



Exchange Job Definition Format

for Developers



The XJDF Book



This presentation is based on the **XJDF Book**, which is available here:



<https://ricebean.net/xjdf>



Introduction

Introduction

XJDF Essentials

Product Description

Process Description

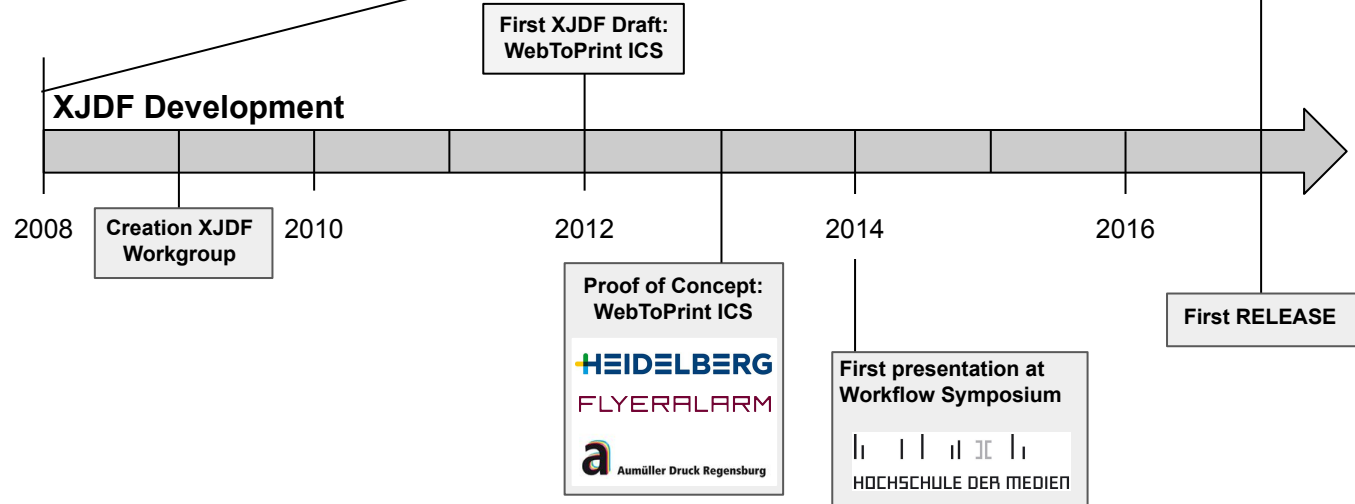
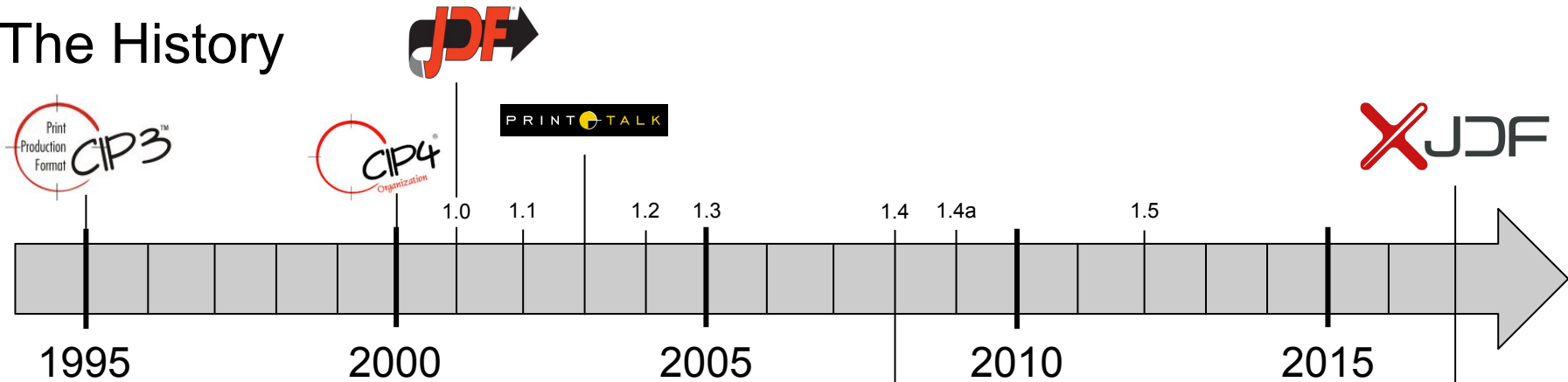
ICS

