributes and elements of resources are optional in the respective ResourceUpdate. In addition, a ResourceUpdate defined within a Resource must contain a unique *UpdateID*. The UpdateID is referenced from outside of the job ticket. The references to ResourceUpdate elements will update the current state of the resource. A ResourceUpdate is specified by adding the suffix “Update” to the Resource name. There can be one or more ResourceUpdate elements defined for each Resource that is defined in a job ticket.

ResourceUpdate elements can only be referenced from outside of the job ticket. Currently only Data streams like PPML have the ability to reference ResourceUpdate elements. When a ResourceUpdate is referenced from outside of the job ticket, e.g., from a PPML TICKET\_REF element, ONLY those elements that are explicitly specified within the ResourceUpdate will be updated.

Each resource in a conforming Digital Printing Job Ticket MAY either be partitioned or contain ResourceUpdate elements, but not both.

In the following example the DigitalPrintingParams resource initially specifies that no collation and jog offset be performed. To perform sheet collation, the outside control will reference the DigitalPrintingParamsUpdate that has the UpdateID of CollateEnabled. To turn collation off again, the outside control will reference the DigitalPrintingParamsUpdate that has the UpdateID of CollateDisabled. The same applies when the outside control turns jog offset on and off. Because PageDelivery resides in the DigitalPrintingParams resource specification, its setting is maintained when the DigitalPrintingParamsUpdate elements are referenced.

```

<DigitalPrintingParams ID="InitialDigitalPrintingParams" UpdateID="NoCollateNoJogOffset"
  Class="Parameter" Status="Available" PageDelivery="SameOrderFaceUp" Collate="None" >
  <Disjointing OffsetDirection="None" />
  <!-- Collate choices -->
  <DigitalPrintingParamsUpdate UpdateID="CollateEnabled" Collate="Sheet" />
  <DigitalPrintingParamsUpdate UpdateID="CollateDisabled" Collate="None" />
  <!-- Jog Offset choices -->
  <DigitalPrintingParamsUpdate UpdateID="JogOffsetEnabled" >
    <Disjointing OffsetDirection="Alternate" />
  </DigitalPrintingParamsUpdate>
  <DigitalPrintingParamsUpdate UpdateID="JogOffsetDisabled" >
    <Disjointing OffsetDirection="None" />
  </DigitalPrintingParamsUpdate>
</DigitalPrintingParams>

```

## 3.2 Features Supported

This section describes the features that are supported by the ICS for Digital Printing. Each feature has a short description, **Scope**, **Processes**, **Syntactic Fragment**, **Type** and **Allowed Values**.

Features defined in the following sections may be used in combination. For example, even though the Sides and Number Up features are shown as appearing in their own LayoutPreparationParams resources in the following section, they can be combined in the same LayoutPreparationParams resource as follows:

```

<LayoutPreparationParams ID="Reference ID" Class="Parameter" Status="Available"
  Sides="OneSidedFront" NumberUp="1 2" />

```

Following are descriptions of the headings that appear for each feature:

<b>Scope</b>	Identifies whether the feature defined by the JDF applies to the entire job, to specific documents in the job, or to specific pages in a document.
<b>Conformance Levels</b>	<p>Identifies the conformance levels that the feature is included in. A producer that conforms to the specified level(s) is capable of producing a job ticket that contains the feature. A consumer that conforms to the specified level(s) is capable of interpreting the feature if it appears in the job ticket. Valid values are:</p> <ul style="list-style-type: none"> <li>▪ 1 and 2</li> <li>▪ 2</li> </ul> <p>Where Level 1 is the conformance level that is useful for low-end black and white or color desktop printing systems with no provision for in-line finishing. The feature supported in Level 1 is a subset of the feature in Level 2.</p> <p>Where Level 2 is the conformance level that is useful for integrated color digital printing systems with in-line finishing capabilities. Encompasses production printing IPP semantics. Contains all features supported by Level 1.</p>
<b>Elements/Resources Used</b>	<p>Identifies the JDF Elements and Resources that are used by the Feature.</p> <p><b>ISSUE: This new heading was only added for a few Elements/Resources to see if it is value. If so then the rest of the Features need to be updated.</b></p> <p><b>6/3 - Yes, add this heading for the rest of the features.</b></p>
<b>Processes</b>	Identifies the processes to include in the JDF node's Types attribute.
<b>Syntactic Fragment</b>	<p>Shows how the JDF for the feature must appear.</p> <p>The syntactic fragment of each feature depends on the scope of the feature. Regardless of the scope of the feature, the syntactic fragment section of the feature will show the syntactic fragment as it would appear at the highest level, which is typically as Job scope. Following are examples of how the syntactic fragment of the feature must appear for each level of scope.</p> <p>For Job scope, the attributes must be specified at the root resource as shown in the following example:</p> <pre>&lt;LayoutPreparationParams ID="Reference ID" Class="Parameter" Status="Available"   Sides="OneSidedFront" /&gt;</pre> <p>For Document scope, the resource must be partitioned using DocIndex and the attributes must be specified in the partition, as shown in the following example:</p> <p>To specify two sided for all pages in the job, except the second document in the job the is to be printed on the front side:</p> <pre>&lt;LayoutPreparationParams ID="Reference ID" Class="Parameter" Status="Available"   Sides="TwoSidedFlipY" PartIDKeys="DocIndex" &gt;   &lt;LayoutPreparationParams DocIndex="1" Sides="OneSidedFront" /&gt; &lt;/LayoutPreparationParams&gt;</pre> <p>For Page scope where the job consists of one document or there are multiple documents in the job and the pages in the aggregate of the documents are to be treated as a single entity, the resource must be partitioned using RunIndex and the attributes must be specified in the partition, as shown in the following example:</p> <p>To specify two sided for all pages in the job, except pages 6 through 10 which are to be printed on the front side:</p>

	<pre>&lt;LayoutPreparationParams ID="Reference ID" Class="Parameter" Status="Available"   Sides="TwoSidedFlipY" PartIDKeys="RunIndex" &gt;   &lt;LayoutPreparationParams RunIndex="6~10" Sides="OneSidedFront" /&gt; &lt;/LayoutPreparationParams&gt;</pre> <p>For Page scope where the job consists of multiple documents and the pages in each document are to be referenced with respect to the beginning of each document, the resource must be partitioned using DocIndex and DocRunIndex and the attributes must be specified in the partition, as shown in the following example:</p> <p>To specify two sides for all pages in the job, except pages 6 through 10 in the second document in the job are to be printed on the front side:</p> <pre>&lt;LayoutPreparationParams ID="Reference ID" Class="Parameter" Status="Available"   Sides="TwoSidedFlipY" PartIDKeys="DocIndex DocRunIndex" &gt;   &lt;LayoutPreparationParams DocIndex="1" &gt;     &lt;LayoutPreparationParams       DocRunIndex="6~10" Sides="OneSidedFront" /&gt;   &lt;/LayoutPreparationParams&gt; &lt;/LayoutPreparationParams&gt;</pre> <p><b>ISSUE: How does BundleItem fit in with Page, Document, and Job scope?</b></p>
<p><b>Type</b></p>	<p>Indicates whether the feature can be specified only at job submission time and not modified after that point (Initially settable), at job submission time and can be modified after the job has been submitted (Resettable), or if it cannot be specified at job submission time or modified afterwards (Non-settable).</p> <p><b>Type</b> also indicates whether the feature can be specified with one value (single-valued) or multiple values (multi-valued).</p>
<p><b>Allowed Values</b></p>	<p>identifies the data type of each JDF attribute and the values that are allowed to be specified for each JDF attribute. The allowed values listed are all of the values that are valid for the ICS for Digital Printing JDF job ticket.</p> <p>The consumer of a ICS for Digital Printing JDF job ticket must be able to consume a job ticket that contains any of the allowed values, but the consumer does not have to support the functionality defined by the allowed values.</p>

**ISSUE:** The TSC is proposing that all SystemSpecified values be removed and omitting the attribute will result in system specified behavior. Following are the attribute where SystemSpecified is not the default value so omitting the attribute will not result in system specified behavior:

1. DigitalPrintingParams Disjointing OffsetDirection (None is the default)
2. Media HoleType (None is the default)
3. TrimmingParams TrimmingType (TrimmingType is required). May be able to deprecate TrimmingType.

**ISSUE:** Should we add a feature that requires or allows a Device to write an element to the AuditPool that indicates that the job has successfully or unsuccessfully completed. See the ProcessRun element. Does this now open up a can of worms because if the Device has to write to the AuditPool then where does the updated ticket go? Does the device then have to write to ticket somewhere?

**Should we add a feature, Job ID, that is the JDF Node JobID attribute?**

### 3.2.1 **Approval**

Whether or not an approval is required for this job. Can be used to get an approval for a job that is a proof print prior to resubmitting the job for a final print.

**Scope**

Job

**Conformance Levels**

2

**Processes**Rendering  
Approval**Syntactic Fragment**

```

<RenderingParams ID="Reference ID" Class="Parameter" Status="Available" >
  <ObjectResolution Resolution="device resolution to use" />
</RenderingParams>
<ApprovalParams ID="Reference ID" Class="Parameter" Status="Available"
  <ApprovalPerson>
    <Contact ContactType="Approver"
      <Address City="city" Country="country"
        PostBox="post office box"
        PostalCode="zip or postal code"
        Region="state or province"
        Street="street address"
        ExtendedAddress="additional address info" />
      <ComChannel ChannelType="communication type"
        Locator="communication details" />
      <Company OrganizationName="company/organization name" />
      <Person FirstName="first name"
        FamilyName="family/last name" />
    </Contact>
  </ApprovalPerson>
</ApprovalParams>

```

ISSUE- Is this how a proof print should be done? What if you want to specify that a proof print be done followed by a final print, both in the same job ticket? Should a boolean **DirectProof** or **ProofType** attribute be added to **DigitalPrintingParams**? Tom will lead a group to figure this out.

ISSUE- The Proof Print feature should be removed because there really isn't a way to print a proof, approve the proof, and then do a final print in an integrated digital printer without tying up the printer which is not reasonable (unless the job can be put into a held state). The proof print and the final print are most likely two separate jobs, although the same job ticket could be reused for the final print with a few modifications. It might be possible to have an Approval feature where a proof print is done followed by the Approval process. The Approval process merely includes the **ApprovalParams** resource that contains information about the approver. Then the separate final job can be submitted following the approval.

6/10— Decided that this feature should be renamed to "Approval" and moved to the A section.

**Type**

Resettable, single-valued

**Allowed Values**

ID

String of up to 255 characters that uniquely identifies the **RenderingParams** or **ApprovalParams** being referenced.

Resolution

Optional.

 $x \ y$ Horizontal and vertical resolution, where  $x$  and  $y$  are in DPI.

ContactType

	Always set to Approver.								
City	Optional. String of up to 255 characters that is the name of the city.								
Country	Optional. String of up to 255 characters that is the name of the country.								
PostBox	Optional. String of up to 255 characters that is the post office box (e.g P.O Box 101).								
PostalCode	Optional. String of up to 255 characters that is the zip or postal code.								
Region	Optional. String of up to 255 characters that is the state or province.								
Street	Optional. String of up to 255 characters that is the street address.								
ExtendedAddress	Optional. String of up to 255 characters that is the additional address information (e.g. Suite 245).								
ChannelType	One of the values in the following enumeration: <table> <tr> <td>Phone</td> <td>Telephone number.</td> </tr> <tr> <td>Email</td> <td>Email address.</td> </tr> <tr> <td>Fax</td> <td>Fax number.</td> </tr> <tr> <td>WWW</td> <td>WWW home page or form.</td> </tr> </table>	Phone	Telephone number.	Email	Email address.	Fax	Fax number.	WWW	WWW home page or form.
Phone	Telephone number.								
Email	Email address.								
Fax	Fax number.								
WWW	WWW home page or form.								
Locator	String of up to 255 characters that is the details about how to communicate (e.g. phone number, fax number, email address).								
OrganizationName	String of up to 255 characters that is the name of the company or organization.								
FirstName	Optional. String of up to 255 characters that is the first name of the contact.								
FamilyName	Optional. String of up to 255 characters that is the family or last name of the contact.								

### 3.2.2 Billing Code

A code to bill charges incurred.

#### Scope

Job

#### Conformance Levels

2

#### Processes

None

#### Syntactic Fragment

```
<CustomerInfo BillingCode="charge code" />
```

#### Type

Resettable, single-valued

#### Allowed Values

BillingCode  
String of up to 255 characters.

### 3.2.3 Binding

How to bind the output.

#### Scope

Page, Document, or Job

#### Conformance Levels

2

#### Processes

SpineTaping  
CoverApplication

#### Syntactic Fragment

SoftCover

To apply glue to the binding edge of a single sheet that wraps from the front around the binding edge to the back. Also referred to as Perfect binding. An example of this type of binding is a softcover book.

```
<CoverApplicationParams ID="Reference ID" Class="Parameter"
  Status="Available"
  NoOp="whether or not to bind" ReferenceEdge="which edge to bind" />
```

Tape

To apply a tape strip along the binding edge to separate front and back covers.

```
<SpineTapingParams ID="Reference ID" Class="Parameter" Status="Available"
  NoOp="whether or not to bind" ReferenceEdge="which edge to bind" />
```

#### Type

Resettable, single-valued

#### Allowed Values

ID

String of up to 255 characters that uniquely identifies the SpineTapingParams or CoverApplicationParams being referenced.

NoOp

Optional. One of the following values:

false	The specified binding is to be performed. This is the default.
true	To turn binding off. No binding is performed. All other binding attributes are ignored.

ReferenceEdge

Optional if reference edge cannot be specified or can be defaulted. One of the values in the following enumeration:

Bottom	Edge that is along the x-axis of the sheet.
Left	Edge that is along the y-axis of the sheet.
Right	Edge that is opposite of the Left edge.
Top	Edge that is opposite of the Bottom edge.
SystemSpecified	This is the default.

**ISSUE: The Coordinate Systems working group decided to replace all ReferenceEdge attributes with Orientation attribute in a new Intermediate ComponentLink**

### 3.2.4 Black Overprint

Automated selection of overprint for black text and/or graphics.

#### Scope

Job

#### Conformance Levels

2

#### Processes

Rendering

#### Syntactic Fragment

```
<RenderingParams ID="Reference ID" Class="Parameter" Status="Available" >
  <AutomatedOverprintParams OverPrintBlackText="whether to overprint black text"
    OverPrintBlackLineArt="whether to overprint black graphics" />
</RenderingParams>
```

#### Type

Resettable, single-valued

#### Allowed Values

ID

String of up to 255 characters that uniquely identifies the RenderingParams being referenced.

OverPrintBlackText

Optional. One of the following values:

false This is the default.

true

OverPrintBlackLineArt

Optional. One of the following values:

false This is the default.

true

### 3.2.5 Collate

Whether the sheets in the documents and the documents in the job should be collated when multiple copies of the job are requested.

#### Scope

Job

#### Conformance Levels

1 and 2

#### Processes

DigitalPrinting

#### Syntactic Fragment

```
<DigitalPrintingParams ID="Reference ID" Class="Parameter" Status="Available"
  Collate="whether to collate or not" />
```

#### Type

Resettable, single-valued

#### Allowed Values

ID	String of up to 255 characters that uniquely identifies the DigitalPrintingParams being referenced.	
Collate	One of the values in the following enumeration:	
	None	Do not collate.
	Sheet	Collate the sheets in the documents in the job.
	SheetSetAndJob	Collate the sheets in the documents in the job and the documents in the job.
	SystemSpecified	Collate as specified by the system. This is the default.

The following values for Collate are included in the JDF Specification Release 1.2, but have intentionally been excluded in the ICS for Digital Printing:

SheetAndJob

Example: A job contains 2 documents, A and B, each have 2 sheets, A1, A2, and B1, B2. The number of job copies requested is 3.

If Collate=None, the sheet order will be: A1A1A1 A2A2A2 B1B1B1 B2B2B2

If Collate=Sheet the sheet order will be: A1A2 A1A2 A1A2 B1B2 B1B2 B1B2

If Collate=SheetSetAndJob, the sheet order will be: A1A2B1B2 A1A2B1B2 A1A2B1B2

### 3.2.6 Comment/Description of Job

A comment that describes the job and will be carried along with the job. This information is about the job not the job ticket. Use the Job Ticket Template Comment or Description to include a description of the job ticket template.

#### Scope

Job

#### Conformance Levels

~~1 and 2~~

#### Processes

None

#### Syntactic Fragment

<Comment Name="Description"> *job comment or description text* </Comment>

#### Type

Resettable, single-valued

#### Allowed Values

Name

Always set to Description.

*job comment or description text*

String of up to 1023 characters

### 3.2.7 Contact Information

Contact information for the job.

#### Scope

Job



**Conformance Levels**

2

**Processes**

None

**Syntactic Fragment**

```

<CustomerInfo
  <Contact ContactTypes="classification of contact" >
    <Address City="city" Country="country" PostBox="post office box"
      PostalCode="zip or postal code" Region="state or province"
      Street="street address" ExtendedAddress="additional address
        info"
    />
    <ComChannel ChannelType="communication type"
      Locator="communication details" />
    <Company OrganizationName="company/organization name" />
    <Person FirstName="first name" FamilyName="family/last name" />
  </Contact>

```

**Type**

Resettable, multi-valued

**Allowed Values**

ContactTypes

String (NMTOKEN) of up to 255 characters. Some typical values are as follows:

Administator
Accounting
Approver
Customer
Delivery
Owner
Pickup
Sender
Supplier
SurplusReturn
ArtReturn

**ISSUE: Get with e-Commerce group about the exact definitions of the above value. 6/3 - Tom will add this issue to the proposed JDF 1.2. Tom and Steve S. will work on clarifying these definitions.**

City

Optional. String of up to 255 characters that is the name of the city.

Country

Optional. String of up to 255 characters that is the name of the country.

PostBox

Optional. String of up to 255 characters that is the post office box (e.g P.O Box 101).

PostalCode

Optional. String of up to 255 characters that is the zip or postal code.

Region

Optional. String of up to 255 characters that is the state or province.

Street

Optional. String of up to 255 characters that is the street address.

ExtendedAddress

Optional. String of up to 255 characters that is the additional address information (e.g. Suite 245).

**ChannelType**

One of the values in the following enumeration:

- Phone Telephone number.
- Email Email address.
- Fax Fax number.
- WWW WWW home page or form.

The following values for ChannelType are included in the JDF Specification Release 1.2, but have intentionally been excluded in the ICS for Digital Printing:

JMF

**Locator**

String of up to 255 characters that is the details about how to communicate (e.g. phone number, fax number, email address).

**OrganizationName**

String of up to 255 characters that is the name of the company or organization.

**FirstName**

Optional. String of up to 255 characters that is the first name of the contact.

**FamilyName**

Optional. String of up to 255 characters that is the family or last name of the contact.

**3.2.8 Covers**

There are two ways to include covers with a job. One, the content that is to be imaged on the covers is not included in the job, the covers are either blank or preprinted. Two, the content that is to be imaged on the covers is included in the job. For the first case, see Insert Sheet on 78 page. For the second case, the following description explains how different media can be specified for the front cover, back cover, and the document body.

Print front and/or back covers with the job, where the first one or two pages and/or the last one of two pages of the job contains the content of the front and/or back covers.