XJMF Messaging

Authoring Process



The History



XJDF Essentials

Introduction

XJDF Essentials

Product Description

Process Description

ICS

XJMF Messaging

Authoring Process



XJDF Essentials

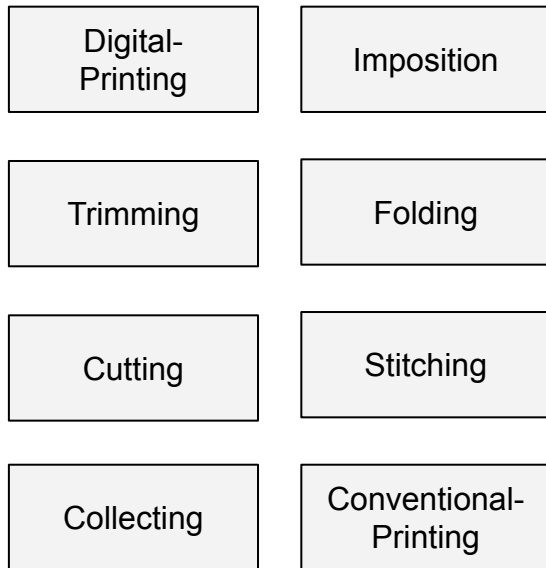
- Gray Boxes
- Common XML Structure
- Resources
- Extensibility
- Identifiers
- Measurements, Units and Data Types
- Folding Schemes and Pagination



Processes

The core of a Production Workflow

Sample of Processes:



- Each process is defined by its unique name, a description plus the input and output resources.
- A production workflow is a concatenation of individual processes
- A process is assumed to be executed on a single-purpose device

→ **The XJDF Specification defines a wide range of individual processes (see chapter 5).**



Processes

Sample of a Press Definition in the XJDF Specification

5.6.15 Folding

Buckle folders or knife folders are used for *Folding* sheets. One or more sheets can be folded at the same time. Web presses often provide in-line *Folding* equipment. Longitudinal *Folding* is often performed using a former, a plow folder or a belt. While jaw folding, chopper folding or drum folding equipment is used for folding the sheets that have been divided.

The **XJDF *Folding*** process covers both operations done in stand-alone *Folding* machinery – typically found for processing printed materials from Sheet-Fed presses – and in-line equipment of Web presses. Creasing and/or slot perforating are sometimes necessary parts of the *Folding* operation that guarantee exact process execution. They depend on the folder used, the **Media** and the folding layout. These operations are specified in the *Creasing* and *Perforating* processes respectively.

Table 5-96: Folding – Input Resources

Name	Description
Component ?	Component resources, including a printed sheet or a pile of sheets, are used in the <i>Folding</i> process.
FoldingParams ?	Specific parameters to set up the machinery.
Generic Input Resources*	See Table 5-1 for additional input resources that are valid for all process types.

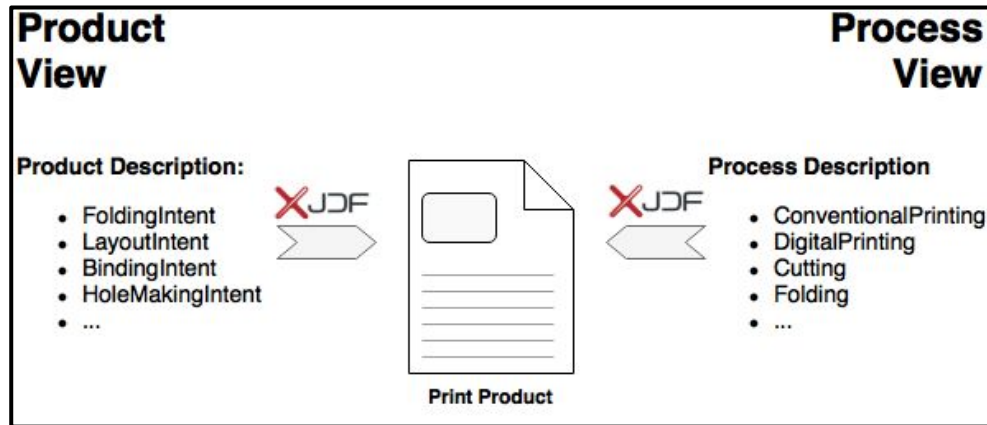
Table 5-97: Folding – Output Resources

Name	Description
Component ?	The process produces a Component , which in most cases is a folded sheet.



XJDF Views

XJDF specifies two views on print which are related but not identical:



The customer's perspective:
What should be produced?

vs.

The technical perspective:
How should be produced?



→ **The both views make XJDF very adaptable to many situations and environments.**

XJDF Essentials

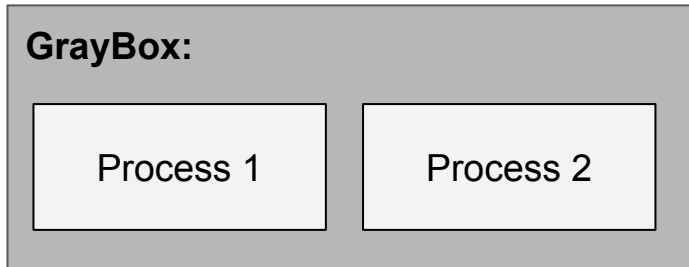
- **Gray Boxes**
- Common XML Structure
- Resources
- Extensibility
- Identifiers
- Measurements, Units and Data Types
- Folding Schemes and Pagination

Gray Boxes



Gray Boxes

What is a Gray Box?



- GrayBox is a commonly used term in XJDF
- A GrayBox describes one or a concatenation of multiple processes
- Each XJDF is targeted to a specific GrayBox
- Each physical device is a conceptually a GrayBox containing one or several processes.

→ **GrayBoxes are playing a fundamental role in XJDF Workflows.**

Gray Boxes

Process Definitions

Folding Machine

Image: <http://www.mb-bauerle.de>



GrayBox: "Folding"

Folding

Collecting Machine with a combined magazine finishing

Image: <http://www.mullermartini.com>



GrayBox: "Collecting Stitching Trimming"

Collecting

Stitching

Trimming

Gray Boxes

Definition of Gray Boxes in an XJDF Document

Folding Machine

Image: <http://www.mb-bauerfe.de>



GrayBox: "Folding"

Folding

XJDF Document defining the corresponding GrayBox **"Folding"**:

```
<xjdf:XJDF JobID="JOB-42" Types="Folding" xmlns:xjdf="http://www.CIP4.org/JDFSchema_2_0">  
  [...]  
</xjdf:XJDF>
```

Gray Boxes

Definition of Gray Boxes in an XJDF Document

Collecting Machine with a combined magazine finishing



Image: <http://www.mullermartini.com>

XJDF Document defining the corresponding GrayBox **“Collecting Stitching Trimming”**:

```
<xjdf:XJDF JobID="JOB-1234" Type="Collecting Stitching Trimming"  
  xmlns:xjdf="http://www.CIP4.org/XJDFSchema_2_0">  
  [...]  
</xjdf:XJDF>
```

GrayBox: “Collecting Stitching Trimming”

Collecting

Stitching

Trimming



Gray Boxes

Product Descriptions

- Even a pure Product Description (Product View) can be seen as a GrayBox.
- The attribute value “**Product**” specifies, the product described shall be produced without complete knowledge of the production workflow (processes):

```
<xjdf:XJDF JobID="JOB-42" Types="Product" xmlns:xjdf="http://www.CIP4.org/JDFSchema_2_0">
  [...]
</xjdf:XJDF>
```

- A combination of Product with Processes is also valid (partial defined Workflow):

```
<xjdf:XJDF JobID="JOB-42" Types="ConventionalPrinting Cutting Product"
  xmlns:xjdf="http://www.CIP4.org/JDFSchema_2_0">
  [...]
</xjdf:XJDF>
```



XJDF Essentials

- Gray Boxes
- **Common XML Structure**
- Resources
- Extensibility
- Identifiers
- Measurements, Units and Data Types
- Folding Schemes and Pagination

Common XML Structure



Common XML Structure

Here an abstract XJDF Document which outlines the common XML structure of such documents:

```
<XJDF JobID="5" Types="ProcessName_1 ProcessName_2" xmlns="http://www.CIP4.org/JDFSchema_2_0">
  <AuditPool />
  <Comment />
  <GeneralID />
  <ProductList>
    <Product>
      <Intent Name="SpecificIntent">
        <SpecificIntent />
      </Intent>
      <!-- further product intents go here -->
    </Product>
    <!-- further products go here -->
  </ProductList>
  <ResourceSet Name="SpecificResource">
    <Resource>
      <AmountPool />
      <Part />
      <SpecificResource />
    </Resource>
    <!-- further resources of the same type may go here -->
  </ResourceSet>
  <!-- further ResourceSets go here -->
</XJDF>
```



Common XML Structure

XJDF Element

- Each XJDF Document starts with the XJDF root node.
- The XJDF node can only exist once per file.

```
<xjdf:XJDF JobID="JOB-42" Types="Folding"
  xmlns:xjdf="http://www.CIP4.org/JDFSchema_2_0">
  [...]
</xjdf:XJDF>
```

Attribute	Description
@JobID (required)	Unique identifier of the current job as a XJDF Document is always within a context of a job.
@Types (required)	Names of a logical processes covered by the Gray Box which the document is targeted to.
@xmlns:xjdf (required)	Namespace of the current document. Namespace is unique per XJDF version.