**Scope**

Job

**Conformance Levels**

2

**Processes**

LayoutPreparation

**Syntactic Fragment**

```

<Media ID="Front cover media reference ID" Class="Consumable" Status="Available" />
<Media ID="Back cover media reference ID" Class="Consumable" Status="Available" />
<Media ID="Document body media reference ID" Class="Consumable"
  Status="Available" />
<LayoutPreparationParams ID="LayoutPreparationParams reference ID" Class="Parameter"
  Status="Available" PartIDKeys="DocIndex RunIndex | DocRunIndex" >
  <LayoutPreparationsParams
    DocIndex="index of document or part of document"
    <!-- Sides and Media for front cover -->
    <LayoutPreparationParams RunIndex | DocRunIndex="0" | "0 1"
      Sides="sides to print front cover on" >
      <MediaRef rRef="Front cover media reference ID" />
    </LayoutPreparationParams>
  </LayoutPreparationParams>
  <!-- Sides and Media for document body -->
  </LayoutPreparationParams

```

```

        RunIndex | DocRunIndex="1~2" | "1~3" | "2~2" | "2~3"
        Sides="sides to print document body on" >
        <Media rRef="Document body media reference ID" />
    </LayoutPreparationParams>
    <!-- Sides and Media for back cover -->
    <LayoutPreparationParams RunIndex | DocRunIndex="-1" | "-2 -1"
        Sides="sides to print back cover on" >
        <MediaRef rRef="Back cover media reference ID" />
    </LayoutPreparationParams>
</LayoutPreparationParams>

<!-- In the case where the media for the front and back covers is the same as the media for the
body of the document and the entire document (covers and body) is to be printed two
sided, the body of the document could end on the front side of a sheet, which would mean
that the front side of the back cover would print on the back side of the last sheet of the
body of the document. The following will force the back side of the last sheet of the body
of the document to be blank if the body of the document ends on the front side of the
sheet. It conditionally inserts a fill sheet. If the body of the document ends on the back
side of the sheet, then a fill sheet is not inserted because the front side of the back cover
will already be printed on the next sheet. -->
<LayoutElement ID="LayoutElement reference ID" Class="Parameter" Status="Available" >
    <FileSpec URL="document file name" />
</LayoutElement>
<RunList ID="RunList reference ID" Class="Parameter" Status="Available"
    PartIDKeys="Run">
    <!-- The following will insert a sheet before the third to the last page if necessary. -->
    <RunList Run="document's page range Run value" Pages="0~3">
        <LayoutElementRef rRef="LayoutElement reference ID" />
        <InsertSheet SheetType="FillSheet" SheetUsage="FillForceBack"
            SheetFormat="Blank" IsWaste="false" />
    </RunList>
    <!-- Include the last two pages which will be the back cover. -->
    <RunList Run="document's page range Run value" Pages="-2 -1">
        <LayoutElementRef rRef="LayoutElement reference ID" />
    </RunList>
</RunList>

```

**Type**

Resettable, single-valued

**Allowed Values**

ID and rRef

Strings of up to 255 characters that uniquely identify the Media, LayoutPreparationParams, and RunList being referenced.

Run

String of up to 255 characters that is a reference to the document's Run value or a subset of pages in a document.

DocIndex

Optional. Index of document being referenced. Specify DocIndex and DocRunIndex to reference the pages in an individual document.

RunIndex or DocRunIndex

Specify RunIndex to reference the pages in the job and DocRunIndex to reference the pages in an individual document.

0 First page. Use for first RunIndex/DocRunIndex for one-sided front cover.

-1	Last page. Use for third RunIndex/DocRunIndex for one-sided back cover.
0 1	First two pages. Use for first RunIndex/DocRunIndex for two-sided front cover.
-2 -1	Last two pages. Use for third RunIndex/DocRunIndex for two-sided back cover.
1~2	Second through second to the last pages. Use for second RunIndex/DocRunIndex for document body when front and back covers are one-sided.
1~3	Second through third to the last pages. Use for second RunIndex/DocRunIndex for document body when front cover is one-sided and back cover is two-sided.
2~2	Third through second to the last pages. Use for second RunIndex/DocRunIndex for document body when front cover is two-sided and back cover is one-sided.
2~3	Third through third to the last pages. Use for second RunIndex/DocRunIndex for document body when front and back covers are two-sided.

## Sides

One of the values in the following enumeration:

OneSidedBackFlipX	Image on back side of the media where the page content is placed such that binding can occur along the x-axis. This is equivalent to tumble.
OneSidedBackFlipY	Image on back side of the media where the pagecontent is placed such that binding can occur along the y-axis.
OneSidedFront	Image on front side of the media. This is the default.
TwoSidedFlipX	Image on front and back sides of media where the page content is placed such that binding can occur along the x-axis. This is equivalent to tumble.
TwoSidedFlipY	Image on front and back sides of media where the page content is placed such that binding can occur along the y-axis.

## URL

Location of the document. See Document File Name on page 65 for usage notes.

## Pages

Always “0~3” for the first occurrence of Pages and and “-2 -1” for the second.

## SheetType

Always set to FillSheet.

## SheetUsage

Always set to FillForceBack.

## SheetFormat

Always set to Blank.

## IsWaste

Always set to false.

Note: Valid combinations of Pages and Sides are:

- 0 and OneSidedFront to print the first page on the front side of the front cover.
- 0 and OneSidedBackFlipX or OneSidedBackFlipY to print the first page on the back side of the front cover.
- -1 and OneSidedBackFlipX or OneSidedBackFlipY to print the last page on the back side of the back cover.
- -1 and OneSidedFront to print last page on the front side of the back cover.
- 0 1 and TwoSidedFlipX or TwoSidedFlipY to print the first two pages on the front and back sides of the front cover.

- -2 -1 and TwoSidedFlipX or TwoSidedFlipY to print the last two pages on the front and back sides of the back cover.
- 1~-2, 1~-3, 2~-2, 2~-3, and any value of Sides to print the body of the document as specified by Sides.

### 3.2.9 Destination or Physical Printer Requested

Printer or destination device ID which is the name of the physical device that will process the job, document, or range of pages. The model number can also be specified.

#### Scope

Page, Document, or Job

#### Conformance Levels

1 and 2

#### Processes

DigitalPrinting

#### Syntactic Fragment

```
<Device ID="Reference ID" Class="Implementation" Status="Available" >
  DeviceID="name or ID of physical device"
  JMFURL="URL of the device port"
  Manufacturer="manufacturer of physical device"
  ModelNumber="model number of physical device" />
</Device>
```

**ISSUE: Is this the correct usage of JMFURL? Can JMFURL be used to identify the URL of the device even if the device is not accepting JMF messages, but it is just being used in the same way that the printer-uri attribute?**

#### Type

Resettable, single-valued

#### Allowed Values

ID

String of up to 255 characters that uniquely identifies the Device being referenced.

DeviceID

String of up to 255 characters.

JMFURL

Optional. URL of the device port that will accept JMF messages.

Manufacturer

Optional. String of up to 255 characters.

ModelNumber

Optional. String of up to 255 characters.

### 3.2.10 Destination or Physical Printer Resolution

Resolution that the destination or physical printer is to use in the horizontal (cross-feed) and vertical (feed) directions in units of DPI.

#### Scope

Page, Document, or Job

#### Conformance Levels

1 and 2

**Processes**

RenderingParams

**Syntactic Fragment**

```
<RenderingParams ID="Reference ID" Class="Parameter" Status="Available" >
  <ObjectResolution Resolution="device resolution to use" />
</RenderingParams>
```

**Type**

Resettable, single-valued

**Allowed Values**

ID

String of up to 255 characters that uniquely identifies the RenderingParams being referenced.

Resolution

 $x$   $y$  Horizontal and vertical resolution where  $x$  and  $y$  are in DPI.**3.2.11 Document Compression**

Compression algorithm to use to compress the document data.

**Scope**

Document

**Conformance Levels**

2

**Processes**

LayoutPreparation

**Syntactic Fragment**

```
<LayoutElement ID="LayoutElement reference ID" Class="Parameter"
  Status="Available">
  <FileSpec Compression="compression algorithm to use" />
</LayoutElement>
<RunList ID="RunList reference ID" Class="Parameter" Status="Available">
  <RunList Run="document's Run value">
    <LayoutElementRef rRef="LayoutElement reference ID" />
  </RunList>
</RunList>
```

**ISSUE:** The TSC has requested that partitioning not be used in the above case when there is only one file in the RunList. So the above RunList would appear as follows. Is this correct? Also see the Document File Format, Document File Name, and Document Natural Language features.

For jobs that contain a single document (file):

```
<LayoutElement ID="LayoutElement reference ID" Class="Parameter"
  Status="Available" >
  <FileSpec Compression="compression algorithm to use" />
</LayoutElement>
<RunList ID="RunList reference ID" Class="Parameter" Status="Available" >
  <LayoutElementRef rRef="LayoutElement reference ID" />
</RunList>
```

For jobs that contain multiple documents (files):

```
<LayoutElement ID="LayoutElement 1 reference ID" Class="Parameter"
  Status="Available" >
  <FileSpec Compression="compression algorithm to use" /></LayoutElement>
<LayoutElement ID="LayoutElement 2 reference ID" Class="Parameter"
  Status="Available" >
  <FileSpec Compression="compression algorithm to use" />
</LayoutElement>
<RunList ID="RunList reference ID" Class="Parameter" Status="Available"
  PartIDKeys="Run" >
  <RunList Run="document's Run value" >
    <LayoutElementRef rRef="LayoutElement 1 reference ID" />
  </RunList>
  <RunList Run="document's Run value" >
    <LayoutElementRef rRef="LayoutElement 2 reference ID" />
  </RunList>
</RunList>
```

#### Type

Resettable, single-valued

#### Allowed Values

ID

String of up to 255 characters that uniquely identifies the LayoutElement being referenced.

Run

String of up to 255 characters that is the reference to the document's Run value.

Compression

One of the values in the following enumeration:

None	Do not compress. This is the default.
Deflate	Use ZIP public domain compression (RFC 1951).
Gzip	Use GNU zip compression technology (RFC 1952).
Compress	Use UNIX compression (RFC 1977).

### 3.2.12 Document File Format

Identifies the format of the data in the document in a job.

#### Scope

Document

#### Conformance Levels

1 and 2

#### Processes

LayoutPreparation

#### Syntactic Fragment

```
<LayoutElement ID="LayoutElement reference ID" Class="Parameter" Status="Available">
  <FileSpec MimeType="data file format" MimeTypeVersion="version or level of
  data file format" />
</LayoutElement>
<RunList ID="RunList reference ID" Class="Parameter" Status="Available"
  PartIDKeys="Run">
  <RunList Run="document's Run value" >
    <LayoutElementRef rRef="LayoutElement reference ID" />
  </RunList>
</RunList>
```

```

</RunList>
</RunList>
For jobs that contain a single document (file):
<LayoutElement ID="LayoutElement reference ID" Class="Parameter"
  Status="Available" >
  <FileSpec MimeType="data file format" MimeTypeVersion="version or level of
  data file format" />
</LayoutElement>
<RunList ID="RunList reference ID" Class="Parameter" Status="Available" >
  <LayoutElementRef rRef="LayoutElement reference ID" />
</RunList>

```

```

For jobs that contain multiple documents (files):
<LayoutElement ID="LayoutElement 1 reference ID" Class="Parameter"
  Status="Available" >
  <FileSpec MimeType="data file format" MimeTypeVersion="version or level of
  data file format" />
</LayoutElement>
<LayoutElement ID="LayoutElement 2 reference ID" Class="Parameter"
  Status="Available" >
  <FileSpec MimeType="data file format" MimeTypeVersion="version or level of
  data file format" />
</LayoutElement>
<RunList ID="RunList reference ID" Class="Parameter" Status="Available"
  PartIDKeys="Run" >
  <RunList Run="document's Run value" >
    <LayoutElementRef rRef="LayoutElement 1 reference ID" />
  </RunList>
  <RunList Run="document's Run value" >
    <LayoutElementRef rRef="LayoutElement 2 reference ID" />
  </RunList>
</RunList>

```

**Type**

Resettable, single-valued

**Allowed Values**

ID and rRef

Strings of up to 255 characters that uniquely identify the LayoutElement and RunList being referenced.

Run

String of up to 255 characters that is the reference to the document's Run value.

MimeType

String of up to 255 characters.

The following values must be supported:

application/pdf

Following are examples of other values that can be supported:

application/postscript

application/octet-stream (supported by devices that are capable of auto-sensing the format)

application/vnd.hp-PCL

image/jpg

image/tiff

text/plain

MimeTypeVersion

String of up to 255 characters.



Use:

“PS/1”, “PS/2”, “PS/3” for *MimeType* = “application/postscript”  
 “PCL/3”, “PCL/4”, “PCL/5”, “PCL/5e” for *MimeType* = “application/vnd.hp-PCL”  
 “PDF/1.3”, “PDF/1.4”, “PDF/X-1a:2001” for *MimeType* = “application/pdf”  
 “PDF/is-1.0” for PDF Image Streamable being developed by the Printer Working Group (PWG)

For file formats not in the Printer MIB, the prefix is the common acronym for the format:  
 “DCS/2.0” for *MimeType* = “DCS”  
 “MSWORD/5.0”, “MSWORD/6.0”, “MSWORD/2000”, “MSWORD/XP” for *MimeType* = “application/msword”  
 “???” for *MimeType* = “ICC Profile” **ISSUE (Ann): What versions for ICC Profiles?**  
 “TIFF-IT/FP:1998”, “TIFF-IT/CT:1998”, “TIFF-IT/LW:1998” for *MimeType* = “TIFF/IT”

### 3.2.13 Document File Name

Name of the document (file) in a job.

#### Scope

Document

#### Conformance Levels

1 and 2

#### Processes

LayoutPreparation

#### Syntactic Fragment

For jobs that contain a single document (file):

```
<LayoutElement ID="LayoutElement reference ID" Class="Parameter"
  Status="Available" >
  <FileSpec URL="document file name" />
</LayoutElement>
<RunList ID="RunList reference ID" Class="Parameter" Status="Available"
  PartIDKeys="Run">
  <RunList Run="document's Run value">
  <LayoutElementRef rRef="LayoutElement reference ID" />
  </RunList>
</RunList>
```

For jobs that contain multiple documents (files):

```
<LayoutElement ID="LayoutElement 1 reference ID" Class="Parameter"
  Status="Available" >
  <FileSpec URL="document 1 file name" />
</LayoutElement>
<LayoutElement ID="LayoutElement 2 reference ID" Class="Parameter"
  Status="Available" >
  <FileSpec URL="document 2 file name" />
</LayoutElement>
<RunList ID="RunList reference ID" Class="Parameter" Status="Available"
  PartIDKeys="Run" >
  <RunList Run="document's Run value" EndOfDocument=
    "whether of not document is to be treated separately" >
  <LayoutElementRef rRef="LayoutElement 1 reference ID" />
```

```

    </RunList>
    <RunList Run="document's Run value" EndOfDocument=
        "whether of not document is to be treated separately" >
        <LayoutElementRef="LayoutElement 2 reference ID" />
    </RunList>
</RunList>

```

**Type**

Initially settable, multi-valued

**Allowed Values**

ID and rRef

Strings of up to 255 characters that uniquely identify the LayoutElement and RunList being referenced.

EndOfDocument

One of the following values:

false            The pages in the files are to be concatenated, so that the job is treated as a single entity. This is the default.

true              Each file is treated as a separate entity so the pages in each file can be referenced individually starting at page 0.

Note: All files in the RunList must have EndOfDocument set to the same value.

Run

String of up to 255 characters that is the reference to the document's Run value.

URL

Optional. Location of the document in the form as specified in [JDF\_URL].

If FileSpec is specified without URL then the JDF job ticket will be used for all of the document(s) that are grouped and sent along with the job ticket. The RunListLink must specify a PipeProtocol, as follows:

```

...
<ResourceLinkPool>
  <RunListLink rRef="RunList reference ID" Usage="Input"
    CombinedProcessIndex="0 1"
    PipeProtocol="AccompanyingFiles" />
...
</ResourceLinkPool>
<ResourcePool>
  <LayoutElement ID="LayoutElement reference ID" Class="Parameter"
    Status="Available" >
    <FileSpec />
  </LayoutElement>
  ...
</ResourcePool>
...

```

ISSUE: Can the above PipeProtocol be removed since now the JDF Spec says the following for URL:

If URL is not specified in an output resource, the system specified location will be assumed. This value must be updated as soon as the output resource is available. For example, an instruction for a digital delivery Device to compress the files may specify the output RunList with the *Compression* attribute without the *URL* attribute.

So, can the URL just be left off and never updated so that "the system specified location will be assume"?

6/3 - Tom added an ISSUE to JDF 1.2.

### 3.2.14 Document Natural Language

The natural language of the document (file) in a job.

#### Scope

Document

#### Conformance Levels

1 and 2

#### Processes

LayoutPreparation

#### Syntactic Fragment

```
<LayoutElement ID="LayoutElement reference ID" Class="Parameter" Status="Available" >
  <FileSpec DocumentNaturalLanguage="data format" />
</LayoutElement>
<RunList ID="RunList reference ID" Class="Parameter" Status="Available"
  PartIDKeys="Run" >
  <RunList Run="document's Run value" >
    <LayoutElementRef rRef="LayoutElement reference ID" />
  </RunList>
</RunList>
```

For jobs that contain a single document (file):

```
<LayoutElement ID="LayoutElement reference ID" Class="Parameter"
  Status="Available" >
  <FileSpec DocumentNaturalLanguage="data format" />
</LayoutElement>
<RunList ID="RunList reference ID" Class="Parameter" Status="Available" >
  <LayoutElementRef rRef="LayoutElement reference ID" />
</RunList>
```

For jobs that contain multiple documents (files):

```
<LayoutElement ID="LayoutElement 1 reference ID" Class="Parameter"
  Status="Available" >
  <DocumentNaturalLanguage="data format" />
</LayoutElement>
<LayoutElement ID="LayoutElement 2 reference ID" Class="Parameter"
  Status="Available" >
  <FileSpec DocumentNaturalLanguage="data format" />
</LayoutElement>
<RunList ID="RunList reference ID" Class="Parameter" Status="Available"
  PartIDKeys="Run" >
  <RunList Run="document's Run value" >
    <LayoutElementRef rRef="LayoutElement 1 reference ID" />
  </RunList>
  <RunList Run="document's Run value" >
    <LayoutElementRef rRef="LayoutElement 2 reference ID" />
  </RunList>
</RunList>
```

#### Type

Resettable, single-valued

#### Allowed Values

ID

String of up to 255 characters that uniquely identifies the RunList being referenced.

Run  
String of up to 255 characters that is the reference to the document's Run value.

DocumentNaturalLanguage  
String of up to 255 characters that represents either a language in the form "en" or a language and country code in the form "en-US". If not specified defaults to system specified.

### 3.2.15 Feed Orientation

Specifies the media edge which is to be fed into the device from the paper tray. [\[JDF\] specifies that the feed direction is such that the paper travels along the Y-axis starting at Y=0, so the X-axis is the leading edge.](#)

#### Scope

Document or Job

#### Conformance Levels

2

#### Processes

DigitalPrinting

#### Syntactic Fragment

[There are two ways that the feed direction can be changed.](#)

1. [Swap the x and y dimensions of the media. For example, if typically x is 612 points and y is 792 points, then the short \(612 point\) edge will be the leading edge. To make the leading edge, the long \(792 point\) edge, make the x dimension 792 points and the y dimension 612 points.](#)

[See Media for a description of how to specify Media Dimensions.](#)

2. [Specify that the media is to be rotated and or flipped in the MediaLink.](#)

```
<MediaLink rRef="ID of media resource" Usage="Input"
  CombinedProcessIndex="index of Digital Printing process"
  Orientation="value description" />
```

#### Type

Initially settable or Resettable or Non-settable, single-valued or multi-valued

#### Allowed Values

ID

String of up to 255 characters that uniquely identifies the ResourceName being referenced.

Orientation

One of the values in the following enumeration:

Rotate0	Do not rotate media.
Rotate90	Rotate media 90 degrees counter-clockwise.
Rotate180	Rotate media 180 degrees.
Rotate270	Rotate media 90 degrees clockwise.
Flip0	Flip media vertically.
Flip90	Rotate media 90 degrees clockwise and flip horizontally.
Flip180	Flip media horizontally.
Flip270	Rotate media 90 degrees counter-clockwise and flip horizontally.

ISSUE: Is MediaLink Orientation really what we want to use to represent Feed Orientation? The function that is required is to be able to specify how the media needs to be loaded in the device. In IPP the allowed values are long edge first and short edge first. Would swapping the dimensions in Media Dimensions in JDF accomplish the same as IPP feed orientation? It seems as though the media dimensions at least need to be considered when determining feed orientation. This topic needs more discussion during the Barcelona meetings.

The latest (discussed in the May 20th DP WG call) thinking on this is that the Feed Orientation feature can be accomplished in one of two ways:

1. Since the Y-axis is always the leading edge then specifying the appropriate X and Y dimensions of the media will define the feed orientation.
2. Use the Orientation attribute in the MediaLink to rotate the media.

Must also state that it is typically not necessary to specify feed orientation. Should only specify it when there is a concern about how media is fed, like feeding labels that result in peeling off the labels if fed in the wrong orientation.

6.1 This needs to be discussed out in the Coordinate System working group.

### 3.2.16 Fit Policy

What to do when the page content does not fit on the media.