Common XML Structure

AuditPool Element

- AuditPool can only exist ONCE per document
- Container for the post facto recorded results of a GrayBox such as
 - Notifications
 - Resource Consumptions
 - Status Information
- Sorting: Chronologically

```
[...]
<AuditPool>
  <AuditCreated>
    <Header ID="42" DeviceID="DEVICE_ID" Time="2016-12-28T10:45:05Z" />
  </AuditCreated>
  <AuditStatus>
    <Header ID="43" DeviceID="DEVICE_ID" Time="2016-12-28T10:47:19Z" />
    <DeviceInfo Status="Production" Speed="6000" />
  </AuditStatus>
  <AuditNotification>
    <Header ID="44" DeviceID="DEVICE_ID" Time="2016-12-28T10:49:26Z" />
    <Notification Class="Information">
      <Event EventID="3092" EventValue="Paper stock is low." />
    </Notification>
  </AuditNotification>
  [...]
</AuditPool>
[...]
```



Common XML Structure

Common Element

- Using comments requires ALWAYS human interactions. Writing a comment assumes always somebody who has to read and interpret the comment
- A comment MUST NOT influence the behavior of an device in any way.

```
<XJDF ...>
  [...]
  <Comment Type="Instruction" TimeStamp="2016-08-20T14:44:02+02:00"
    PersonalID="cnorris">
    Please take a special care for the image on the upper right.
    This is the leading motive of the product.
  </Comment>
  <Comment Type="JobDescription" TimeStamp="2016-08-20T14:44:02+02:00"
    PersonalID="cnorris">
    Flyer A6
    4c both sides
    135 gsm glossy paper
  </Comment>
  <!-- further comments may go here -->
  [...]
</XJDF>
```

→ **Comments cause less automated systems.**



Common XML Structure

GeneralID Element

- GeneralID describes a generic Identifier
- GeneralID can be used as a Key-Value pair for almost any purpose
- **Due to the generic behavior GeneralID is responsible for many incompatibility issues in JDF.**

```
[...]  
<GeneralID IDUsage="Height" DataType="double" IDValue="1440.0"/>  
<GeneralID IDUsage="Width" DataType="double" IDValue="720.0"/>  
[...]
```

→ **Before using GeneralIDs it is a much better way to extend XJDF by a private namespace.**



Common XML Structure

ProductList Element

- ProductList can exist ONCE per document
- Container for the (process independent) Product Description
- It is recommended to provide a full Product Description in each document in order to bring the process steps into the right context.



```
<ProductList>
  <Product Amount="250">
    <Intent Name="ColorIntent">
      <ColorIntent>
        <SurfaceColor ColorsUsed="Cyan Magenta Yellow Black" Surface="Front"/>
      </ColorIntent>
    </Intent>
    <Intent Name="LayoutIntent">
      <LayoutIntent FinishedDimensions="1683.779 2383.937 0.0" Sides="OneSided"/>
    </Intent>
    <Intent Name="MediaIntent">
      <MediaIntent MediaQuality="135_GLOSS"/>
    </Intent>
  </Product>
  <Product Amount="1000">
    <Intent Name="ColorIntent">
      <ColorIntent>
        <SurfaceColor ColorsUsed="Cyan Magenta Yellow Black" Surface="Front"/>
      </ColorIntent>
    </Intent>
    <Intent Name="LayoutIntent">
      <LayoutIntent FinishedDimensions="841.889 1190.551 0.0" Sides="OneSided"/>
    </Intent>
    <Intent Name="MediaIntent">
      <MediaIntent MediaQuality="135_GLOSS"/>
    </Intent>
  </Product>
</ProductList>
```

Common XML Structure

ResourceSet Element

- Product Descriptions as well as Process Descriptions, both require additional input and output resources
- Resource elements represent the ‘things’, such as inks, plates or glue that are produced or consumed by Processes. Resource elements also define the details of Processes, as well as any non-physical computer data such as files used by a Process.
- A ResourceSet element is a container for a single resource or for a logical group of resources.

```
[...]  
<xjdf:ResourceSet Name="RunList" Usage="Input" >  
  <xjdf:Resource>  
    <xjdf:RunList>  
      <xjdf:FileSpec URL="http://example.org/artwork/my.pdf"/>  
    </xjdf:RunList>  
  </xjdf:Resource>  
</xjdf:ResourceSet>  
[...]
```



Common XML Structure

XML Sample: Folding (1)

- As a conclusion of this chapter there is a real-world XJDF folding sample on the next slide.
- Input: Flat paper PLUS Parameters (such as Folding Params)
- Output: The folded Product or Product Part

Image: <http://www.mb-bauerle.de>



→ **Many details of the sample below are explained in detail later.**

→ **The focus of this sample is to get an overview of a real-world XJDF Document rather than having a complete understanding now.**