#### Scope

Page, Document, or Job

#### Conformance Levels

[1](#) and [2](#)

#### Processes

LayoutPreparation

#### Syntactic Fragment

```
<LayoutPreparationParams ID="Reference ID" Class="Parameter" Status="Available" >
  <FitPolicy SizePolicy="what to do when content doesn't fit"
    RotatePolicy="how to rotate when content doesn't fit" />
  <ImageShift
    PositionX="how to horizontally position the content"
    PositionY="how to vertically position the content" />
</LayoutPreparationParams>
```

#### Type

Resettable, single-valued

#### Allowed Values

ID

String of up to 255 characters that uniquely identifies the LayoutPreparationParams being referenced.

RotatePolicy

One of the values in the following enumeration:

NoRotate	This is the default.
RotateOrthogonal	Rotate by 90° in either direction, i.e., rotate if necessary to fit.
RotateClockwise	Rotate clockwise by 90°.
RotateCounterClockwise	Rotate counter-clockwise by 90°.

SizePolicy

One of the values in the following enumeration:

Abort	Emit an error and abort printing. This is the default.
ClipToMaxPage	The page contents should be clipped to the size of the container. The printed area is either centered in the source image.
FitToPage	The page contents should be scaled up or down to fit the media. The aspect ratio is maintained.
ReduceToFit	The page contents should be scaled down but not scaled up to fit the media. The aspect ratio is maintained.
Tile	The page contents should be split into several tiles, each printed on its own surface.

PositionX

One of the values in the following enumeration:

Center	Center the contents horizontally without regard to the limitations of the media's printable area.
Left	Position the left edge of the contents to be coincident with the left edge of the media's printable area.
Right	Position the right edge of the contents to be coincident with the right edge of the media's printable area.
None	Place the contents where the print data specifies. This is the default.

PositionY

One of the values in the following enumeration:

Bottom	Position the bottom edge of the contents to be coincident with the bottom edge of the media's printable area.
Center	Center the contents vertically without regard to the limitations of the media's printable area.
Top	Position the top edge of the contents to be coincident with the top edge of the media's printable area.
None	Place the contents where the print data specifies. This is the default.

### 3.2.17 Folding

How to fold the output.

#### Scope

Page, Document, or Job

#### Conformance Levels

2

#### Processes

Folding

#### Syntactic Fragment

```
<FoldingParams ID="Reference ID" Class="Parameter" Status="Available" >
  NoOp="whether or not to fold"
  DescriptionType="FoldCatalog"
  FoldCatalog="type of folding operation to perform"
  ReferenceEdge="reference edge to use for folding operation" />
ISSUE: 6/10 - Add FoldFromFront boolean attribute.
```

#### Type

Resettable, single-valued

#### Allowed Values

ID

String of up to 255 characters that uniquely identifies the FoldingParams being referenced.

NoOp

Optional. One of the following values:

- false To use other FoldingParams attributes to specify folding. This is the default.
- true To turn folding off. All other FoldingParams attributes are ignored.

DescriptionType

Always set to FoldCatalog.

**ISSUE: There is a proposal to be able to specify a FoldCatalog along with the Fold attribute, which means that DescriptionType is not needed. Can DescriptionType be deprecated? Yes.**

FoldCatalog

A string in the form *Fx-y* that specified the type of fold operation to perform. A solid line indicates to fold up (valley fold) and a dashed line indicates to fold down (mountain fold):

**FP4-1** perform a saddle-fold operation. This fold is used for booklets.



**FP6-2** perform a z-fold in thirds operation. This fold is used for brochures.



**FP6-4** perform a c-fold operation



**P6-6** perform a z-fold operation



ISSUE: It appears that the IPP fold-z-short-sheet value indicates to first fold in half down and then fold the quarter up instead of the fold up and then fold down that the F8-2 specifies.

ISSUE: F8-2 is not really a z-fold in the JDF FoldCatalog because the folding actually occurs in half and then in half again (so the finished folded sheet is 1/4 of the original size). Jim Mekis, Rainer, and Claudia talked in SJ and decided that a new FoldCatalog value should be added that first folds up 1/4 followed by a fold down 1/2. This is more in line with the IPP z-fold although the IPP z-fold does not specify that the first fold is 1/4 and the second fold is 1/2. So propose that an F6-6 be added to the JDF FoldCatalog.

A proposal to add a new fold catalog value was appended to the Finishing wg forum.

6/3- Tom will add this ISSUE to JDF 1.2.

6/16 - TSC decided that there should be a new catalog of P-# values that describe the various product folds.

ISSUE: Who will define these new values?

10/21 - Discussed during DP WG meeting. Should there be another catalog value (P6-7) that allows for a "generic" z-fold where the exact locations of the folds is not specified, so is specified by the system?

All of the other values for FoldCatalog that are included in the JDF Specification Release 1.2 have intentionally been excluded in the ICS for Digital Printing.

#### ReferenceEdge

Optional if reference edge cannot be specified or can be defaulted. One of the values in the following enumeration:

Bottom	Edge that is along the x-axis of the sheet.
Left	Edge that is along the y-axis of the sheet.
Right	Edge that is opposite of the Left edge.
Top	Edge that is opposite of the Bottom edge.
SystemSpecified	This is the default.

ISSUE: What does the ReferenceEdge refer to when folding? Related to Coordinate system discussion.

6/10 - It defines the edge from which folding is performed.

ISSUE: 6/10 - Add new boolean attribute FoldFromFront that is optional where the default is system specified.

ISSUE: The Coordinate Systems working group decided to replace all ReferenceEdge attributes with Orientation attribute in a new Intermediate ComponentLink.

### 3.2.18 Force Page

Force the next page after the specified range of pages on either the front or back side of the sheet. This feature can be used to ensure that chapters will start on a specific side of the sheet and is only honored when two-sided printing is specified.

#### Scope

Page



**Conformance Levels**

2

**Processes**

LayoutPreparation

**Elements/Resources Used**

InsertSheet  
LayoutElement  
RunList

**Syntactic Fragment**

```
<RunList ID="RunList reference ID" Class="Parameter" Status="Available"
  PartIDKeys="Run">
  <!-- The following will insert a sheet after the last page of the specified page range so
  that the next page after the page range will print on the desired side of the sheet. -->
  <RunList Run="document's page range Run value"
    Pages="range of pages before forcing a new sheet">
    <LayoutElementRef rRef="LayoutElement reference ID" />
    <InsertSheet SheetType="FillSheet"
      SheetUsage="whether to force to back or front side"
      SheetFormat=Blank" IsWaste="false" />
  </RunList>
  ... Include as many partitioned RunLists here as needed in the form as shown above.
  ... There should be one for each page range after which the next page will be forced
  ... to print on the specified side.
  <!-- The last page range is for the end of the document. -->
  <RunList Run="document's page range Run value"
    Pages="last page range of document" >
    <LayoutElementRef rRef="LayoutElement reference ID" />
  </RunList>
</RunList>
```

**Type**

Resettable, multi-valued

**Allowed Values**

ID and rRef  
Strings of up to 255 characters that uniquely identify the RunList being referenced.

Run  
String of up to 255 characters that is a reference to a document or to a subset of pages in a document.

Pages  
 $x$ - $y$  Range of pages that are to occur before forcing a new sheet. The last page range begins with the last page to force to a specific side through the end of the document (-1).

SheetType  
Always set to FillSheet.

SheetUsage  
One of the values in the following enumeration:

FillForceBack	If necessary, add a fill page to the back of the last page of the specified page range, so that the next page will print on a new sheet.
FillForceFront	If necessary, add a fill page to the front of the last page of the specified page range, so that the next page will print on a new

sheet.

The other values for SheetUsage that are included in the JDF Specification Release 1.2 are not used when specifying the Start Page feature.

SheetFormat

Always set to Blank.

IsWaste

Always set to false.

Example: When printing two sided, to ensure that pages 3 and 6 are printed on the front side of a sheet, set SheerUsage to FillForceBack, and specify the following page ranges:

x	y	Result
0	2	A fill page will be inserted for the back side of page 2, if page 2 is on the front side.
3	5	A fill page will be included for the back side of page 5, if page 5 is on the front side.
6	-1	Last page range of the document

### 3.2.19 Hold Job

Hold the job on the spooled queue.

#### Scope

Job

#### Conformance Levels

2

#### Processes

None

#### Syntactic Fragment

<JDF ID=... Type="Combined" Types=... Status="Ready" **Activation**="held status of job"  
...>

#### Type

Resettable, single-valued

#### Allowed Values

Activation

One of the values in the following enumeration:

Held Job is held and cannot be processed.

Active Job is released and can be processed.

The following values for Activation are included in the JDF Specification Release 1.2, but have intentionally been excluded in the ICS for Digital Printing:

Inactive

Informative

TestRun

TestRunAndGo

### 3.2.20 ICS Versions

[The versions of the Interopability and Comformance Specifications that were used to generate the job ticket.](#)

**Scope**

n/a

**Conformance Levels**

?

**Processes**

None

**Syntactic Fragment**

<JDF ID=... Type="Combined" Types=... Status="Ready" **ICSVersions="ICS versions"**  
...>

**Type**

Resettable, multi-valued

**Allowed Values**ICSVersions

Strings (NMTOKENS) of the form *ICSName-Version*. To include this ICS for Digital Printing for each of the two conformance levels, specify:

DigitalPrinting\_LVL1-1.0DigitalPrinting\_LVL2-1.0

**ISSUE: There is a new JDF Node attribute called MaxVersion that is being added to JDF 1.2. Should it be included in this ICS?**

**3.2.21 Image Shift Back Side**

Offset the logical page origin as specified, in micrometers, to the right or left and/or above or below the physical page on the back of the sheet.

**Scope**

Page, Document, or Job

**Conformance Levels**

2

**Processes**

LayoutPreparation

**Syntactic Fragment**

<LayoutPreparationParams ID="*Reference ID*" Class="Parameter" Status="Available" >  
<**ImageShift ShiftBack**="*x and y offset amount*" />  
</LayoutPreparationParams>

**Type**

Resettable, single-valued

**Allowed Values**

ID

String of up to 255 characters that uniquely identifies the LayoutPreparationParams being referenced.

ShiftBack

Two doubles, that are the amount to shift the image in the x and y directions.

### 3.2.22 Image Shift Front Side

Offset the logical page origin as specified, in micrometers, to the right or left and/or above or below the physical page on the front of the sheet.

#### Scope

Page, Document, or Job

#### Conformance Levels

2

#### Processes

LayoutPreparation

#### Syntactic Fragment

```
<LayoutPreparationParams ID="Reference ID" Class="Parameter" Status="Available" >
  <ImageShift ShiftFront="x and y offset amount" />
</LayoutPreparationParams>
```

#### Type

Resettable, single-valued

#### Allowed Values

ID

String of up to 255 characters that uniquely identifies the LayoutPreparationParams being referenced.

ShiftFront

Two doubles, that are the amount, in micrometers, to shift the image in the x and y directions.

### 3.2.23 Input Tray Name

Name of the input tray that contains the media to be used.

#### Scope

Page, Document, or Job

#### Conformance Levels

1 and 2

#### Processes

LayoutPreparation

DigitalPrinting

#### Syntactic Fragment

```
<Media ID="Media reference ID" Class="Consumable" Status="Available"
  <Location LocationName="name of input tray" />
</Media>
<LayoutPreparationParams ID="LayoutPreparationParams reference ID" Class="Parameter"
  Status="Available >
  <MediaRef rRef="Media reference ID" />
</LayoutPreparationParams>
<DigitalPrintingParams ID="DigitalPrintingParams reference ID" Class="Parameter"
  Status="Available ManualFeed="whether or not media will be manually fed">
  <MediaRef rRef="Media reference ID" />
</DigitalPrintingParams>
```

**Type**

Resettable, single-valued

**Allowed Values**

ID and rRef

Strings of up to 255 characters that uniquely identify the Media, LayoutPreparationParams, and DigitalPrintingParams being referenced.

LocationName

Optional. String of up to 255 characters characters. Some typical values are as follows:

- AnySmallFormat
- AnyLargeFormat
- AutoSelect
- Bottom same as Lower
- BypassTray
- BypassTray-*n* where *n* is an integer
- ~~Center~~
- Continuous
- Disc
- Disc-*n* where *n* is an integer
- Envelope
- Envelope-*n* where *n* is an integer
- Front
- InsertTray
- InsertTray-*n* where *n* is an integer
- LargeCapacity same as Main
- LargeCapacity-*n* where *n* is an interger
- Left
- Middle
- Rear
- Right
- Roll
- Roll-*n* where *n* is an integer
- Side
- Top same as Upper
- Tray
- Tray-*n* where *n* is an integer, same as Cassette
- SystemSpecified This is the default.

ISSUE: FSG has tray names of Disc, Disc-*n*, Continuous, Roll, Roll-*n*, Tray, and Front. JDF already has the MediaUnit attribute which "describes the format of the media as it is delivered to the device" and Roll is a possible enumeration value. MediaTypeDetails already has the value of Continuous, so should Disc and Roll be added as new values for MediaTypeDetails? 6/3 - See MediaTypeDetails for details of the change to be made to it.

ISSUE: Should Center be removed since it is the same as Middle? 6/3 - Yes?

ISSUE: The above list needs to be updated to match the latest list in the JDF 1.2 Spec. Can't remove any values that were in the JDF 1.1 Spec for backward compatibility reasons?

ManualFeed

Optional. One of the following values:

- false The media will not be fed manually. This is the default.

true                   The media will be fed manually. If LocationName is specified for the input tray name, then the media will be manually fed using the specified input tray name. If LocationName is not specified, then the system specified input tray will be used to manually feed the media.

### 3.2.24 Insert Sheet

Insert a blank (or preprinted) sheet before and/or after a specified page range. If any type of finishing (hole making, stitching, etc.) is specified for the page range, then the insert sheet will be included with the specified page range.

#### Scope

Document or Job

#### Conformance Levels

2

#### Processes

LayoutPreparation

#### Elements/Resources Used

FileSpec  
InsertSheet  
LayoutElement  
Media  
RunList  
Sheet

**ISSUE:** The above heading has been added for a few features to see if it is off any value. There are links to the tables for each Element/Resource used by this feature. Should this be added to the rest of the features?

6/3 - Yes. Change the above links to point to the respective entries in table 3-7 since section 3.2 will be removed. **How do I link to something that is not a heading?**

#### Syntactic Fragment

Following demonstrates how to place an insert sheet before or after a specific page range of a document:

```
<LayoutElement ID="LayoutElement reference ID" Class="Parameter"
  Status="Available" >
  <FileSpec Url="file name" />
</LayoutElement>
<Media ID="Media 2 reference ID" Class="Consumable" Status="Available" />
<RunList ID="RunList reference ID" Class="Parameter" Status="Available"
  PartIDKeys="Run" >
  <RunList Run="insert sheet Run value"
    Pages="x~y page range before or after insert sheet" >
    <LayoutElementRef rRef="LayoutElement reference ID" />
    <InsertSheet SheetType="InsertSheet"
      SheetUsage="where to insert sheet" IsWaste="false" >
      <RunList PageCopies="number of copies of insert sheet" />
      <Sheet>
        <MediaRef rRef="Media 2 reference ID" />
      </Sheet>
    </InsertSheet>
  </RunList>
  <RunList Run="Run value for remainder of document"
```

```

        Pages="z~-1 page range after insert sheet" >
        <LayoutElementRef rRef="LayoutElement reference ID" />
    </RunList>
</RunList>

```

Following demonstrates how to place an insert sheet before or after a specific page range of a job:

... example goes here once we figure out how to specify this ...

### Type

Resettable, multi-valued

### Allowed Values

ID and rRef

Strings of up to 255 characters that uniquely identify the LayoutElement, Media, and RunList being referenced.

Run

Strings that are the references to the Run values for the page range before/after the insert sheet or the page range after the insert sheet. The value can be anything that uniquely identifies this RunList partition.

Pages

x~y            Insert before (SheetUsage=Header) or after (SheetUsage=Trailer) the range of pages starting with *x* and ending with *y*. *x* is 0 to indicate the first page of the document. *y* is -1 to indicate the last page of the document.

z~-1            Page range after insert sheet to the end of the document.

PageCopies

Optional. Integer in the range 1 through 999. Default is 1.

SheetType

Always set to InsertSheet

SheetUsage

One of the values in the following enumeration:

Header            Insert sheet before the specified page(s)

Trailer            Insert sheet after the specified page(s)

The other values for SheetUsage that are included in the JDF Specification Release 1.2 are not used when specifying the InsertSheet feature.

IsWaste

Always set to false.

## 3.2.25 Interleave Sheets

Whether to interleave a printed or blank sheet between each sheet of a document.

### Scope

Document or Job

### Conformance Levels

2

### Processes

LayoutPreparation

### Syntactic Fragment

```
<Media ID="Interleave media reference ID" Class="Consumable" Status="Available" />
```

```

<LayoutPreparationParams ID="LayoutPreparationParams Reference ID"
Class="Parameter"
  Status="Available"
  <InsertSheet SheetType="SeparatorSheet"
    SheetUsage="where to insert interleave sheet"
    SheetFormat="type of sheet to interleave" >
    <Sheet>
      <MediaRef rRef="Interleave media reference ID" />
    </Sheet>
  </InsertSheet>
</LayoutPreparationParams>

```

**Type**

Resettable, single-valued

**Allowed Values**

ID and rRef

Strings of up to 255 characters that uniquely identify the Media and LayoutPreparationParams being referenced.

SheetType

Always set to Separator.

SheetUsage

One of the values in the following enumeration:

Interleaved	Interleave sheet is produced after each page.
InterleavedBefore	Interleave sheet is produced before each page.

SheetFormat

A string (NMTOKEN) of up to 255 characters. Some typical values are as follows:

Duplicate	Print same content as previous sheet onto each interleave sheet.
Blank	Do not print on the interleave sheet. Sheet may be blank or preprinted.
SystemSpecified	This is the default.

The other values for SheetFormat that are included in the JDF Specification Release 1.2 are not used when specifying the Interleave Sheets feature.

**3.2.26 Job Created By**

Name or user ID of the person who submitted the job. This information can be included in the job ticket when the job ticket is created or later.

**Scope**

Job

**Conformance Levels**

1 and 2

**Processes**

None

**Syntactic Fragment**

```

<AuditPool>
  <Created Author="name or user ID of person who created job"
    TimeStamp="date and time of creation" />
</AuditPool>

```

**ISSUE—The TimeStamp attribute is required—can it be optional?**  
**6-3—Leave as is in JDF spec—Add TimeStamp to the above Syntactic Fragment**



**Type**

Initially settable, single-valued

**Allowed Values**

Author

String of up to 255 characters. This may contain the user ID and hostname in the form *userID@hostname*.TimeStampDate and time in ISO 8601 format. If the date and time is not known then zeros for the date and time values can be provided.**3.2.27 Job Modified By**

Name or user ID of the person who modified the job. The hostname may also be specified.

**Scope**

Job

**Conformance Levels**

?

**Processes**

None

**Syntactic Fragment**

&lt;AuditPool&gt;

<Modified Author="name or user ID of person *who modified job*"  
TimeStamp="date and time of modification" />

&lt;/AuditPool&gt;

**ISSUE--The TimeStamp attribute is required, can it be optional?  
6-3--Same as above****Type**

Initially settable, multi-valued

**Allowed Values**

Author

String of up to 255 characters. This may contain the user ID and hostname in the form *userID@hostname*.TimeStampDate and time in ISO 8601 format. If the date and time is not known then zeros for the date and time values can be provided.**3.2.28 Job Name**

User friendly name of the job.

**Scope**

Job

**Conformance Levels**~~1 and 2~~**Processes**

None

**Syntactic Fragment**

```
<CustomerInfo CustomerJobName="job name" />
```

**Type**

Resettable, single-valued

**Allowed Values**

CustomerJobName

User friendly name of the job of up to 255 characters.

**3.2.29 Job Priority**

Scheduling priority of job.

**Scope**

Job

**Conformance Levels**

2

**Processes**

None

**Syntactic Fragment**

```
<NodeInfo JobPriority="priority" />
```

**Type**

Resettable, single-valued

**Allowed Values**

JobPriority

Integer in the range of 1 through 100, where 1 is the lowest priority and 100 is the highest priority.

**3.2.30 Job Ticket Created By**

The name and version of the agent application that created the job ticket.

**Scope**

n/a

**Conformance Levels**

2

**Processes**

None

**Syntactic Fragment**

```
<AuditPool >
```

```
  <Created AgentName="name of agent application"
```

```
    AgentVersion="version of agent application" />
```

```
</AuditPool>
```

**Type**

Initially settable, single-valued

**Allowed Values**

AgentName

Optional. String of up to 255 characters that identifies the agent application that created the job ticket. This can be a company and or product name.

AgentVersion

Optional. String of up to 255 characters that identifies the version of the agent application that created the job ticket.

**3.2.31 Job Ticket Template Comment or Description**

Text that describes what the job ticket template will be used for or what function it provides.

**Scope**

n/a

**Conformance Levels**

2

**Processes**

None

**Syntactic Fragment**

```
<JDF ID=... Type="Combined" Types=... Status="Ready" Template= "true" >
  <Comment Name="TemplateDescription"
    > job ticket template comment or description text
  </Comment>
```

**Type**

Resettable, single-valued

**Allowed Values**

Template

Always set to true.

Name

Always set to TemplateDescription.

*job ticket template comment or description text*

String of up to 1023 characters.

**3.2.32 Job Ticket Template ID**

Unique job ticket template identifier. The intended use is by the application or user that created the job ticket template as a way to uniquely identify a job ticket template.

**Scope**

n/a

**Conformance Levels**

2

**Processes**

None

**Syntactic Fragment**

```
<JDF ID=... Type="Combined" Types=... Status="Ready" Template= "true"
  TemplateID="job ticket template ID" ...>
```

**Type**  
Resettable, single-valued

**Allowed Values**  
 Template  
     Always set to true.  
 TemplateID  
     String of up to 255 characters. Could be a name or ID.

### 3.2.33 Job Ticket Template Version

The version of the job ticket template. The intended use is by the application or user that created the job ticket template as a way to identify the job ticket template.

**Scope**  
n/a

**Conformance Levels**  
2

**Processes**  
None

**Syntactic Fragment**  
 <JDF ID=... Type="Combined" Types=... Status="Ready" **Template**="true"  
     **TemplateVersion**="job ticket template version" ...>

**Type**  
Resettable, single-valued

**Allowed Values**  
 Template  
     Always set to true.  
 TemplateVersion  
     String of up to 255 characters that may be in the form a.b, where a is the major version number and b is the minor version number.

### 3.2.34 Jog Offset

Jog offset between copies of a job and between jobs.

**Scope**  
Job

**Conformance Levels**  
2

**Processes**  
DigitalPrinting

**Syntactic Fragment**  
 <DigitalPrintingParams ID="Reference ID" Class="Parameter" Status="Available"  
     <**Disjointing OffsetDirection**="whether to job offset between job copies" />  
 </DigitalPrintingParams>

**Type**

Resettable, single-valued

**Allowed Values**

ID

String of up to 255 characters that uniquely identifies the DigitalPrintingParams being referenced.

OffsetDirection

One of the values in the following enumeration:

- Alternate      The position of the first component is opposite to the position of the previous component and subsequent components are each offset to alternating positions. For example, if the last item in the stack was positioned to the right then the subsequent items will be positioned to the left, right, left, right, etc.
- Left            Offset consecutive components sideways to the left, next to the right. Left is with respect to the coordinate system feed direction.
- None            Do not offset consecutive components. The position of all components is the same as the position of the previous component. This is the default.
- Right           Offset consecutive components sideways to the right, next to the left.
- Straight        Same as *None*.
- SystemSpecified    Offset consecutive components to a system specified position, which may be *None*.

ISSUE: Remove above value descriptions since they are a duplicate of what is in the JDF spec.

**3.2.35 Margins**

The left, bottom, right, and top margins that the device should allow. Specifying zero for all margins indicates that the page image data is to be imaged to all edges and beyond the edges of the imageable area.

**Scope**

Page, Document, or Job

**Conformance Levels**

[?1 and 2](#)

**Processes**

DigitalPrinting

**Syntactic Fragment**

```
<DigitalPrintingParams ID="Reference ID" Class="Parameter" Status="Available"
  NonPrintableMarginsBottom = "left, bottom, right, top marginswidth of bottom margin"
  NonPrintableMarginLeft = "width of left margin"
  NonPrintableMarginRight = "width of right margin"
  NonPrintableMarginTop = "width of top margin" />
```

ISSUE: Make this 4 separate attributes, NonPrintableMarginLeft, NonPrintableMarginBottom, NonPrintableMarginTop, NonPrintableMarginRight. Tom and Steve S will work on the descriptions for the 4 attributes.

**Type**

Resettable, single-valued

**Allowed Values**

ID

String of up to 255 characters that uniquely identifies the DigitalPrintingParams being referenced.

NonPrintableMarginsBottom

~~Four white space separated doubles "l b r t" that are to be the left, bottom, right, and top margins~~ Double that is the width of the bottom margin, in points. Specifying zero for any of the doubles indicates that there is to be no bottom margin ~~for that side~~.

NonPrintableMarginLeft

Double that is the width of the left margin, in points. Specifying zero indicates that there is to be no left margin.

NonPrintableMarginRight

Double that is the width of the right margin, in points. Specifying zero indicates that there is to be no right margin.

NonPrintableMarginTop

Double that is the width of the top margin, in points. Specifying zero indicates that there is to be no top margin.

### 3.2.36 Media

Name and properties of the media on which to print.

#### Scope

Page, Document, or Job

#### Conformance Levels

1 and 2

**ISSUE:** Should only a subset of Media be supported in level 1?

#### Processes

DigitalPrinting

LayoutPreparation

#### Syntactic Fragment

To print the entire document using the same Media:

```
<Media ID="Media reference ID" Class="Consumable" Status="Available"
  BackCoating="coating already applied to back side of media"
  Brand="brand name of media"
  Dimension="media x and y dimensions"
  DimensionName="dimension name of the media"
  FrontCoating="coating already applied to front side of media"
  HoleType="number of pre-punched holes"
  ImagableSide="side of media that may be marked"
  MediaColorMeasurement="media color in CIEXYZLAB color space"
  MediaColorName="name of media color"
  MediaColorNameDetails="details of media color name"
  MediaType="media type"
  MediaTypeDetails="media type details"
  MediaSetCount="number of media in each set"
  MediaUnit="format of media as it is delivered"
  Opacity="opacity of the media"
  Preprinted="whether media is preprinted or not"
  Recycled="whether recycled media is requested or not"
  RecycledPercentage="percentage of recycled material in media"
  StockType="stock type of media"
  Weight="media weight"
  <Comment Name="Description"> text description of media </Comment>
```

```

</Media>
<LayoutPreparationParams ID="LayoutPreparationParams reference ID"
  Class="Parameter" Status="Available" >
  <MediaRef rRef="Media reference ID" />
</LayoutPreparationParams>
<DigitalPrintingParams ID="DigitalPrintingParams reference ID"
  Class="Parameter"
  Status="Available" >
  <MediaRef rRef="Media reference ID" />
</DigitalPrintingParams>

```

For example, to print part of the document using one Media and the other part using another Media.

```

<Media ID="Media 1 reference ID" Class="Consumable"
  Status="Available"
  ... same options as shown above ... />
</Media>
<Media ID="Media 2 reference ID" Class="Consumable"
  Status="Available"
  ... same options as shown above ... />
</Media>
<LayoutPreparationParams ID="LayoutPreparationParams reference ID"
  Class="Parameter" Status="Available
  PartIDKeys="DocIndex DocRunIndex" >
  <MediaRef rRef="Media 1 reference ID" />
  <LayoutPreparationParams DocIndex="index of document" >
    <LayoutPreparationParams
      DocRunIndex="pages or ranges of pages" >
      <MediaRef rRef="Media 2 reference ID" />
    </LayoutPreparationParams>
  </LayoutPreparationParams>
</LayoutPreparationParams>
<DigitalPrintingParams ID="DigitalPrintingParams reference ID"
  Class="Parameter" Status="Available
  PartIDKeys="DocIndex DocRunIndex" >
  <MediaRef rRef="Media 1 reference ID" />
  <DigitalPrintingParams DocIndex="index of document"
    <DigitalPrintingParams
      DocRunIndex="pages or ranges of pages" >
      <MediaRef rRef="Media 2 reference ID" />
    </DigitalPrintingParams>
  </DigitalPrintingParams>
</DigitalPrintingParams>

```

ISSUE: Should DigitalPrintingParams above be changed to use BundleItemIndex instead of DocIndex and DocRunIndex? No, the specification of media does not require the use of bundle items. In fact there could be a conflict if BundleItemIndex was used and the bundle item contained insert sheets and media was also specified for the InsertSheet in the RunList.

#### Type

Resettable, single-valued

#### Allowed Values

ID and rRef

Strings of up to 255 characters that uniquely identify the Media, LayoutPreparationParams, and DigitalPrintingParams being referenced.

#### DocIndex

Index of document being referenced.

#### DocRunIndex

- $x$  Single page to print on media. Zero or more page numbers can be specified.
- $x\ y$  Range of pages to print on media. Zero or more page ranges can be specified.

#### BackCoatings

Optional. One of the values in the following enumeration:

- None This is the default.
- Coated A coating of a system specified type.
- Glossy
- HighGloss
- InkJet A coating intended for use with inkjet technology.
- Matte
- Satin
- Semigloss

#### Brand

Optional. Strings of up to 255 characters that is information about the media, such as the manufacturer, model, part number, and/or type.

#### Dimension

Optional. Doubles for x and y dimensions expressed in points. Required if DimensionName is present.

#### DimensionName

Optional. String (NMTOKEN) of up to 255 characters. Dimension must be specified if DimensionName is present. See [IEEE-ISTO5101.1] for a list of typical values to use for DimensionName.

**ISSUE: TSC believes that DimensionName is not necessary, that Dimension is all that is needed to describe the Media. Does the DP WG agree with this decision?**

#### FrontCoatings

Optional. One of the values listed above in BackCoating.

#### HoleType

Optional if number of holes cannot be specified or can be defaulted. One of the values in the following enumeration:

- None This is the default.
- S1-generic 1 pre-punched hole
- R2-generic 2 pre-punched holes
- R3-generic 3 pre-punched holes
- R4-generic 4 pre-punched holes
- R5-generic 5 pre-punched holes
- R6-generic 6 pre-punched holes
- R7-generic 7 pre-punched holes
- SystemSpecified

All of the other values for HoleType that are included in the JDF Specification Release 1.2 have intentionally been excluded in the ICS for Digital Printing.

#### ImagableSide

Optional. One of the values in the following enumeration:

- Back
- Both This is the default.
- Front
- Neither

#### MediaColorMeasurement



Optional. Absolute LabColor of the media in CIE~~XYZ~~LAB color space. Must be in the form "~~x~~-~~y~~-~~z~~l a b" where ~~x~~l, ~~y~~a, and ~~z~~b are space delimited doubles.

MediaColorName

Optional. One of the values in the following enumeration:

Black  
Blue  
Brown  
Buff  
Cyan  
Gold  
Goldenrod  
Gray  
Green  
Ivory  
Magenta  
MultiColor  
Mustard  
NoColor  
Orange  
Pink  
Red  
Silver  
Turquoise  
Violet  
White  
Yellow

SystemSpecified This is the default.

Any of the above, except SystemSpecified, with the prefix "Clear", "Dark" or "Light".

MediaColorNameDetails

Optional. String of up to 255 characters without spaces. Some typical values are as follows:

Cardinal  
Royal  
Ruby  
MyColor  
UserColor  
UserColor-*n*                    where *n* is an integer

MediaType

Optional. One of the values in the following enumeration:

Disc  
Other  
Paper  
Transparency  
Unknown                    This is the default.

ISSUE: Should Disc be added here or should an Other value be added and then Disc added as a new value for MediaTypeDetails?

6/5- Add both Disc and Other..

MediaTypeDetails

Optional. String (NMTOKEN) of up to 255 characters without spaces. Some typical values are as follows:

CardBoard                    Thick stock.  
~~Continuous~~                    ~~Continuously connected sheets of an opaque material. Which edge is connected is not specified.~~

ISSUE- Move to MediaUnit

ContinuousLong	Continuously connected sheets <del>of an opaque material</del> connected along the long edge. <del>ISSUE - Move to MediaUnit</del>
ContinuousShort	Continuously connected sheets <del>of an opaque material</del> connected along the short edge. <del>ISSUE - Move to MediaUnit</del>
Envelope	Envelopes that can be used for conventional mailing purposes.
EnvelopePlain	Envelopes that are not preprinted and have no windows.
EnvelopeWindow	Envelopes that have windows for addressing purposes.
FullCutTabs	Media with tabs that run full length of media, where only one tab is visible.
Labels	Label stock, e.g., a sheet of peel-off labels.
Letterhead	Media that has a letterhead.
MultiLayer	Form medium composed of multiple layers which are preattached to one another, e.g., for use with impact printers.
MultiPartForm	Form medium composed of multiple layers not preattached to one another; each sheet may be drawn separately from an input source.
<del>Paper</del>	<del>Proof or product component paper. ISSUE - Can Paper be removed since there is it provides no value when Media Type = "Paper"? Also isn't Paper the same as Stationery? Also MediaIntent's UserMedia Type doesn't include Paper? 6/3 - Agreed to deprecate Paper.</del>
Photographic	Separately cut sheets of an opaque material to produce photographic quality images.
PreCutTabs	Media with tabs that are cut so that more than one tab is visible.
Stationery	Media that is stationery.
TabStock	Media with tabs that are either FullCutTabs or PreCutTabs. Use with MediaSetCount for number of tabs.
<del>Tractor</del>	<del>Tractor fed paper.</del>
<del>Transparency</del>	<del>Media is a transparency. 6/3 - Deprecate this since it is already in Media Type.</del>
<del>Unknown</del>	<del>Media type details are unknown. This is the default.</del>

ISSUE: Should Roll be added as a value for MediaTypeDetails? IPP includes it 6/3 - Decided to do the following:

- Move Continuous, ~~ContinuousLong, and ContinuousShort~~ from MediaTypeDetails to MediaUnit
- Add MediaUnit attribute to this ICS.
- Add Fan as a new value for MediaUnit. Roll is already a value of MediaUnit. 6/10 - Don't add Fan since it is the same as Continuous.
- Add FanFed to DigitalPrintingParams PrintingType

ISSUE: Should DigitalPrintingParams PrintingType be included in this ICS? If so, where? As a new feature, as part of the Media feature, or some other feature? 6/10 - Add ContinuousFed to DigitalPrintingParams instead of FanFed.

- 6/10 - Add Tractor to MediaTypeDetails.

The following values for MediaTypeDetails are included in the JDF Specification Release 1.2, but have intentionally been excluded in the ICS for Digital Printing:

Aluminum  
 DryFilm  
 CtPVisiblePhotoPolymer  
 CtPThermal  
 ImageSetterPaper  
 PlateUV  
 Polyester  
 WetFilm

ISSUE: In SJ, Rainer proposed that a group be assigned the task of deciding valid MediaTypeDetails. Martin suggested looking at papiNet. Rainer briefly looked at the spec during the meeting and didn't see that it would be of any help. 6/3 - This still needs to be looked into.

#### MediaSetCount

Optional. Integer in the range 1 through 65535.

#### MediaUnit

Optional. One of the values in the following enumeration:

Continuous  
Roll  
Sheet

#### Opacity

Optional. One of the values in the following enumeration:

Opaque            This is the default.  
 Transparent  
 Translucent

#### Preprinted

Optional. One of the following values:

true            The media is preprinted.  
 false          The media is not preprinted. This is the default.

#### Recycled

Optional. One of the following values:

true            The media is made of recycled material.  
 false          The media is not made of recycled material. This is the default.

#### RecycledPercentage

Optional. Double that is a percentage between 0 and 100 of recycled material that the media contains. If RecycledPercentage is not present the media contains a system specified percentage of recycled material. Recycled must be set to true when RecycledPercentage is present. If Recycled is false or omitted, RecycledPercentage is ignored.

#### StockType

Optional. String (NMTOKEN) of up to 255 characters without spaces. Some typical values are as follows:

Bristol  
 Bond  
 Cover  
 Newsprint  
 Index  
 Offset

Tag  
Text

ISSUE: What is the default? Should Unknown be added and it be the default like MediaTypeDetails?  
6-10— Is Unknown the same as system specified? If it is then Unknown should be removed for MediaType and MediaTypeDetails and not be added to StockType because if these attributes are omitted then the result is system specified.

*text description of media*

String of up to 1023 characters.

Weight

Optional. Double that is weight of media in grams per square meter.

### 3.2.37 Message to Operator

Message from the end user to indicate something about the processing of the job. Note: The natural language is not localized by any recipient, since this message is generated by a human.

#### Scope

Job

#### Conformance Levels

2

#### Processes

None

#### Syntactic Fragment

`<Comment Name=" OperatorText"> message text to the operator </Comment>`

#### Type

Resettable, single-valued

#### Allowed Values

Name

Always set to OperatorText.

*message text to the operator*

String of up to 1023 characters.

### 3.2.38 Notification

Specifies who should be notified when certain events pertaining to the job occur.

#### Scope

Job

#### Conformance Levels

2

#### Processes

None

#### Syntactic Fragment

`<NodeInfo>`

`<NotificationFilter`

`Classes="Event"`

```

    Types="events to deliver">
    ChannelType="how to deliver notification"
    Locator="where to deliver notification"
    <Comment
      Language="language code or language and country code"
      Name="Description"> comment text to accompany notification
    </Comment>
  </NotificationFilter>
</NodeInfo>

```

**ISSUE:** Still open. See Digital Printing Action item #13. The Asset Transfer group has a proposal for this feature. See June 10th AT Forum proposal. Fix this feature once proposal is completed and accepted.

## Type

Resettable, multi-valued (so multiple Notification elements can be present)

## Allowed Values

### Classes

Always set to Event.

### Types

Optional. Strings of up to 255 characters containing event identifiers as defined by the system.

Note: Ignored by systems where the type of events delivered are pre-defined and cannot be specified.

### ChannelType

Optional if type of delivery is system specified or can be defaulted. One of the values in the following enumeration:

Email An email address to send the events to.

Fax A fax number to send the events to.

File A file to write the events to.

**ISSUE:** How is this specified? Note asking this question was appended to AT forum on 6/10.

6/11 - [Israel Viente](#) responded that he will propose to add a LogFile value to the new ChannelTypeDetails attribute.

JMF A JMF messaging channel to sent the events to.

Phone A phone number to send the events to.

WWW A WWW home page to send events to.

None Do not send any events.

**ISSUE:** ChannelType is not currently in JDF as an attribute for NotificationFilter. Propose that it be added.

### Locator

Optional if ChannelType is None. String of up to 255 characters that specifies where or to whom the events are to be send.

**ISSUE:** Locator is not currently in JDF as an attribute for NotificationFilter. Propose that it be added.

### Comment

#### Language

Optional. String of up to 255 characters that represents either a language in the form "en" or a language and country code in the form "en-US". If not specified defaults to system specified.

#### Name

Always set to Description.

*comment text to accompany notification*

Optional if comment cannot be specified. String of up to 255 characters

Note: Ignored by systems that don't allow a comment to be specified.

### 3.2.39 Number of Copies

Number of copies of the job to print.

#### Scope

Job

#### Conformance Levels

1 and 2

#### Processes

DigitalPrinting

#### Syntactic Fragment

```
<ComponentLink rRef="ID of Component resource" Usage="Output"
  CombinedProcessIndex="index of Digital Printing process or last finishing process"
  Amount="number of copies"/>
```

#### Type

Resettable, single-valued

#### Allowed Values

rRef

String of up to 255 characters that uniquely identifies the Component resource being linked.

Amount

Optional. Integer in the range of 1 through 32 bits. If not specified, then the result is as if 1 is specified.

### 3.2.40 Number Up

Number of pages to image on a single side of the sheet. The Fit Policy feature on page 69 can be used with the Number Up feature.

#### Scope

Page, Document, or Job

#### Conformance Levels

1 and 2

#### Processes

LayoutPreparation

#### Syntactic Fragment

```
<LayoutPreparationParams ID="Reference ID" Class="Parameter" Status="Available"
  NumberUp="number of pages along yx axis and xy axis" />
```

**ISSUE:** Decision was made (see DP forum append on 5/26) to swap meaning of x and y so that NumberUp is defined as x columns and y rows.

#### Type

Resettable, single-valued

#### Allowed Values

ID

String of up to 255 characters that uniquely identifies the LayoutPreparationParams being referenced.

NumberUp

~~$y \times x$~~   ~~$x \times y$~~  Place the pages in  $y$  rows  $x$  columns and  $x$  columns  $y$  rows on the sheet, where  $x$  and  $y$  are integers.