Common XML Structure

XJDF Sample: Folding (2)

Source: <http://ricebean.net/xjdf>

```
<xjdf:XJDF JobID="JOB-42" Types="Folding"
  xmlns:xjdf="http://www.CIP4.org/JDFSchema_2_0">
  <xjdf:AuditPool>
    <xjdf:Created AgentName="MIS" AgentVersion="1.0"
      TimeStamp="2016-05-30T21:49:47Z" />
  </xjdf:AuditPool>
  <xjdf:ProductList>
    <xjdf:Product Amount="2500" DescriptiveName="My Folded Leaflet">
      <xjdf:Intent Name="LayoutIntent">
        <xjdf:LayoutIntent Dimensions="841.88 595.27"
          Sides="TwoSidedHeadToHead" Pages="2"
          FinishedDimensions="280.69 595.27 0.0"/>
      </xjdf:Intent>
      <xjdf:Intent Name="FoldingIntent">
        <xjdf:FoldingIntent FoldCatalog="F6-2"/>
      </xjdf:Intent>
      <xjdf:Intent Name="ColorIntent">
        <xjdf:ColorIntent>
          <xjdf:SurfaceColor ColorsUsed="Cyan Magenta Yellow
            Black" Surface="Front"/>
          <xjdf:SurfaceColor ColorsUsed="Cyan Magenta Yellow
            Black" Surface="Back"/>
        </xjdf:ColorIntent>
      </xjdf:Intent>
      <xjdf:Intent Name="MediaIntent">
        <xjdf:MediaIntent MediaQuality="135_MATT"/>
      </xjdf:Intent>
    </xjdf:Product>
  </xjdf:ProductList>
```

[Continued on next row...]

```
<xjdf:ResourceSet Name="Component" Usage="Input">
  <xjdf:Resource ExternalID="COMP_ID_UNFOLDED">
    <xjdf:Component Dimensions="841.88 595.27 0.42"
      MediaRef="MEDIA-ID" />
  </xjdf:Resource>
</xjdf:ResourceSet>
<xjdf:ResourceSet Name="Component" Usage="Output">
  <xjdf:Resource ExternalID="COMP_ID_FOLDED">
    <xjdf:Component Dimensions="280.69 595.27 1.26"
      MediaRef="MEDIA-ID" />
  <xjdf:AmountPool>
    <xjdf:PartAmount Amount="2580" />
  </xjdf:AmountPool>
</xjdf:ResourceSet>
<xjdf:ResourceSet Name="FoldingParams" Usage="Input">
  <xjdf:Resource>
    <xjdf:FoldingParams FoldCatalog="F6-2"/>
  </xjdf:Resource>
</xjdf:ResourceSet>
<xjdf:ResourceSet Name="Media">
  <xjdf:Resource ID="MEDIA-ID">
    <xjdf:Media MediaType="Paper" Thickness="150" Weight="135"
      ISOPaperSubstrate="PS3" />
  </xjdf:Resource>
</xjdf:ResourceSet>
<xjdf:ResourceSet Name="NodeInfo" Usage="Input">
  <xjdf:Resource>
    <xjdf:NodeInfo Start="2016-08-30T15:00:00Z" />
  </xjdf:Resource>
</xjdf:ResourceSet>
</xjdf:XJDF>
```

XJDF Essentials

- Gray Boxes
- Common XML Structure
- **Resources**
- Extensibility
- Identifiers
- Measurements, Units and Data Types
- Folding Schemes and Pagination

Resources



Resources

- Resource elements represent the ‘things’, such as inks, plates or glue that are produced or consumed by Processes. Resource elements also define the details of Processes, as well as any non-physical computer data such as files used by a Process.
- A RunList is mainly used in prepress processes as well as in web-to-print scenarios in order to reference the customer artworks or other digital data to be processed.
- The attribute @Usage defines the usage of the ResourceSet within the XJDF Document. Valid values are “Input” and “Output”.

```
[...]
<xjdf:ResourceSet Name="RunList" Usage="Input" >
  <xjdf:Resource>
    <xjdf:RunList>
      <xjdf:FileSpec URL="http://example.org/artwork/my.pdf"/>
    </xjdf:RunList>
  </xjdf:Resource>
</xjdf:ResourceSet>
[...]
```