### 3.2.41 Output Bin Name

Name of the output bin to which output will be directed. Specifying output bin name overrides any output bin name used by the form definition.

**Scope**

Page, Document, or Job

**Conformance Levels**

1 and 2

**Processes**

DigitalPrinting

**Syntactic Fragment**

<DigitalPrintingParams ID="*Reference ID*" Class="Parameter" Status="Available"  
**OutputBin**="*output bin name*" />

**Type**

Resettable, single-valued

**Allowed Values**

ID

String of up to 255 characters that uniquely identifies the DigitalPrintingParams being referenced.

OutputBin

String (NMTOKEN) of up to 255 characters without spaces. Some typical values are as follows:

- Booklet
- Bottom same as Lower
- Center
- FaceDown
- FaceUp
- FitMedia
- LargeCapacity same as Main
- Left
- Mailbox-*n* where *n* is an integer
- Middle
- MyMailbox
- Rear
- Right
- Side
- Stacker-*n* where *n* is an integer
- SystemSpecified this is the same as AutoSelect and is the default
- Top same as Upper
- Tray-*n* where *n* is an integer

**ISSUE: Should Center be removed and Middle used instead? Should Front be added?**

### 3.2.42 Page Delivery

How pages are to be delivered to the output bin or stacker.

**Scope**

Document or Job

**Conformance Levels**2**Processes**

DigitalPrinting

**Syntactic Fragment**

```
<DigitalPrintingParams ID="Reference ID" Class="Parameter" Status="Available"
  PageDelivery="how to deliver pages" />
```

**Type**

Resettable, single-valued

**Allowed Values**

ID

String of up to 255 characters that uniquely identifies the DigitalPrintingParams being referenced.

PageDelivery

One of the values in the following enumeration:

Fanfold	Alternate face up and face down.
SameOrderFaceUp	Page order as is with the front sides of the media up.
SameOrderFaceDown	Page order as is with the front sides of the media down.
ReverseOrderFaceUp	Page order reversed with the front sides of the media up.
ReverseOrderFaceDown	Page order reversed with the front sides of the media down.
SystemSpecified	Order is face up or face down as specified by the system. This is the default.

**3.2.43 Page Distribution**

How pages are to be distributed onto a multi-up grid of finished page cells. To be used with the Number Up and Presentation Direction features.

**Scope**

Document or Job

**Conformance Levels**

2

**Processes**

LayoutPreparation

**Syntactic Fragment**

```
<LayoutPreparationParams ID="Reference ID" Class="Parameter" Status="Available"
  PageDistributionScheme="how to distribute the pages" />
```

**Type**

Resettable, single-valued

**Allowed Values**



ID	String of up to 255 characters that uniquely identifies the LayoutPreparationParams being referenced.
PageDistributionScheme	One of the values in the following enumeration:
Saddle	Distribute pages onto a sequence of one or more imposition layouts in proper order for saddle stitch binding
Perfect	Distribute pages onto a sequence of one or more signatures in proper order for perfect binding
Sequential	The pages are distributed onto the multi-up layout according to the value of the PresentationDirection attribute. This is the default.

### 3.2.44 Page Order

How the pages in a document or job are ordered.

#### Scope

Document or Job

#### Conformance Levels

2

#### Processes

LayoutPreparation

#### Syntactic Fragment

```
<LayoutPreparationParams ID=Reference ID Class="Parameter" Status="Available
  PageOrder=order of pages />
```

#### Type

Resettable, single-valued

#### Allowed Values

ID

String of up to 255 characters that uniquely identifies the LayoutPreparationParams being referenced.

PageOrder

One of the values in the following enumeration:

- |         |   |
|---------|---|
| Booklet | Pages are in booklet order.                     |
| Reader  | Pages are in reader order. This is the default. |

### 3.2.45 Presentation Direction

Indicates the order in which content pages will be distributed into the page cells. To be used with the Number Up and Page Distribution features.

#### Scope

Document or Job

#### Conformance Levels

2

**Processes**

LayoutPreparation

**Syntactic Fragment**

<LayoutPreparationParams ID=*Reference ID* Class="Parameter" Status="Available"  
**PresentationDirection**=*order to distribute pages* />

**Type**

Resettable, single-valued

**Allowed Values**

ID

String of up to 255 characters that uniquely identifies the LayoutPreparationParams being referenced.

PresentationDirection

One of the values in the following enumeration:

FoldCatalog

Pages are imaged so that the result is compatible with a finished product produced from the folding catalog as specified in FoldCatalog.

XYZ

Permutations of the letters XYZ and xyz so that exactly one of upper or lower case of x y and z define the order in which content pages are flowed along each axis with respect to the coordinate system of the front side of the sheet.

X Specifies flowing left to right across a sheet surface.

x Specifies flowing right to left across a sheet surface.

Y Specifies flowing bottom to top vertically across a sheet surface.

y Specifies flowing top to bottom vertically across a sheet surface.

Z Specifies flowing bottom of stack to top of stack through the stack.

z Specifies flowing top of stack to bottom of stack through the stack.

If not present, the default value is *SystemSpecified*:

The following table specifies how cells are ordered on simplex 4-up depending on XYZ.

SystemSpecified

Pages are imaged onto the Number Up layout as determined by the device. This is the default.

**3.2.46 Print Quality**

The print quality that the device is to use for the documents in a job.

**Scope**

Document or Job

**Conformance Levels**

[?1 and 2](#)

**Processes**

Interpreting

**Syntactic Fragment**

```
<InterpretingParams ID="Reference ID" Class="Parameter" Status="Available"
  PrintQuality="print quality" />
```

**Type**

Resettable, single-valued

**Allowed Values**

ID

String of up to 255 characters that uniquely identifies the InterpretingParams being referenced.

PrintQuality

One of the values in the following enumeration:

Draft                   Lowest quality available on the device.

High                    Highest quality available on the device.

Normal                 Default quality provided by the device. This is the default.

**ISSUE:** FSG also has Fine, Economy, and Photo values. These values are part of the printer MIB. Should they be added to JDF Spec?

6/10 - Discussed this during FSG meeting. These are not in the printer MIB. Decided that Economy and Fine should be added, but Photo should not.

6/24 - Proposal was sent to PWG to add new print-optimize attribute and to leave print-quality as is. Proposal is to be discussed during PWG meeting on 7/10. 8/22-FSG and PWG decided to add a new attribute called print-content-optimize (possible values are photo, graphics, text and text-and-graphics) that describes the optimizing processing that is to be performed on the content and will work with the print-quality attribute, so the print-quality attribute will not change. **ISSUE:** Should we wait until the next version of the ICS to add this new attribute or add it now?

**3.2.47 Process Color Model**

The model to be used for rendering the colorants defined in color spaces into process colorants.

**Scope**

Page, Document, or Job

**Conformance Levels**

2

**Processes**

DigitalPrinting

**Syntactic Fragment**

```
<ColorantControl ID="Reference ID" Class="Parameter" Status="Available"
  ProcessColorModel="process color mode" />
```

**Type**

Resettable, single-valued

**Allowed Values**

ID

String of up to 255 characters that uniquely identifies the ColorantControl being referenced.

ProcessColorModel

A string (NMTOKEN) of up to 255 characters. Some typical values are as follows:

DeviceCMY

DeviceCMYK

DeviceGray

DeviceN  
 DeviceRGB  
 SystemSpecified                      This is the default.

### 3.2.48 Proof Print

Whether or not to proof print the job and if so who should approve the proof..

#### Scope

Job

#### Conformance Levels

2

#### Processes

Rendering  
 Approval  
 DigitalPrinting

#### Syntactic Fragment

```
<RenderingParams ID="Reference ID" Class="Parameter" Status="Available" >
  <ObjectResolution Resolution="device resolution to use" />
</RenderingParams>
<DigitalPrintingParams ID="Reference ID" Class="Parameter" Status="Available"
  DirectProofAmount="number of copies to print for the proof" />
<ApprovalParams ID="Reference ID" Class="Parameter" Status="Available"
  <ApprovalPerson>
    <Contact ContactType="Approver"
      <Address City="city" Country="country"
        PostBox="post office box"
        PostalCode="zip or postal code"
        Region="state or province"
        Street="street address"
        ExtendedAddress="additional address info" />
      <ComChannel ChannelType="communication type"
        Locator="communication details" />
      <Company OrganizationName="company/organization name" />
      <Person FirstName="first name"
        FamilyName="family/last name" />
    </Contact>
  </ApprovalPerson>
</ApprovalParams>
```

ISSUE: Is this how a proof print should be done? What if you want to specify that a proof print be done followed by a final print, both in the same job ticket? Should a boolean **DirectProof** or **ProofType** attribute be added to **DigitalPrintingParams**? Tom will lead a group to figure this out.

ISSUE: The Proof Print feature should be removed because there really isn't a way to print a proof, approve the proof, and then do a final print in an integrated digital printer without tying up the printer which is not reasonable (unless the job can be put into a held state). The proof print and the final print are most likely two separate jobs, although the same job ticket could be reused for the final print with a few modifications. It might be possible to have an Approval feature where a proof print is done followed by the Approval process. The Approval process merely includes the **ApprovalParams** resource that contains information about the approver. Then the separate final job can be submitted following the approval.

6/10 - Decided that this feature should be renamed to "Approval" and moved to the A section.

7/8 – DP WG decided to add a new DirectProofAmount integer attribute to the DigitalPrintingParams resource. If present then the specified number of proof copies is produced and the JDF node's Status attribute is set to "Stopped". After the proof is approved by the specified contact, the Status attribute is changed to "Ready", and then the final run occurs. The Approval feature and the Proof Print feature can both be specified if an approval of a proof is required and an approval of the final run is also required. If there are two Approval processes then the DigitalPrintingParams resource's DirectProofAmount attribute applies to the first Approval process.

**Type**

Resettable, single-valued

**Allowed Values****ID**

String of up to 255 characters that uniquely identifies the RenderingParams, DigitalPrintingParams, or ApprovalParams being referenced.

**Resolution**

Optional.

$x$   $y$  Horizontal and vertical resolution, where  $x$  and  $y$  are in DPI.

**DirectProofAmount**

**Optional. Integer.**

**ContactType**

Always set to Approver.

**City**

Optional. String of up to 255 characters that is the name of the city.

**Country**

Optional. String of up to 255 characters that is the name of the country.

**PostBox**

Optional. String of up to 255 characters that is the post office box (e.g P.O Box 101).

**PostalCode**

Optional. String of up to 255 characters that is the zip or postal code.

**Region**

Optional. String of up to 255 characters that is the state or province.

**Street**

Optional. String of up to 255 characters that is the street address.

**ExtendedAddress**

Optional. String of up to 255 characters that is the additional address information (e.g. Suite 245).

**ChannelType**

One of the values in the following enumeration:

Phone Telephone number.

Email Email address.

Fax Fax number.

WWW WWW home page or form.

**Locator**

String of up to 255 characters that is the details about how to communicate (e.g. phone number, fax number, email address).

**OrganizationName**

String of up to 255 characters that is the name of the company or organization.

**FirstName**

Optional. String of up to 255 characters that is the first name of the contact.

**FamilyName**

Optional. String of up to 255 characters that is the family or last name of the contact.

### 3.2.49 Punching and Hole Making

How many holes to punch and where to punch the holes in the output.

#### Scope

Page, Document, or Job

#### Conformance Levels

?

#### Processes

HoleMaking

#### Syntactic Fragment

```
<HoleMakingParams ID=Reference ID Class="Parameter" Status="Available"
  NoOp=whether or not to punch holes
  HoleType=number of holes to punch
  HoleReferenceEdge=which edge to punch />
```

#### Type

Resettable, single-valued

#### Allowed Values

ID

String of up to 255 characters that uniquely identifies the HoleMakingParams being referenced.

NoOp

Optional. One of the following values:

true	To turn hole making off. All other HoleMakingParams attributes are ignored.
false	To use other HoleMakingParams attributes to specify hole making. This is the default.

HoleType

Optional if number of holes cannot be specified or can be defaulted. One of the values in the following enumeration:

<a href="#">S1-generic</a>	
R2-generic	
R3-generic	
R4-generic	
R5-generic	
R6-generic	
R7-generic	
SystemSpecified	This is the default.

HoleReferenceEdge

Optional if reference edge cannot be specified or can be defaulted. One of the values in the following enumeration:

Bottom	Edge that is along the x-axis of the sheet.
Left	Edge that is along the y-axis of the sheet. This is the default.
Right	Edge that is opposite of the Left edge.
Top	Edge that is opposite of the Bottom edge.

**ISSUE:** The Coordinate Systems working group decided to replace all ReferenceEdge attributes with Orientation attribute in a new Intermediate ComponentLink

The following values for HoleReferenceEdge are included in the JDF Specification Release 1.2, but have intentionally been excluded in the ICS for Digital Printing:

Explicit

### 3.2.50 Range of Pages to Include

Range of pages in a document that will be included and that the job ticket will refer to.

#### Scope

Document or Job

#### Conformance Levels

2

#### Processes

LayoutPreparation

#### Syntactic Fragment

```
<LayoutElement ID="LayoutElement reference ID" Class="Parameter" Status="Available" >
  <FileSpec URL="document file name" />
</LayoutElement>
<RunList ID="RunList reference ID" Class="Parameter" Status="Available"
  PartIDKeys="Run" >
  <RunList Run="document's Run value"
    Pages="pages or ranges of pages to include" >
    <LayoutElementRef="LayoutElement reference ID" />
  </RunList>
</RunList>
```

#### Type

Resettable, single-valued

#### Allowed Values

ID and rRef

Strings of up to 255 characters that uniquely identify the LayoutElement and RunList being referenced.

Run

String of up to 255 characters that is the reference to the document's Run value.

Pages

- x Page numbers to include to generate the job ticket. Zero or more page numbers can be specified.
- x~y Page ranges to include to generate the job ticket, where x is the first page and y is the last page. When y is -1, page x through the last page. Zero or more page ranges can be specified.

### 3.2.51 Range of Pages to Output

Range of pages in the job or document to output. The value specified must be a subset of Range of Pages to Include on page 103. If not specified, it defaults to Range of Pages to Include.

If two-sided printing is specified, then the pages will appear on the same side of the sheet that they would have appeared on if the entire range of pages was being processed. So, pages that would appear on the front side when the entire range of pages is processed will appear on the front side of the sheet when a subset of pages is being processed and pages that would appear on the back side when the entire range of pages is processed will appear on the back side of the sheet when a subset of pages is being processed.

#### Scope

Document or Job

#### Conformance Levels

1 and 2

### Processes

LayoutPreparation

### Syntactic Fragment

The following syntactic fragment must be used when specific pages of the job are to be output:

```
<RunListLink rRef="ID of RunList resource being referenced" Usage="Input"
  CombinedProcessIndex="0 1" >
  <Part RunIndex="pages or ranges of pages of job to output" />
</RunListLink>
```

The following syntactic fragment must be used when specific documents are to be output:

```
<RunListLink rRef="ID of RunList resource being referenced" Usage="Input"
  CombinedProcessIndex="0 1" >
  <Part DocIndex="indices of documents" />
</RunListLink>
```

The following syntactic fragment must be used when specific pages of specific documents are to be output:

```
<RunListLink rRef="ID of RunList resource being referenced" Usage="Input"
  CombinedProcessIndex="0 1" >
  <Part DocIndex="index of document" DocRunIndex=
    "pages or ranges of pages of document to output" />
    ... Partitions for additional DocIndex/DocRunIndex may appear here
</RunListLink>
```

### Type

Resettable, multi-valued

### Allowed Values

ID and rRef

Strings of up to 255 characters that uniquely identify the RunList being referenced. RunIndex or DocRunIndex

x Page numbers that are to be processed. Zero or more page numbers can be specified.

x~y Page ranges to process where x is the first page and y is the last page to be processed. When y is -1, page x through the last page is processed. Zero or more page ranges can be specified.

DocIndex

One or more integers that represent the indices of the documents in the job.

## 3.2.52 Rotate Page

Degrees of rotation of the text or image on a page with respect to the way the form is loaded in the device.

### Scope

Page, Document, or Job

### Conformance Levels

1 and 2

### Processes

LayoutPreparation



**Syntactic Fragment**

```
<LayoutPreparationParams ID=Reference ID Class="Parameter" Status="Available"
  Rotate=degrees of rotation />
```

**Type**

Resettable, single-valued

**Allowed Values**

ID

String of up to 255 characters that uniquely identifies the LayoutPreparationParams being referenced.

Rotate

One of the values in the following enumeration:

Rotate0 This is the default.

Rotate90 Rotate 90 degrees counter-clockwise.

Rotate180

Rotate270 Rotate 90 degrees clockwise.

**3.2.53 Screening Family**

Vendor specific screening family to use.

**Scope**

Job

**Conformance Levels**

2

**Processes**

Screening

**Syntactic Fragment**

```
<ScreeningParams ID=Reference ID Class="Parameter" Status="Available" >
  <ScreenSelector ScreeningFamily=vendor-specific screening family name />
</ScreeningParams>
```

**Type**

Resettable, single-valued

**Allowed Values**

ID

String of up to 255 characters that uniquely identifies the ScreeningParams being referenced.

ScreeningFamily

String of up to 255 characters.

**3.2.54 Settings Policy**

Indicates what happens when unsupported attributes, attribute values, or subelements of a resource or element are present.

**Scope**

N/A

**Conformance Levels**

1 and 2

### Processes

None

### Syntactic Fragment

To specify the same Settings Policy for all resources and elements:

```
<JDF ... SettingsPolicy="what to do" />
```

To specify a Settings Policy for a specific resource:

```
<Resource ... SettingsPolicy="what to do"
BestEffortExceptions="attributes that are BestEffort"
MustHonorExceptions="attributes that are MustHonor"
OperatorInterventionExceptions="attributes that are OperatorIntervention"
/>
```

To specify a Settings Policy for a specific element that is not derived from Abstract Resource and is not the Combined Digital Printing Node:

```
<Element ...
BestEffortExceptions="attributes that are BestEffort"
MustHonorExceptions="attributes that are MustHonor"
OperatorInterventionExceptions="attributes that are OperatorIntervention"
/>
```

### Type

Resettable, single-valued

### Allowed Values

SettingsPolicy

One of the values in the following enumeration:

BestEffort	Substitute or ignore unsupported attributes, attribute values, default attribute values, or elements and continue processing the job.
MustHonor	Reject the job when (1) any unsupported attributes, attribute values, or elements are present or (2) any omitted attributes have an unsupported default value defined in this specification.
OperatorIntervention	Pause the job and query the operator when (1) any unsupported attributes, attribute values, or elements are present or (2) any omitted attributes have an unsupported default value defined in this specification. If a device has no operator intervention capabilities, OperatorIntervention is treated as MustHonor.

If not specified for a resource or element, the value defined for the JDF node is used.

If not specified for the JDF Node, BestEffort is the default.

BestEffortExceptions

Optional. Strings (NMTOKENS) of up to 255 characters that are the attributes that are to be treated as BestEffort when SettingsPolicy is set to MustHonor or OperatorIntervention.

MustHonorExceptions

Optional. Strings (NMTOKENS) of up to 255 characters that are the attributes that are to be treated as MustHonor when SettingsPolicy is set to BestEffort or OperatorIntervention.

OperatorInterventionExceptions

Optional. Strings (NMTOKENS) of up to 255 characters that are the attributes that are to be treated as OperatorIntervention when SettingsPolicy is set to BestEffort or MustHonor.

### 3.2.55 Sides

Whether to print on one or two sides of the sheet.

#### Scope

Page, Document, or Job

#### Conformance Levels

1 and 2

#### Processes

LayoutPreparation

#### Syntactic Fragment

To print the entire document the same:

```
<LayoutPreparationParams ID="Reference ID" Class="Parameter"
  Status="Available" Sides="sides to print on" />
```

#### Type

Resettable, single-valued

#### Allowed Values

ID

String of up to 255 characters that uniquely identifies the LayoutPreparationParams being referenced.

Sides

One of the values in the following enumeration:

OneSidedBackFlipX	Image on back side of the media where the page content is placed such that binding can occur along the x-axis. This is equivalent to tumble.
OneSidedBackFlipY	Image on back side of the media where the page content is placed such that binding can occur along the y-axis.
OneSidedFront	Image on front side of the media. This is the default.
TwoSidedFlipX	Image on front and back sides of media where the page content is placed such that binding can occur along the x-axis. This is equivalent to tumble.
TwoSidedFlipY	Image on front and back sides of media where the page content is placed such that binding can occur along the y-axis.

### 3.2.56 Spot Color

Separation to render spot color.

#### Scope

Job

**Conformance Levels**

2

**Processes**

Interpreting

**Syntactic Fragment**

```

<ColorantControl ID=Reference ID Class="Parameter" Status="Available" >
  <ColorantParams>
    <SeparationSpec Name=separation name />
  </ColorantParams>
</ColorantControl>

```

**Type**

Resettable, single-valued

**Allowed Values**

ID

String of up to 255 characters that uniquely identifies the ColorantControl being referenced.

Name

String of up to 255 characters.

**3.2.57 Stapling and Stitching**

How specific pages, documents, or job should be stapled/stitched together.

**Scope**

Page, Document, or Job

**Conformance Levels**

?

**Processes**

Stitching

**Syntactic Fragment**

```

<StitchingParams ID=Reference ID Class="Parameter" Status="Available"
  NoOp=whether or not to stitch
  Angle=angle of stitch
  NumberOfStitches=number of stitches
  ReferenceEdge=which edge to stitch
  StitchType=position of stitches />

```

**Type**

Resettable, single-valued

**Allowed Values**

ID

String of up to 255 characters that uniquely identifies the StitchingParams being referenced.

NoOp

Optional. One of the following values:

true	To turn stitching off. All other StitchingParams attributes are ignored.
false	To use other StitchingParams attributes to specify stitching.

This is the default.

Angle

Optional if angle cannot be specified or can be defaulted. Integer that is the angle, in degrees, of each stitch; Increasing in the counterclockwise direction. Valid values are 0 through 359, where 0 is horizontal or parallel to the x axis. The default is system specified. If StitchType is Saddle, Angle is ignored.

NumberOfStitches

Optional if number of staples cannot be specified or can be defaulted. Integer in the range of 1 through 255. If NumberOfStitches is not present or is set to -1, the system specifies the number of stitches. The default is -1.

ReferenceEdge

Optional if reference edge cannot be specified or can be defaulted. One of the values in the following enumeration:

Bottom	Edge that is along the x-axis of the sheet.
Left	Edge that is along the y-axis of the sheet.
Right	Edge that is opposite of the Left edge.
Top	Edge that is opposite of the Bottom edge.
SystemSpecified	This is the default.

**ISSUE: The Coordinate Systems working group decided to replace all ReferenceEdge attributes with Orientation attribute in a new Intermediate ComponentLink**

**ISSUE: Should the StitchFromFront attribute be added to this ICS?**

StitchType

Optional if position of staple cannot be specified or can be defaulted. One of the values in the following enumeration:

Corner	Stitch in the corner that is at the clockwise end of the ReferenceEdge.
Saddle	Stitch on the middle fold which is on the saddle parallel to the reference edge.
Side	Stitch along the reference edge. If NumberOfStitches is 1, then the stitch is placed parallel to the ReferenceEdge and in the corner that is at the clockwise end of the ReferenceEdge.
SystemSpecified	This is the default.

**3.2.58 Start, Separator/Slip, End Sheets**

Whether start, separator/slip, and/or end sheets are to be included with the job. If any type of finishing (hole making, stitching, etc.) is specified for any part of the job, the start, separator/slip, and end sheets will not be included in the finishing.

**ISSUE: What if the desired behavior is that the start, separator/slip, or end sheet be finished with the job, job copy, document, or document copy? Can this be accomplished in JDF? Craig and Claudia talked about this in SJ and came up with a possible solution. The proposal is to add an attribute to InsertSheet called something like IncludeInBundleItem. IncludeInBundleItem has 3 possible values, After, Before, and None, where After specifies that the InsertSheet is to be included in the BundleItem that occurs after the InsertSheet. Before specifies that the InsertSheet is to be included in the BundleItem that occurs before the InsertSheet. None specifies that the InsertSheet is to be all alone in a new BundleItem.**

**6/10 - Yes add new IncludeBundleItem attribute to InsertSheet. Tom will add to JDF 1.2 Spec.**

**6/13 - Maybe "None" should be when the InsertSheet isn't to be included in any BundleItem and that "New" is when the InsertSheet is to be included in a new BundleItem. Then subsequent InsertSheets can be included in a bundle with an InsertSheet that was previously defined and put in its own**

**BundleItem. 8/22 - Decided to add another value called New to create a new bundle item for this insert sheet**

**Scope**

Job

**Conformance Levels**

2

**Processes**

LayoutPreparation

**Syntactic Fragment**

```

<Media ID="Media reference ID" Class="Consumable" Status="Available" ... />
<LayoutPreparationParams ID="Reference ID" Class="Parameter" Status="Available" >
  <InsertSheet SheetType="type of insert sheet to include"
    IsWaste="whether or not to discard separator sheets"
    SheetUsage="where to include separator sheet"
    SheetFormat="what it to appear on separator sheet" >
    IncludeInBundleItem="which bundle item to include this insert sheet in"
    <Sheet>
      <MediaRef rRef="Media reference ID" />
      <Surface Side="side to print UserText on">
        <MarkObject>
          <JobField ShowList="UserText"
            UserText="text to image on insert sheet"
          </JobField>
        </MarkObject>
      </Surface>
    </Sheet>
  </InsertSheet>
</LayoutPreparationParams>

```

**Type**

Resettable, multi-valued

**Allowed Values**

ID

String of up to 255 characters that uniquely identifies the LayoutPreparationParams being referenced.

IncludeInBundleItem

Optional. One of the following values:

- |               |   |
|---------------|---|
| <u>After</u>  | <u>InsertSheet is to be included in the BundleItem that occurs after the InsertSheet. If IncludeInBundleItem is set to After and a BundleItem is not defined after the insert sheet then IncludeBundleItem is ignored.</u>    |
| <u>Before</u> | <u>InsertSheet is to be included in the BundleItem that occurs before the InsertSheet. If IncludeInBundleItem is set to Before and a BundleItem is not defined before the insert sheet then IncludeBundleItem is ignored.</u> |
| <u>New</u>    | <u>InsertSheet is to start a new BundleItem.</u>  |
| <u>None</u>   | <u>InsertSheet is <i>not</i> to be included in a BundleItem. This is the default.</u>   |

ISSUE: Update above descriptions with final version.

IsWaste

Optional. One of the following values:

- |       |  |
|-------|--|
| false | Separator sheets are not to be discarded after the completion of the job.                  |
| true  | Separator sheets are to be discarded after the completion of the job. This is the default. |

SheetFormat

Optional. A string (NMTOKEN) of up to 255 characters. Some typical values are as follows:

Blank	Content of Separator sheet is to be blank.
Brief	Content of Separator sheet is the brief version.
Full	Content of Separator sheet is the full version.
Standard	Content of Separator sheet is the standard.
SystemSpecified	Content of Separator sheet is to be defined by the system. This is the default.

#### SheetType

One of the values in the following enumeration:

JobSheet	Include a sheet that delimits the job
SeparatorSheet	Include a sheet that delimits each copy of each document in the job.

The other values for SheetType that are included in the JDF Specification Release 1.2 are not used when specifying the Start, Separator/Slip, End Sheet feature.

#### SheetUsage

One of the values in the following enumeration:

Trailer	Include an insert sheet at the end of the job when SheetType=JobSheet or at the end of each copy of each document in the job when SheetType=SeparatorSheet.
Slip	Include an insert sheet between each copy of each document in the job when SheetType=SeparatorSheet .
SlipCopy	Include an insert sheet between copies of the job when SheetType=SeparatorSheet.
Header	Include an insert sheet at the start of the job when SheetType=JobSheet or at the start of each copy of each document in the job when SheetType=SeparatorSheet.

The other values for SheetUsage that are included in the JDF Specification Release 1.2 are not used when specifying the Start, Separator/Slip, End Sheet feature.

#### Side

One of the values in the following enumeration:

Back
Front

#### ShowList

Always set to UserText.

#### UserText

String of up to 255 characters that is to appear on the Separator Sheet.

### 3.2.59 Trapping

Whether trapping is to be performed or not.

ISSUE: In SJ we talked about just having a basic turn trapping on or off feature for the first version of the ICS for DP. So here is basic trapping. Is this still what we want to do?  
6/10 - Yes

#### Scope

Job

#### Conformance Levels

2

#### Processes

Trapping

#### Syntactic Fragment

```
<TrappingDetails ID="Reference ID" Class="Parameter" Status="Available"
  NoOp="whether or not to perform trapping"
  Trapping="whether of not to perform trapping"
/>
```

**Type**

Resettable, single-valued

**Allowed Values**

ID

String of up to 255 characters that uniquely identifies the TrappingDetails being referenced.

NoOp

Optional. One of the following values:

true	To turn trapping off. All other TrappingDetails attributes are ignored.
false	To use other TrappingDetails attributes to specify trapping. This is the default.

Trapping

Optional. One of the following values:

true	Enable trapping.
false	Disable trapping.

If not specified system specified trapping is performed.

ISSUE: Are both NoOp and Trapping needed since Trapping itself can turn trapping on and off? But we are trying to use NoOp to turn functionality off?

6/10 - Seems like this should work. Tom will add an issue to the JDF 1.2 spec.

6/20 - TSC decided that both NoOp and Trapping are needed.

**3.2.60 Trimming**

Perform a trimming operation on the final product.

**Scope**

Page, Document, or Job

**Conformance Levels**

2

**Processes**

Trimming

**Syntactic Fragment**

```
<TrimmingParams ID="Reference ID" Class="Parameter" Status="Available"
  NoOp="whether or not to trim" TrimmingType="SystemSpecified" />
```

**Type**

Resettable, single-valued

**Allowed Values**

ID

String of up to 255 characters that uniquely identifies the TrimmingParams being referenced.

NoOp

Optional. One of the following values:

true	To turn trimming off. All other TrimmingParams attributes are ignored.
------	--



false To use other TrimmingParams attributes to specify trimming.  
This is the default.

#### ~~TrimmingType~~

~~Always set to SystemSpecified to perform trimming operation.~~

~~The following values for TrimmingType are included in the JDF Specification Release 1.2, but have intentionally been excluded in the ICS for Digital Printing:~~

~~Detailed~~

~~ISSUE-6/10—Propose to deprecate TrimmingType and use NoOp to turn trimming on and off—Tom will add an issue to the JDF-1.1 spec.~~

Figure 3-8 shows an example Logical Diagram for a JDF Controller and a JDF Device. You MAY want to use this diagram with appropriate modifications or create your own diagram. The downward arrow shows the Controller sending a JMF request to the Device. The connecting upward arrow shows the Device returning a Response to the Controller. And the other vertical arrow shows the Device sending an Event Notification to the Controller. The dashed curved line suggests that the Response follows immediately after the Request in time. The UI Interface is not specified very much at all, except that the Controller SHOULD localize User Output and display GUI information for User Input. The dashed bi-directional arrow between the Device and the Local Operator indicates information flow in both directions using an unspecified mechanism.

**ISSUE:** Add versions of the following two figures to Chapter 1 that are tailored for the ICS for DP. Tom will recreate these diagrams using Words drawing tool. Tom H. will provide a template in the base ICS for us to copy and modify for the ICS for DP.

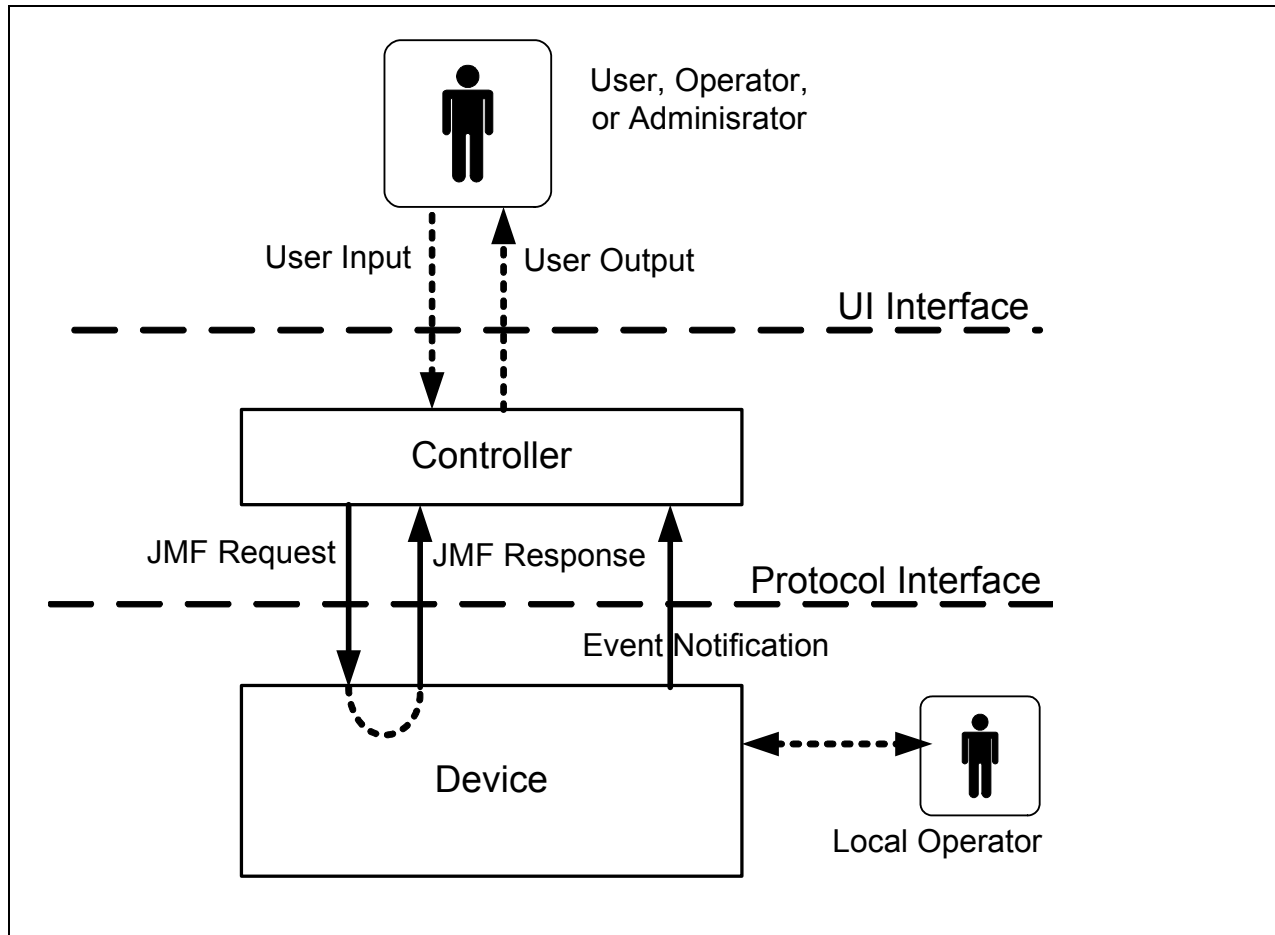


Figure 3-8 - Logical Producer - Consumer Diagram

Figure 3-9 and Figure 3-10 are example Configuration Diagrams. You MAY want to use these diagrams with appropriate modifications or create your own diagram. The vertical dashed line labeled “Interface” represents the Interface(s) in each configuration between Controller and Device. The vertical dashed line labeled “UI” represent the User Interface between the User and the Controller. The single headed solid arrow labeled JMF represents a JMF request and response. The arrow labeled “?” represent any sort of communication which may be uni-directional or bi-directional.

From the diagrams, a Controller can directly control a Device as in Configurations A-C (see Figure 3-9) or can be a Slave Controller as in Configurations D-F (see Figure 3-10). A Slave Controller MUST conform to both (1) the Device Interface in accepting requests and (2) the Controller Interface in order to control Device(s). A Device MUST conform to the Interface to a Slave Controller and to a Controller (see Figure 3-10).

Your ICS MUST state any slight differences in Interface Conformance Requirements for the Interfaces between (1) a Controller and a Slave Controller, (2) a Slave Controller and a Device, and (3) a Controller and a Device. For example, a Slave Controller might generate the Job Id when it submits a job to the Device, instead of the Device assigning the Job Id when it receives the job. So the Device MUST be able to accept the Job Id assigned by the Slave Controller when the Device is configured to accept jobs from the Slave Controller. On the other hand, the Device might have to generate a Job Id when accepting jobs directly from a Controller that isn’t a Slave Controller. Or perhaps both the Slave Controller and the Device always generate their own Job Ids for all jobs.