Resources

References of Resources

- ResourceSets may contain Resources which are neither input nor output of a processes (@Usage does not exist at all).
- Resources can also being linked by third party resources such as the Media Resource in the Folding Sample

```
[...]  
<xjdf:ResourceSet Name="Component" Usage="Input">  
  <xjdf:Resource ExternalID="COMP_ID_UNFOLDED">  
    <xjdf:Component Dimensions="841.88 595.27 0.42"  
      MediaRef="MEDIA-ID" />  
  </xjdf:Resource>  
</xjdf:ResourceSet>  
<xjdf:ResourceSet Name="Component" Usage="Output">  
  <xjdf:Resource ExternalID="COMP_ID_FOLDED">  
    <xjdf:Component Dimensions="280.69 595.27 1.26"  
      MediaRef="MEDIA-ID" />  
  </xjdf:Resource>  
</xjdf:ResourceSet>  
[...]  
<xjdf:ResourceSet Name="Media">  
  <xjdf:Resource ID="MEDIA-ID">  
    <xjdf:Media MediaType="Paper" Thickness="150" Weight="135"  
      ISOPaperSubstrate="PS3" />  
  </xjdf:Resource>  
</xjdf:ResourceSet>  
[...]
```



Resources

Partitioning of Resources (1)

- When describing a logically set of resources in a ResourceSet, XJDF provides a partitioning mechanism in order to identify the appropriate resource
- Partitioning of Resources is a JDF / XJDF specific mechanism
(No XML Standard)
- Part Element is the key element for partitioning -
the attributes of Part are called Partition Keys.

```
[...]
<xjdf:ResourceSet Name="RunList">
  <xjdf:Resource>
    <xjdf:Part Side="Front" />
    <xjdf:RunList>
      <xjdf:FileSpec URL="http://example.org/artwork/my-page-1.pdf"/>
    </xjdf:RunList>
  </xjdf:Resource>
  <xjdf:Resource>
    <xjdf:Part Side="Back" />
    <xjdf:RunList>
      <xjdf:FileSpec URL="http://example.org/artwork/my-page-2.pdf"/>
    </xjdf:RunList>
  </xjdf:Resource>
</xjdf:ResourceSet>
[...]
```



Resources

Partitioning of Resources (2)

Here a sample using the Partition Key “Separation” in order to define the logical group of four printing plates required by a four colored printing job:

```
<xjdf:XJDF JobID="JOB-42" Types="ConventionalPrinting"
  xmlns:xjdf="http://www.CIP4.org/JDFSchema_2_0">
  [...]
  <ResourceSet Name="ExposedMedia" Usage="Input" ProcessUsage="Plate">
    <Resource>
      <Part Separation="Cyan"/>
      <ExposedMedia MediaRef="MEDIA-PLATE-ID" PlateType="Exposed" />
    </Resource>
    <Resource>
      <Part Separation="Magenta"/>
      <ExposedMedia MediaRef="MEDIA-PLATE-ID" PlateType="Exposed" />
    </Resource>
    <Resource>
      <Part Separation="Yellow"/>
      <ExposedMedia MediaRef="MEDIA-PLATE-ID" PlateType="Exposed" />
    </Resource>
    <Resource>
      <Part Separation="Black"/>
      <ExposedMedia MediaRef="MEDIA-PLATE-ID" PlateType="Exposed" />
    </Resource>
  </ResourceSet>
  [...]
</XJDF>
```



Resources

Partitioning of Resources (3)

A part element may also define multiple Partition Keys or a resource element may even contain multiple Part elements:

```
[...]
<xjdf:ResourceSet Name="ExposedMedia">
  <xjdf:Resource>
    <xjdf:Part Separation="Cyan" PartVersion="German" />
    <xjdf:Part Separation="Cyan" PartVersion="English" />
    <xjdf:ExposedMedia ... />
  </xjdf:Resource>
  <xjdf:Resource>
    <xjdf:Part Separation="Magenta" PartVersion="German" />
    <xjdf:Part Separation="Magenta" PartVersion="English" />
    <xjdf:ExposedMedia ... />
  </xjdf:Resource>
  <xjdf:Resource>
    <xjdf:Part Separation="Yellow" PartVersion="German" />
    <xjdf:Part Separation="Yellow" PartVersion="English" />
    <xjdf:ExposedMedia ... />
  </xjdf:Resource>
  <xjdf:Resource>
    <xjdf:Part Separation="Black" PartVersion="German" />
    <xjdf:ExposedMedia ... />
  </xjdf:Resource>
  <xjdf:Resource>
    <xjdf:Part Separation="Black" PartVersion="English" />
    <xjdf:ExposedMedia ... />
  </xjdf:Resource>
</xjdf:ResourceSet>
[...]
```



XJDF Essentials

- Gray Boxes
- Common XML Structure
- Resources
- **Extensibility**
- Identifiers
- Measurements, Units and Data Types
- Folding Schemes and Pagination

Extensibility



Foreign Namespaces

Foreign Namespaces can be used to extend XJDF by additional elements and attributes.