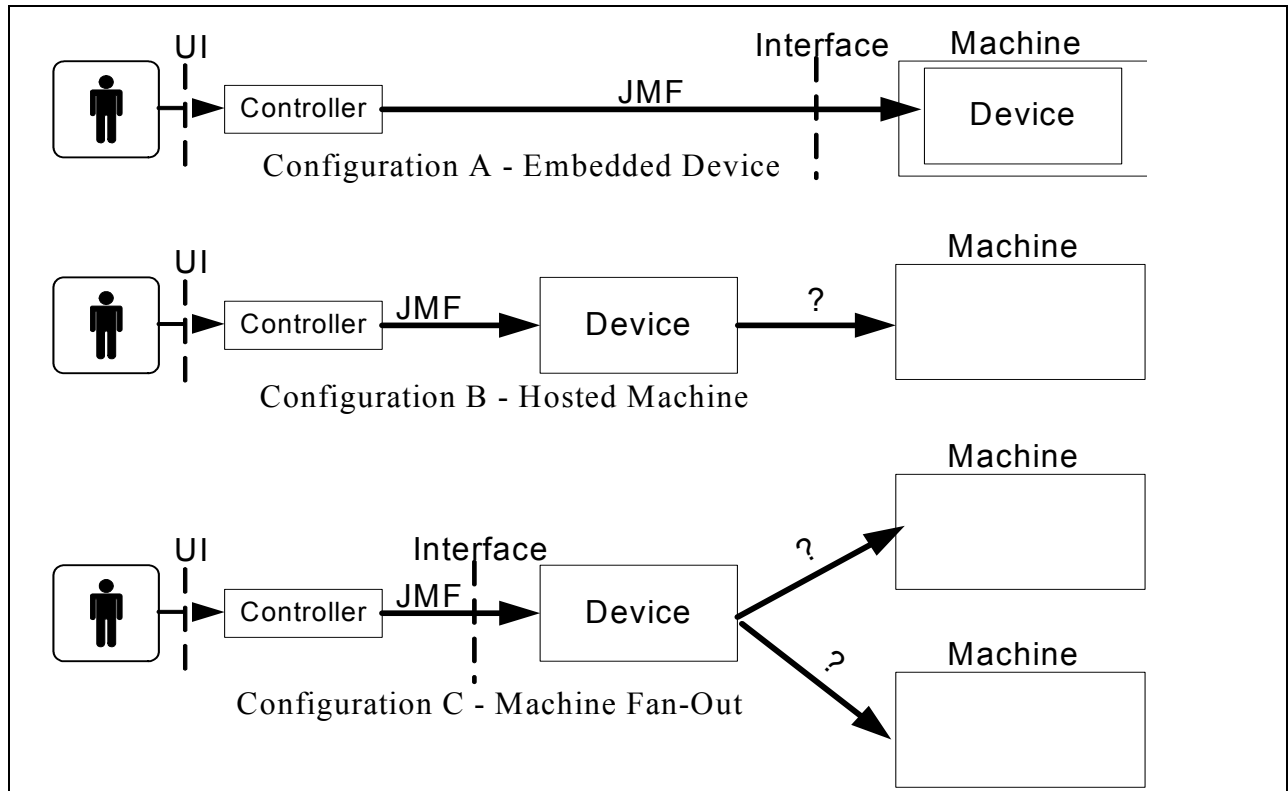


Figure 3-9 - Configurations of a Controller controlling a Device

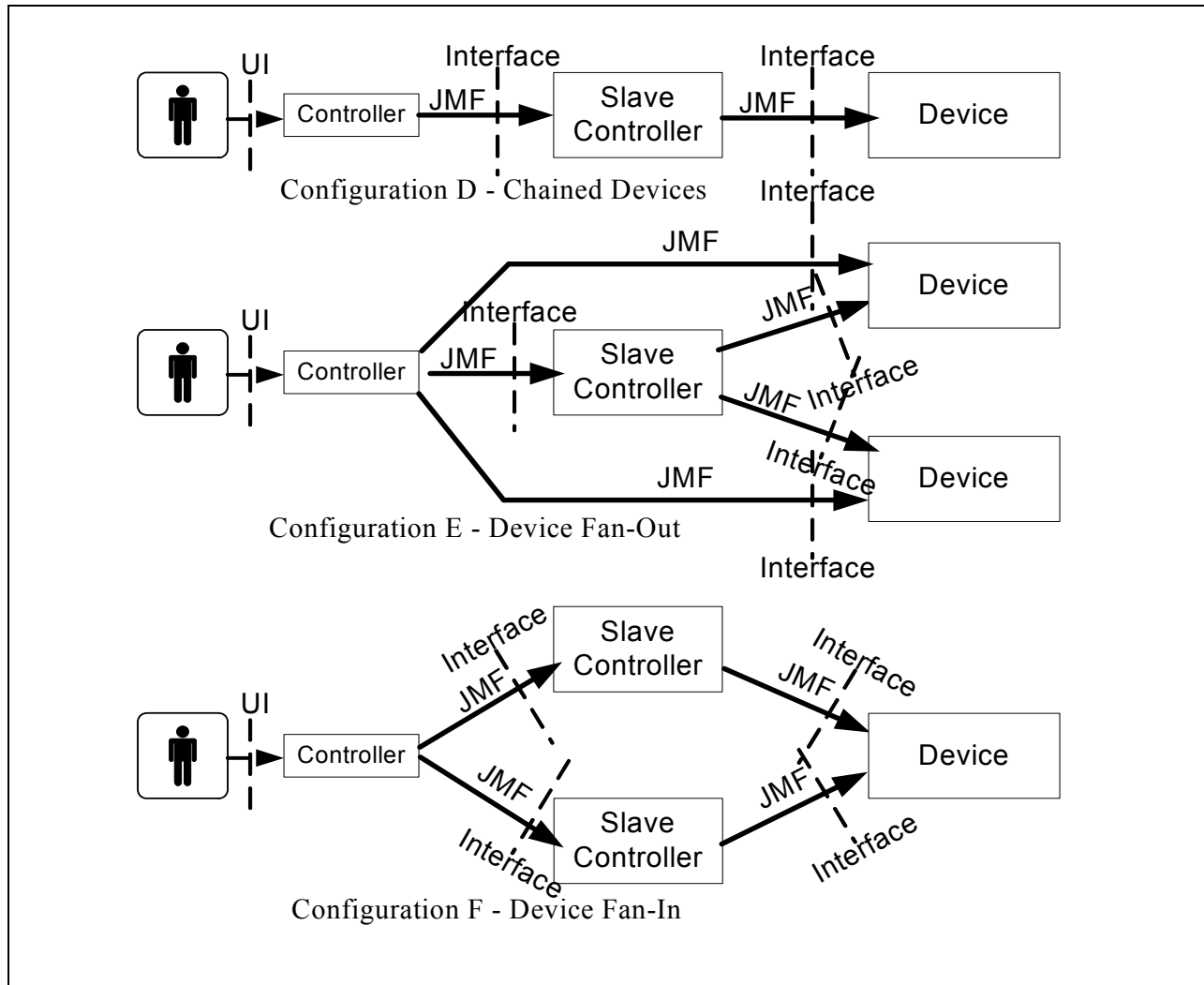


Figure 3-10 - Configurations of a Controller controlling Slave Controllers controlling Devices

### 3.3 Conformance Level(s), Modules, and/or Profiles

Your ICS **MUST** specify a minimum Conformance Level for the chosen Interface(s), that is, a minimum number of Features that a Consumer implementation **MUST** support in order to claim conformance to an Interface covered in the ICS. Each ICS **MUST** also specify a set of Features that the Producer implementation **MUST** be able to invoke when the Producing User wishes to invoke the Feature. In other words, if one vendor sells a JDF Producer that conforms to a given Interface in your ICS and another vendor sells a JDF Consumer for the same Interface in the ICS, a Customer buying the two products can expect them to interoperate, at least at the minimum Conformance Level. Furthermore, the interoperability **MUST** require no prior agreements between the Vendors, except to conform to a particular Interface in the ICS.

*See [ICS-std] for more details and tradeoffs between Levels, Modules, and/or Profiles. For example, your ICS **SHOULD** have a Level, Module, or Profile that **REQUIRES** support for the Capability mechanism in JDF/1.1a (see [JDF] section 7.3).*

*For each Conformance Level, Module, and Profile, your ICS **MUST** specify which Process, Resources, Attributes and Values are **REQUIRED** for a Consumer to support and for a Producer to be capable of including if the Producing User wishes to invoke the Feature. Without such specification, interoperability will not be achieved.*

*Your ICS MUST specify ranges and sizes of Features supported. See [ICS-std] for more details.*

*You MUST decide whether to burden the Consumer or the Producer to support two similar Features, in order to achieve interoperability. For example, REQUIRING a Producer versus a Consumer to support both JMF and Hot Folders in order to achieve interoperability. Or REQUIRING a Producer versus a Consumer to support both PDL x and PDL y in order to achieve interoperability. See [ICS-std] for tradeoffs between Producer versus Consumer requirements to support two ways to do the same or similar things.*

## 4 Conformance Requirements for <fill in your ICS>

*[You MUST include a conformance section that is a summary list of the important conformance requirements for Consumers and Producers of your Interface.*

*You SHOULD reference each major section in your document that contains conformance statements. Such a listing is a useful checklist for the implementer of the conformance requirements they need to follow without repeating the conformance statements themselves (which could get out of synch).*

*If your Interface has a large number of Feature that Producers or Consumers MAY support, you SHOULD attempt to group them into Conformance Levels, Modules, and/or Profiles (see [ICS-std]).*

*The Conformance section MUST make it clear which statements apply to Producers and which apply to Consumers and SHOULD list them in two separate sub-sections. Usually, such clarity is easiest by having separate sub-sections, one for Producers and one for Consumers. Such separation also simplifies an implementer's job of going over the list of conformance requirements that apply to the implementation, since most implementers are implementing either a Producer or a Consumer, but not both.]*

See section 1.4.2.4 Conformance Requirements for Support of Combined Processes of the JDF Specification Release 1.2. This section specifies the conformance requirements for <fill in Producers and Consumers of your Interface>.

### 4.1 Conformance Requirements for <fill in Producers of your Interface>

*[You MUST list the conformance requirements for the Producer of your Interface as one more Conformance Levels, Modules, and/or Profiles.]*

A <fill-in Producer> of this JDF Interoperability Conformance Specification:

1. MUST ...
2. SHOULD ...
3. MAY ...

### 4.2 Conformance Requirements for <fill-in Consumers of your Interface>

*[You MUST list the conformance requirements of a Consumer of your Interface as one more Conformance Levels, Modules, and/or Profiles.]*

A <fill-in Consumer> of this JDF Interoperability Conformance Specification:

1. MUST ...
2. SHOULD ...
3. MAY ...

## 5 References

*[You MUST list the references that are required in order to conform to your ICS.]*

*You MUST include references inside square brackets ([]), so it is clear that you are making a reference and that the reader can look up the reference in one of the two References sections. It is acceptable to talk about a referenced document using the [] notation directly. For example, “The Slave Controller is defined in [JDF] section 2.1.2.4.”*

*You SHOULD use mnemonic labels that are suggestive of the reference. Then the reader can remember them and doesn’t need to keep referring back to this References section. However, once a standard has been assigned a number, you MUST refer to that specification in references using some short prefix and the number. The prefix indicates the source of the numbering. For example, “rfc” indicates an IETF specification, “pwg” indicates an ISTO-IEEE specification, “iso” indicates an ISO standard, “ieee” indicates an IEEE standard, “icc” indicate an ICC standard or registration, etc.*

*Note: The IETF recommended practice [IETF-TEM] is for two top level sections. However, combining them into one with two sub-sections is also acceptable, which we have done here.]*

CIP4 documents are available at <http://www.cip4.org>. RFCs are available at: <http://www.rfc-editor.org/rfcsearch.html>. Internet-Drafts are available at: <http://www.ietf.org/ID.html>. IEEE-ISTO PWG standards are located at: <http://www.pwg.org> or more specifically at: <http://www.pwg.org/ipp/index.html>.

### 5.1 Normative References

*[You MUST list references that contain conformance requirements that also apply to a Producer and/or Consumer of your Interface whether they be REQUIRED, RECOMMENDED, or OPTIONAL. You MUST include the Editors’ names, but only their family name (usually last names). Using family names only avoids mistaken initials for nick names, improves readability, and avoids putting family names in the wrong order in cultures where the family name is normally written first. Normative References MUST be at the same level of approval as your ICS.]*

*Here are some example references that your ICS might reference:]*

[ASCII]

Coded Character Set - 7-bit American Standard Code for Information Interchange (ASCII), ANSI X3.4-1986. This standard is the specification of the US-ASCII charset.

[IANA-CS]

IANA Registry of Coded Character Sets: <ftp://ftp.iana.org/in-notes/iana/assignments/character-sets>

[IANA-MT]

IANA Registry of Media Types: <ftp://ftp.iana.org/in-notes/iana/assignments/media-types/>

- [IEEE-ISTO5101.1]  
PWG Standard for Standardized Media Names, February 2002.
- [JDF]  
CIP4 Job Definition Format (JDF), JDF/1.2, September 2003.
- [JDF-URL]  
[JDF] Appendix A "Application Note on File URL in JDF", September 2003.
- [RFC1766]  
Alvestrand, "Tags for the Identification of Languages", RFC 1766, March 1995.
- [RFC1951]  
Deutsch, "DEFLATE Compressed Data Format Specification version 1.3 ", RFC 1951, May 1996.
- [RFC1952]  
Deutsch, "GZIP file format specification version 4.3", RFC 1952, May 1996.
- [RFC1977]  
Schryver, "PPP BSD Compression Protocol", RFC 1977, August 1996.
- [RFC2045]  
Fried, Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies " RFC 2045, November 1996.
- [RFC2046]  
Freed and Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types." November 1996. (Obsoletes RFC1521, RFC1522, RFC1590), RFC 2046.
- [RFC2048]  
Freed, Klensin, and Postel, "Multipurpose Internet Mail Extension (MIME) Part Four: Registration Procedures". RFC 2048, November 1996.
- [RFC2616]  
Fielding, Gettys, Mogul, Frystyk, Masinter, Leach, and Berners-Lee, "Hypertext Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.
- [RFC2228]  
Horowitz, S. Lunt, "FTP Security Extensions", RFC 2228, October 1997.
- [RFC2277]  
Alvestrand, "IETF Policy on Character Sets and Languages" RFC 2277, January 1998.
- [RFC2278]  
Freed, Postel: "IANA CharSet Registration Procedures", RFC 2278, January 1998.
- [RFC2279]  
Yergeau , "UTF-8, a transformation format of ISO 10646", RFC 2279. January 1998.

- [RFC2316]  
Bellovin , "Report of the IAB Security Architecture Workshop", RFC 2316, April 1998.
- [RFC2396]  
Berners-Lee, Fielding, Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.
- [RFC2617]  
Franks, Hallam-Baker, Hostetler, Lawrence, Leach, Luotonen, Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.
- [RFC2717]  
Petke and King, "Registration Procedures for URL Scheme Names", RFC 2717, November 1999.
- [RFC2910]  
Herriot, Butler, Moore, and Turner, "Internet Printing Protocol/1.1: Encoding and Transport", RFC 2910, September 2000.
- [RFC2911]  
deBry, Hastings, Herriot, Isaacson, and Powell, "Internet Printing Protocol/1.1: Model and Semantics", RFC 2911, September 2000.
- [SSL]  
Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.

## 5.2 Informative References

*[You MUST list additional references that will help the reader understand the subject or a work from which the current work was derived, but that does not contain REQUIRED, RECOMMENDED, or OPTIONAL conformance requirements for Requester or Providers of the Interface in your specification. You MUST include the Editors' names, but only their family name (usually last names). Using family names only avoids mistaken initials for nick names, improves readability, and avoids putting family names in the wrong order in cultures where the family name is normally written first. Informative References MAY be still works in progress, in which case they MUST be labeled "work in progress".]*

- [IANA-CON]  
Narte, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.
- [ICS-std]  
Hastings, "Standard for Writing a JDF Interoperability Conformance Specification (ICS)", version 1/0, 30 January 2003. See file: Standard-for-ICS-v1.0.doc, .pdf
- [ICS-tem]  
Hastings, "Template for a JDF Interoperability Conformance Specification (ICS)", version 1.0, 30 January 2003. See file: Template-for-ICS-v1.0.doc, .pdf
- [mem-guide]  
Daun, Guide for CIP4 Members, *CIP4 Secretariat / Fraunhofer IGD*



## 6 Registration Considerations

*[Often extensible Interfaces include a means for an implementer to register the extensions with a registration authority who then published the extensions in a public place. This section MUST include any registration procedures or indicate that there are none.]*

Each JDF Conformance Subset Standard MUST register its subset with CIP4 so that it may be uniquely identified in Vendor literature and in the Interface when needed. If there are levels or OPTIONAL modules, each level and/or module MUST have a distinct registration identifier.

## 7 Internationalization Considerations

*[You MUST indicate whether your Interface caters to internationalization, that is regional character sets and/or natural languages.]*

An ICS MUST give any internationalization considerations that implementers of Producers and/or Consumers MUST, SHOULD, or MAY take into account when implementing.

## 8 Security Considerations

*[You MUST include considerations that an implementer of a Producer or Consumer needs to take into account concerning security. Security includes authentication, authorization, and privacy. Authentication is when a Consumer is able to verify that a Producer is who he says he is. Authorization is comparing an authenticated identify with a list of allowed Consumers. Privacy is ensuring that the exchange has not been modified unbeknownst to the Producer and Consumer. For example, include Denial of Service attacks, Man in the Middle attacks, and Trojan Horse scenarios that might occur if the implementer doesn't consider them.]*

A JDF Conformance Subset Standard MUST give any security considerations that implementers of Producers and/or Consumers MUST, SHOULD, or MAY take into account when implementing.

## 9 Contributors

*[You MAY include this section contain contributors who deserve significant credit for the document, including their affiliation. Then the list of authors can be just the person or people who did the actual writing.]*

William Smith - ACME Corporation

## 10 Acknowledgments

*[You MAY include this section which contains acknowledgement of individuals, including their affiliation, instead of, or in addition to, the Contributors section. Acknowledgements SHOULD include an explanation of the contribution of each individual.]*

Much of the experience gained in writing interoperable standards came from writing the IETF Internet Printing Protocol (IPP) standards [IPP], testing the products and improving the standards. The IPP development included three interoperability testing events where developers of IPP Consumers and Producers connected their client and Printer products on an intranet in the same room over a three day period. See <http://www.pwg.org/pub/pwg/ipp>.

At the second interoperability testing event, there were 11 IPP Client implementations and 27 IPP Printer implementations. Out of the 297 possible Client/Printer combinations, 296 were successful in completing a simple "Print-Job" operation. That is a success rate of 99.66%. At the first testing event, there were 8 Clients and 16 Printer with a success rate of 96.9%.

At the third testing event, 18 participants provided 17 IPP Printers, 9 IPP Clients, 2 firewalls and 2 HTTP Proxies. Out of the 153 possible combinations of Clients and Printers, 151 were tested.

- The overall success rate was 93%.
- Limiting the scope to IPP v1.1 provided a success rate of 96%.
- With IPP v1.0 Clients and v1.1 Printers, the success rate was 100%.
- The tests with v1.1 Clients and v1.0 Printers resulted in a success rate of 31%, which is not surprising given that some printer implementations explicitly disallowed that combination. Some Clients were able to retry in v1.0 mode raising the success rate to 69%. It should also be noted that for v1.0 Printers that allow v1.1 Clients the success rate was 100%.

## 11 Author's Address

***[You MUST include the list of authors, including address, phone number, and email. Only include people who performed significant writing of the document. Include reviewers in the Contributors or Acknowledgments sections. For example:***

Tom Hastings  
Xerox Corporation  
737 Hawaii St.  
El Segundo, CA 90245

Phone: 1+310 333-6413  
Fax: 1+310 333-5514  
e-mail: [hastings@cp10.es.xerox.com](mailto:hastings@cp10.es.xerox.com)

## **Annex A <fill-in a title for Annex> (Normative)**

*[You MUST indicate for each Annex whether it is Normative or Informative, by including and “(Normative)” and “(Informative)” to the heading line.]*

## **Annex B <fill-in a title for Annex> (Informative)**

*[You MUST indicate for each Annex whether it is Normative or Informative, by including and “(Normative)” and “(Informative)” to the heading line.]*

## Annex C *Tips for Good Technical Writing (Normative)*

This annex contains some tips for good technical writing that you SHOULD follow. Delete it after a few versions of your document:

### C.1 Test each sentence for understandability

Test each sentence. Be your own Devil's Advocate against each sentence. If there are no ways or more than one way to understand the sentence and its implication, re-work the sentence until there is only one way to understand it. Are any special terms used but not defined? Do any terms have several meanings, so it not clear to the reader which meaning you intend? If so, you need to define your term in the Terminology section up front.

Follow the "SMART" sentence testing criteria when writing and reviewing each sentence:

1. **Specific** - Is the sentence clear, unambiguous, and open to only one interpretation?
2. **Measurable** - Is there a quantitative way of determining if the requirement is satisfied? Could be through an actual measurement, an enumeration, inspection, or any number of other means.
3. **value Adding** - Is this requirement something that adds value? It could be specific, measurable, real and testable, and not make any positive difference to the actual Interface.
4. **Real** - Is this a real requirement, or just someone's opinion or pet issue? We want to make sure Interface requirements are validated against some customer need.
5. **Testable** - If you were to test this, how would you do it? If you know in advance how you'll verify you got the Interface requirement right.

### C.2 Avoid the passive voice

Avoid the passive voice. This tip is a special case of tip #C.1. That is, instead of writing: "The job is deleted" you should write: "The server deletes the job." The active voice is more specific and absolutely necessary for a specification that talks about both a Producer (client) and a Consumer (server). Otherwise, a reader could misunderstand and think that it was the client that deletes the job. Its even more critical when the sentence contains a conformance term, such as "The job **MUST** be deleted". Who **MUST** delete the job? The client or the server?

### C.3 Be careful of the word "or"

Be careful of the word "or". This tip is another special case of tip #C.1. The word "or" means a choice for someone or something. Does it mean a choice for the implementor or of the implementation.

- Consider the sentence: "The Device **MUST** return a or b." Who gets to choose? Must the implementer support returning both and the implementation chooses which one based on circumstances, or **MAY** an implementer choose to always return one fixed value?
- Consider the sentence: "The Device **MAY** return a or b." This sentence is even worse, because it could mean that a conforming implementation doesn't have to return either value.

### C.4 Distinguish between "support" versus "send" for Producer

For Producers, clearly distinguish between "MUST support xxx", that is the Producer contains code that emits xxx across the Interface to the Consumer on some occasions, and "MUST send xxx", that is the Producer emits xxx across the Interface to the Producer on *all* occasions.

### C.5 Avoid the use of pronouns

Avoid the use of third person pronouns, such as “he”, “she” “it”, or “they”. Instead, use the noun that is intended. Otherwise, the reader isn’t sure whether the pronoun refers to the subject or object of the previous phrase or sentence. Also it slows down the reader if the reader attempts to determine which you meant. For example, don’t write: “Producers MUST accept all instances. They NEED NOT conform.” Does “They” refer to instances or Producers? The second sentence is ambiguous and the reader is slowed down trying to figure out what you meant or worse, misinterprets the sentence. Instead, write: “Producers MUST accept all instances. Instances NEED NOT conform.” As another example: “The Producer MUST shut down the Consumer. Then it MUST un-register.” Who MUST un-register”? The Producer or the Consumer? Just re-state the antecedent, whichever you intend the sentence to apply, instead of using “it”.

## C.6 Include the “why” justifying “what” statements

Include the “why” justifying “what” statements for all of the following reasons:

1. Including “why” helps the reader to understand the “what”.
2. Also the reader is much more likely to remember the “what” and follow the standard, if they understand the “why”.
3. Furthermore, writing down an explicit “why” forces you to reveal the reasons for such a decision. If you can’t articulate a “why”, perhaps you didn’t have a good reason for the “what” and there are other alternatives that you should consider.
4. Finally, being explicit about your “why” will allow reviewers to see if they agree with your reasoning about your “what” or whether there are other better “whats”.