- Extensions MUST NOT duplicate the functionality which is already defined in XJDF.
- **Custom extends may harm the long term stability of interfaces.**
- Custom extends are allowed in the following predefined locations:
 - Custom intents and resources
 - Custom processes
 - Custom values in NMTOKEN lists.
 - Custom notifications

```
<xjdf:XJDF          Types="ProcessName"          JobID="5"
                  xmlns:xjdf="http://www.CIP4.org/JDFSchema_2_0"
                  xmlns:foo="http://example.org/foo">
  [...]
</xjdf:XJDF>
```

→ **When working with private namespaces it is recommended to have only one namespace URI and only one standardized prefix per company.**

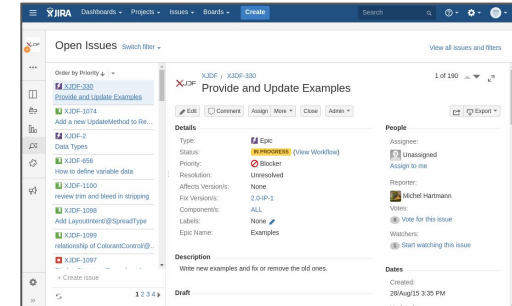




Foreign Namespaces

When missing functionality, the **preferred way** is to **start a new discussion** in the **CIP4s Issue tracking system (JIRA)** (see Spec Authoring Process).

The result of the discussion is either a solution of how to resolve the request using the current standard methods or an extension or modification of the specification for the next version of XJDF. Within the interim period, private namespaces can be used.



<https://jira.cip4.org/browse/XJDF>

→ **Custom extensions should only being used temporarily to bridge the next Version of the XJDF Specification.**



Custom Resources

- The new custom specific resource element is a subelement of Resource.
- The full name of the element (including the prefix) is the value of the @Name attribute in ResourceSet.

```
<xjdf:XJDF          Types="ProcessName"          JobID="5"
                  xmlns:xjdf="http://www.CIP4.org/JDFSchemata\_2\_0"
                  xmlns:foo="http://example.org/foo">
  [...]
  <xjdf:ResourceSet Name="foo:bar">
    <xjdf:Resource>
      <foo:bar attribute="myAttributeValue" />
    </xjdf:Resource>
  </xjdf:ResourceSet>
  [...]
</xjdf:XJDF>
```



Custom Intents

- The new custom specific Intent element is a subelement of Intent.
- The full name of the element (including the prefix) is the value of the @Name attribute in Intent.

```
<xjdf:XJDF          Types="ProcessName"          JobID="5"
                  xmlns:xjdf="http://www.CIP4.org/JDFSschema_2_0"
                  xmlns:foo="http://example.org/foo">
  [...]
  <xjdf:ProductList>
    <xjdf:Product>
      <xjdf:Intent Name="foo:bar">
        <foo:bar attribute="myAttributeValue" />
      </xjdf:Intent>
    </xjdf:Product>
  </xjdf:ProductList>
  [...]
</xjdf:XJDF>
```



Custom Processes

- The types attributes allows custom process names.
- The process's name including the company's prefix has to be written in the @Types attribute

```
<xjdf:XJDF      Types="foo:MyProcess"      JobID="5"  
              xmlns:xjdf="http://www.CIP4.org/JDFSchema_2_0"  
              xmlns:foo="http://example.org/foo">  
  [...]  
</xjdf:XJDF>
```



Custom Values in NMTOKEN lists

- NMTOKENS are open lists by definition.
- Custom keywords has been prefixed with a namespace-like syntax

4.8 FoldingIntent
This *Product Intent* element specifies the straight line folding, creasing and perforating of a product.

Intent Properties
Process Resource Pairing: [CreasingParams](#), [CuttingParams](#), [Fold](#), [FoldingParams](#), [PerforatingParams](#)