The “why” information can be included as a phrase (“..., in order to ...”), a parenthetical remark, a “Note: ...”, or information in an Informative Appendix. Some standards use the notation: “Rationale: ...”, instead of “Note: ...” so that it is clear that the statement is explaining why a decision was made and isn’t a conformance requirement. For example, consider the following sentence in section **Error! Reference source not found.**: “You MUST capitalize first letter of each word in your terms throughout the document, so that the reader knows that you are referring to the specially defined term.” This sentence explains both the “what” (capitalize) and “why” (so that the reader can distinguish the special terms from ordinary use of the words).

The following is a real world example of the mistakes that can be made because a why wasn’t explicitly written down so that reviewers could consider alternatives. In IPP/1.0 we decided to make the Print-Job operation REQUIRED for Printers to support and the Create-Job operation OPTIONAL. The unspoken “why” was that Print-Job creates a single document Job and Create-Job, in combination with a separate Send-Document operation for each document, creates a multiple document Job; we all agreed not to REQUIRE Printers to support multi-document jobs. However, the down side to our IPP/1.0 decision is that the client has to send the data as part of the Print-Job operation so that if the Printer rejects the Print-Job operation, the client still winds up wasting time sending the document anyway. In IPP/1.1 we merely allowed a Printer to support Create-Job and Send-Document for single document jobs and provided a means for the client to determine whether or not additional Documents would be accepted with additional Send-Document operations for the Job. Had we written down the “why” for Print-Job being REQUIRED and Create-Job being OPTIONAL, IPP/1.0 reviewers would have pointed out the mistake allowing us to fix the problem in IPP/1.0 instead of having to wait for IPP/1.1.

## C.7 Don’t specify conformance in a Notes or Informative Annexes

Don’t use MUST, SHOULD, and MAY conformance terminology in a “Note: ...”, “Rationale: ...”, or Informative Annexes that is explaining the “why” of conformance statements.

## C.8 Use lots of examples

Use lots of examples. Examples allow your reader to test his understanding of statements. In fact, for many readers, if they cannot construct an example in their minds, they do not understand a general statement. If you give the reader an example, you are helping him along the road to understanding. Also examples help you test that your statements are understandable, rigorous, and correct. If you can't generate an example from your statement or there might be unintended examples generated from your statement, you'll know your statement needs more work.

Here's a real example where the failure to include examples resulted in mis-understanding the descriptions. The Printer MIB [RFC1759] has the following three attributes for Interpreters in the Printer:

**prtInterpreterLangLevel:** The level of the language which this interpreter is interpreting or emulating. This might contain a value like '5e' for an interpreter which is emulating level 5e of the PCL language. It might contain '2' for an interpreter which is emulating level 2 of the PostScript language. Similarly it might contain '2' for an interpreter which is emulating level 2 of the HPGL language."

**prtInterpreterLangVersion:** The date code or version of the language which this interpreter is interpreting or emulating."

**prtInterpreterVersion:** The date code, version number, or other product specific information tied to this interpreter. This value is associated with the interpreter, rather than with the version of the language which is being interpreted or emulated."

The **prtInterpreterLangLevel** attribute had examples, and so is used by most implementations consistently, but the **prtInterpreterLangVersion** and **prtInterpreterVersion** is used very inconsistently by various implementations, if it is used at all, since no examples were included.

## C.9 Avoid repeating the same conformance statements

Avoid repeating the same conformance statements in various parts of the specification, since you may forget to update all occurrences, if you improve or change one occurrence of the statements. Instead, use cross-references generously to avoid repetition. Use MS-WORD cross-references, so that they will always be correct, even after sections are added or removed. If the cross-references are inserts as a hyperlink in MS-WORD then they will also be hot links in the PDF file, if you use Adobe™ PDFMaker (which uses Distiller) rather than using PDFWriter. With hot links, the reader (whether reading the .doc or the .pdf file) will be able to jump to your cross references.

## C.10 Use “which” versus “that” properly

Use “which” versus “that” properly. A “which” phrase is a parenthetical clause which is giving additional information but is *not* restricting the class of items, while “that” is restricting the class of items. You can easily test whether a phrase is restrictive or non-restrictive: If removing the phrase does not affect conformance, then the phrase is a which phrase and SHOULD be set off with commas to indicate that it is a parenthetical remark. On the other hand, if removing the phrase would affect conformance, then the phrase is a that phrase and MUST NOT be set off by commas, since it is not parenthetical. For example, “A Consumer MUST reject a JDF that is non-conforming” is a proper use of “that”, since not all JDFs are non-conforming. An another example, “It is important to conform to JDF, which is a new standard” is a proper use of “which”, since this example isn't restricting the class of JDF.

## C.11 Avoid the vague word “use”

Avoid the vague word “use”. Use an action verb, especially if it implies a direction across the Interface you are specifying, such as “send” or “include” for a Producer and “support” or “accept” for a Consumer.

## C.12 Avoid vague sentences

Avoid vague sentences. For example, instead of the following vague sentence in ISO DPA 10175: “If no distinction is specified, it is treated as both logical-and-physical.” be much more specific: “If the printer-realization attribute of a print object is missing, the server that manages the print object **MUST** treat the printer as if the printer-realization attribute has the value "logical-and-physical.”

## C.13 Additional general good writing tips

1. Avoid long sentences. When in doubt, break it up.
2. Don't leave out the definite and indefinite articles (the, a/an). Don't write: “The new mechanism **MUST** allow for subsequent addition of new members.” Instead, write: The new mechanism **MUST** allow for *the* subsequent addition of new members.
3. Don't end sentences with prepositions (with, for, to, at), unless the result is really stilted. For example, don't write: “I don't like the group which you belong to.” Instead, write: “I don't like the group to which you belong.” On the other hand, the following is correct, but very stilted: “Ending sentences with a preposition is a practice up with which I will not put.” On the other hand, recasting such an awkward sentence to make it simple is often the best solution. So instead write, “I will not put up with ending sentences with a preposition.”
4. Don't change subjects in mid paragraph.
5. Always spell check your document before sending it out. Build a custom dictionary while you type, so that any special terms are included in the spell checking with the correct spelling.
6. Always proof-read your document before sending it out for review.
7. Don't begin sentences with "This" as a noun referring to some unstated thing in the previous sentence. Only start sentences with “This” as an adjective with the noun immediately following. For example, "This section describes ..." is okay, but "This means ..." is not. Also avoid starting sentences with "There are".
8. Use adjectives to qualify nouns and thus be more specific. On the other hand, squeeze out excess verbiage that doesn't add any meaning; for example, question phrases like "are to be" and "is used to".
9. Instead of using “i.e.,” and “e.g.,” which mean that is and such as, respectively, use “that is” and “such as” (or “for example”), respectively. Many people don't understand the subtle distinction between the two. The former means “the items are the exhaustive list and no other values are allowed” and the latter means “the items are examples but are not an exhaustive list so that other values are allowed as well”.
10. When two or more items are listed and then a parallel list is included, but each is intended to be pair-wise related, add the phrase: “, respectively. For example, Instead of using “i.e.,” and “e.g.,” use “that is” and “such as” (or “for example”), respectively.



## **Annex D Change Log (Informative)**

### **D.1 Changes to make version 0.01, January 31, 2003**

Initial version.

### **D.2 Changes to make version 0.02, March 2, 2003**

1. Changed definitions for Scope, Syntax, Type, and Allowed Values in Section 3 to be in table format. Added Processes to table.
2. Added Processes heading and appropriate process(es) for each function in section 3.
3. Changed Binding function to use SpineTapingParams and CoverApplicationParams instead of BindingParams. Removed ISSUE.
4. Changed Comment/Description of Job function to use Name="Description" instead of "JobDescription" and to not be under RunList. Removed ISSUE.
5. Added Approver as value for ContactTypes in Contact Information function.
6. Removed DescriptionType="FoldProc", From, To, and Travel attributes from Folding function so that the only folding that is possible is by referring to a specific value for FoldCatalog.
7. Renamed Start Page function to Force Page.
8. Added AnySmallFormat, AnyLargeFormat, BypassTray-*n*, Envelope-*n*, InsertTray, InsertTray-*n*, LargeCapacity-*n* as values for the Input Tray Name function's LocationName attribute. Removed ISSUE.
9. Changed Interleave Sheets function's InsertSheet element to reside under LayoutPreparationParams instead of RunList. Removed SheetFormat ISSUE.
10. Changed Job Ticket Comment or Description function to be Job Ticket Template Comment or Description. Changed Name="Description" to Name="TemplateDescription". Removed ISSUE.
11. Changed Job Ticket ID function to Job Ticket Template ID. Changed Syntax to use JDF node TemplateID attribute instead of Comment Name="JobTicketID". Removed ISSUE.
12. Removed Job Ticket Name function.
13. Changed Job Ticket Version function to Job Ticket Template Version. Changed Syntax to use JDF node TemplateVersion instead of Comment Name="JobTicketVersion". Removed ISSUE.
14. Removed DescriptiveName attribute from Media function and added DimensionName attribute. Renamed PercentRecycled attribute to RecycledPercentage. Removed ColorName and added MediaColor. Removed ColorPool resource. Removed sRGB attribute. Removed ColorName ISSUE.
15. Changed Message To Operator function to use Comment Name="OperatorText" instead of "MessageToOperator". Removed ISSUE.
16. Removed Main and Upper values from Output Bin Name function OutputBin attribute. Removed ISSUE.

17. Removed ProofingParams from Printer Resolution function.
18. Removed SystemSpecified ISSUE from Process Color Model function.
19. Changed Proof Print function to use RenderingParams instead of ProofingParams. Removed ProofType attribute. Removed ContactType ISSUE. Removed ISSUE about Origination and Prepress working group revamping ProofingParams.
20. Removed HoleType ISSUE from Punching and HoleMaking function.
21. Added SlipCopy value to SheetUsage attribute in Start, Separator, and End Sheets function. Removed Slip ISSUE.

### **D.3 Changes to make version 0.03, March 17, 2003**

1. Added "Level 1 and Level 2" to ICS title.
2. Added content to Introduction section.
3. Added content to Terminology section.
4. Changed "Syntax" heading to "Syntactic Fragment" for each function.
5. Added description for "Conformance Levels" heading. Added "Comformance Levels" heading for each function.
6. Changed "Order Pages" function to "Page Order" and fixed description and value descriptions to reflect that the function describes how the pages have already been ordered and not how they are to be reordered.

### **D.4 Changes to make version 0.04, March 24, 2003**

1. Added captions to tables and figures. Added Table of Figures after Table of Contents.
2. Added a statement at the end of section "1.2 JDF Included in the ICS for Digital Printing" to say that combined process node may be part of a job ticket.
3. Added a statement in section "1.2.2 Structure of an ICS for Digital Printing JDF Job Ticket" to say that Activation attribute in a combined process JDF node is always set to Active. Also added an ISSUE.
4. Added a table to section 1.2.2.2 Resources and ResourcePool that lists all JDF Resources/Elements that are supported by the ICS for DP, the feature they are used by, the process(s) they are inputs to or outputs from, their attributes that are supported, and the conformance level.
5. Changed "function" terminology to "feature" in section "3.1 Features Supported".
6. Added statement to section "3.1 Features Supported" to say that features that use the same resource can be combined into a single resource definition in the job ticket. Added an example.
7. Added "(NMTOKEN)" text to description of values that are NMTOKENs in section "3.1 Features Supported".
8. Added text to FileSpec URL describing the format of the URL (e.g. file or http) in Document File Name feature.

9. Clarified FoldCatalog description in Fold feature.
10. Made Media Location LocationName Optional in Media feature
11. Corrected Interleave Sheets feature so that MediaRef is under a Sheet resource.
12. Added JDF Template="true" attribute/value to Job Ticket Template Comment or Description, Job Ticket Template ID, and Job Ticket Template Version features.
13. Changed MediaColor to MediaColorMeasurement for Media resource in Media feature to match updated JDF Spec.
14. Added StockType attribute to Media resource in Media feature.
15. Added None and SystemSpecified to HoleType attribute for Media resource in Media feature. Added ISSUE to address adding SystemSpecified to JDF Spec.
16. Added Cardinal, Cyan, Magenta, Royal, Ruby, and UserColor-n as values for MediaColorName in Media feature. Added ISSUE to address adding to JDF Spec.
17. Added statement to Number Up feature stating that it can be combined with Fit Policy feature.
18. Changed Process Color Model feature to have Page, Document, or Job scope (it only had Job scope).
19. Added an ISSUE to Proof Print feature to see if this is how the JDF should appear.
20. Added description to Range of Pages to Process describing the behavior printing 2-sided.
21. Removed ErrorDiffusion value from ScreeningFamily attribute in Screening Family feature to match updated JDF Spec.

## **D.5 Changes to make version 0.05, April 16, 2003**

1. Added new figure "Example of JDF and JMF workflow interactions" to section 1.22 JDF Included in ICS for Digital Printing .
2. Cleaned up some wording in section 1.2 JDF Included in ICS for Digital Printing.
3. Changed CombinedProcessIndex for ComponentLink in figure "Content of a ICS for Digital Printing Job Ticket" to state that it contains index of DigitalPrinting process or last finishing process.
4. Added wording to section 1.2.2 Structure of an ICS for Digital Printing Job Ticket about the order of commutative processes not being significant. Added an ISSUE about how this interacts with SettingsPolicy.
5. Updated table "Supported Processes and their Resources" to show optional Trapping after Interpreting and after Rendering. Clarified whether Trapping is vector or raster.
6. Added new table " Possible successor processes of each predecessor process" to section 1.2.2 Structure of an ICS for Digital Printing Job Ticket.
7. Added new figure " Process Flow" to section 1.2.2 Structure of an ICS for Digital Printing Job Ticket.

8. Added clarification in section 1.2.2.1 JDF Node that the JDF XML namespace must be declared as the default namespace in the job ticket.
9. Added statement in section 1.2.2.2 Resources and ResourcePool under Status description stating that Input resources are always Available and output resources are always Unavailable.
10. Removed ISSUE in table section 3.1 Features Supported of whether to use SheetIndex and DocSheetIndex. It was decided that BundeltemIndex would be used.
11. For Fit Policy feature, moved Abort and Tile values for SizePolicy attribute from the excluded list and added them as possible values.
12. Added ReferenceEdge attribute to Folding feature.
13. For Margins feature change NonPrintableMargins order from top, rught, bottom,left to left, bottom, right, top.
14. Added MediaColorNameDetails and MediaType attributes to Media feature.
15. Added Coated and InkJet values to Media features BackCoating and FrontCoating attributes.
16. Added Brand attribute to Media feature.
17. Removed ISSUE from Media feature's HoleType, SystemSpecified is being added as new value.
18. Change Media feature's MediaColorName attribute to be an enumeration. Removed Cardinal, Grey, Purple, Royal, Ruby, UserColor, UserColor-*n*. Added SystemSpecified and made it the default. Added "Clear" as a possible prefix.
19. For Media feature MediaTypeDetails, moved Unknown from excluded list and add it as a possible value, and the default.
20. Changed Media feature's RecycledPercentage to be a double instead of an integer. Removed -1 as a value and as the default.
21. Changed CombinedProcessIndex for ComponentLink in Number of Copies feature to state that it contains index of DigitalPrinting process or last finishing process.
22. Added Booklet as a value for Output Bin Name feature's OuputBin attribute.
23. Added Page Distribution feature.
24. Changed Printer Resolution to Device Resolution, changed it to have Page, Document, or Job scope, and Comformance Levels 1 and 2 instead of ?.
25. Added Presentation Presentation feature.
26. Changed Settings Policy description to state that SettingsPolicy can be set for elements as well as resouces. Removed the ISSUE asking whether SettingsPolicy can be set for processes. It was decided that it cannot.

## D.6 Changes to make version 0.06, May 23, 2003

1. Added a number of **ISSUEs** that need to be addressed/resolved.

2. Reorganized, rewrote parts of, added new information and sub-sections to sections 1, 2, and 3.1.
3. Made major enhancements to table in Figure 3-7 - Elements Supported in a Digital Printing Job Ticket. Added XPath, Element Attributes, Feature Attributes and Child Elements, and Feature Attribute Values columns.
4. Added section 3.2 Element and Resource Descriptions for a few JDF Elements and Resources to see if the section is of value.
5. Added Elements/Resource Used heading to section 3.3 Features Supported for a few features to see if the heading and its contents are of value.
6. Changed Device Resolution feature to Destination or Physical Printer Resolution.
7. Added MimeType attribute to Document File Format feature.
8. Added RotatePolicy attribute to FitPolicy feature.
9. Clarified descriptions of Disjointing OffsetDirection attribute.
10. Removed typical values for Media feature's DimensionName attribute and referenced PWG Standard of Standardized Media Names.
11. Removed Screening Family feature's ScreeningFamily typical values.
12. Added BestEffortExceptions, MustHonorExceptions, OperatorInterventionExceptions attributes to Settings Policy feature.
13. Added Trapping feature.

## **D.7 Changes to make version 0.07, June 10, 2003**

1. Added notes and ISSUES from 6/3, 6/5, and 6/10 Digital Printing working group meetings.

## **D.8 Changes to make version 0.08, August 22, 2003**

1. Added bundle item example to Partitionable Resources section.
2. Added Approval and ICS Versions to Features Supported section and to Elements Supported in a Digital Printing Job Ticket table.
3. Added explanation of swapping x and y dimension of Media resource to Feed Orientation feature.
4. Added Continuous, Disc, Disc-n, Front, Roll, Roll-n, and Tray to Location element's LocationName attribute in Input Tray Name feature and to Elements Supported in a Digital Printing Job Ticket table.
5. Added TimeStamp attribute to Job Created By and Job Modified By features and to Elements Supported in a Digital Printing Job Ticket table.
6. Split NonPrintableMargins into NonPrintableMarginsBottom, NonPrintableMarginsLeft, NonPrintableMarginsRight, and NonPrintableMarginsTop in Printable Margins feature and in Elements Supported in a Digital Printing Job Ticket table and to Elements Supported in a Digital Printing Job Ticket table.

7. Added S1-generic value to Media resource's HoleType attribute in Media feature and to Elements Supported in a Digital Printing Job Ticket table.
8. Added Disc and Other values to Media resource's MediaType attribute in Media feature and to Elements Supported in a Digital Printing Job Ticket table.
9. Removed Continuous, Paper, Transparency, and Unknown values from Media resource's MediaTypeDetails attribute in Media feature and from Elements Supported in a Digital Printing Job Ticket table.
10. Added Tractor value to Media resource's MediaTypeDetails attribute in Media feature and to Elements Supported in a Digital Printing Job Ticket table.
11. Added MediaUnit attribute with values Continous, Fan, and Sheet to Media resource in Media feature and to Elements Supported in a Digital Printing Job Ticket table..
12. Removed Unknown value from Media resource's StockType attribute and from Elements Supported in a Digital Printing Job Ticket table.
13. Swap y x to be x y for NumberUp attribute in Number Up feature.
14. Added DirectProofAmount attribute to DigitalPrintingParams resource in Proof Print feature and to Elements Supported in a Digital Printing Job Ticket table.
15. Added S1-generic to HoleMakingParms resource's HoleType attribute in Punching and HoleMaking feature and to Elements Supported in a Digital Printing Job Ticket table.
16. Added IncludeInBundleItem attribute to InsertSheet resource in Start, Separator/Slip, End Sheets feature and to Elements Supported in a Digital Printing Job Ticket table.
17. Removed TrimmingType from TrimmingParams resource in Trimming feature and from Elements Supported in a Digital Printing Job Ticket table.

## **D.9 Changes to make version 0.09, October 7, 2003**

1. Added an abstract to title page.
2. Reworded parts of "1. Introduction" and "1.1.1What is JDF?" sections.

## **D.10 Changes to make version 0.09, October 14, 2003**

1. Added clarification to Terminology section stating that references to JDF assume that it is the JDF representing a job.

## **D.11 Changes to make version 0.10, October 21, 2003**

1. Added "- JDF Process" to the title of this ICS.
2. Modified Comformance Levels for some features (Comment/Description of Job, Feed Orientation, Fit Policy, Job Name, Jog Offset, Margins, Notification, Page Delivery, and Print Quality) based on the decisions made by the FSG Open Print Job Ticket working group for Comformance Level 1.

3. Changed syntax of Document Compression, Document File Name, Document Format, and Document Natural Language features to not partition by Run if there is only a single document (file) in the job.
4. Changed Folding feature's FoldingParams FoldCatalog attribute's values to be Px-y instead of Fx-y.