Table 4.27: FoldingIntent Element

Name	Data Type	Description
<i>FoldCatalog</i> ?	NMTOKEN	Describes the folding scheme according to the folding catalog in Appendix B Fold Catalogs. Note: The folding scheme in this context refers to the folding of the finished product as seen after the cutting, not the folding, of the sheet as seen in production.
<i>FoldingDetails</i> ?	NMTOKEN	<i>@FoldingDetails</i> is a system dependent descriptor of the folding. <i>@FoldingDetails</i> MAY be used to differentiate differing fold dimensions with the same general topology, such as asymmetrical Z-folds. <i>@FoldingDetails</i> SHALL NOT be specified if <i>@FoldCatalog</i> is not present
<i>Orientation</i> ?	enumeration	<i>@Orientation</i> indicates the orientation of the unfolded product with respect to the lay of the fold. A value of <i>Rotated</i> SHALL be mapped to the lay of the fold on the lower left of the product prior to folding and the front side of the product oriented in the direction of an upward fold. Allowed value is from: <i>Orientation</i>
<i>Crease</i> *	element	<i>Crease</i> elements describe the details of any creasing operations in the coordinate system of the final product. If no geometrical details are specified in the <i>crease</i> element and a <i>@FoldCatalog</i> is specified, the customer is requesting production creasing.
<i>Fold</i> *	element	<i>Fold</i> describe the details of folding operations in the sequence described by the value of

```
<xjdf:XJDF           Types="ProcessName"           JobID="5"
           xmlns:xjdf="http://www.CIP4.org/JDFSchema_2_0"
           xmlns:foo="http://example.org/foo">
  [...]
  <FoldingIntent FoldCatalog="foo:F6-MY-FOLD" />
  [...]
</xjdf:XJDF>
```



Identifiers



XJDF Essentials

- Gray Boxes
- Common XML Structure
- Resources
- Extensibility
- **Identifiers**
- Measurements, Units and Data Types
- Folding Schemes and Pagination

Extract of Common XJDF Identifiers

ID Name	Description	Scope	Sample
ExternalID	Unique identifiers of physical resources.	Production	COMP-34
ID / IDRef	Unique identifiers and references within a single XJDF Document	Document	j3hduw83
Partition Key	Unique identifier of a specific partition.	Production	Sheet-1
JobID	Unique identifier of a job.	Production	46478349
DeviceID	Unique identifier of a device.	Production	Press-1
...



Identifier: ExternalID

Many Resources that are consumed by production Processes are things that are tracked for inventory management purposes.

These Resources MAY be identified in XJDF with Resource/@ExternalID

Each resource can be assigned by an item code, article number or identifiers

```
[...]  
<xjdf:ResourceSet Name="Component" Usage="Input">  
  <xjdf:Resource ExternalID="COMP_ID_UNFOLDED">  
    <xjdf:Component Dimensions="841.88 595.27 0.42"  
      MediaRef="MEDIA-ID" />  
  </xjdf:Resource>  
</xjdf:ResourceSet>  
<xjdf:ResourceSet Name="Component" Usage="Output">  
  <xjdf:Resource ExternalID="COMP_ID_FOLDED">  
    <xjdf:Component Dimensions="280.69 595.27 1.26"  
      MediaRef="MEDIA-ID" />  
  </xjdf:Resource>  
</xjdf:ResourceSet>  
[...]
```

→ **ExternalID is unique within a PRODUCTION.**



Identifier: ID / IDRef

- ID / IDRef attributes are for internal document references only.
- ID / IDRefs must not be used to identify resources within the production workflow
- Unless otherwise specified, an @IDREF or @IDREFS will refer to an individual Resource rather than an entire ResourceSet.

```
[...]
<xjdf:ResourceSet Name="Component" Usage="Output">
  <xjdf:Resource ExternalID="COMP_ID_FOLDED">
    <xjdf:Component Dimensions="280.69 595.27 1.26"
      MediaRef="MEDIA-ID" />
  </xjdf:Resource>
</xjdf:ResourceSet>
[...]
<xjdf:ResourceSet Name="Media">
  <xjdf:Resource ID="MEDIA-ID">
    <xjdf:Media MediaType="Paper" Thickness="150"
      Weight="135"
      ISOPaperSubstrate="PS3" />
  </xjdf:Resource>
</xjdf:ResourceSet>
[...]
```

→ ID / IDRefs are unique within a single XJDF DOCUMENT.



Identifier: Partition Keys (1)

As soon as a ResourceSet contains several Resource elements and these elements are NOT referenced using ID / IDRefs, the subselection of the appropriate resources is based on partitioning.

- Partition Keys are the attributes of the Part element
- Valid Values are given by the specification.

```
[...]
<ResourceSet Name="ExposedMedia" Usage="Input" ProcessUsage="Plate">
  <Resource>
    <Part Separation="Cyan"/>
    <ExposedMedia MediaRef="MEDIA-PLATE-ID" PlateType="Exposed" />
  </Resource>
  <Resource>
    <Part Separation="Magenta"/>
    <ExposedMedia MediaRef="MEDIA-PLATE-ID" PlateType="Exposed" />
  </Resource>
  <Resource>
    <Part Separation="Yellow"/>
    <ExposedMedia MediaRef="MEDIA-PLATE-ID" PlateType="Exposed" />
  </Resource>
  <Resource>
    <Part Separation="Black"/>
    <ExposedMedia MediaRef="MEDIA-PLATE-ID" PlateType="Exposed" />
  </Resource>
</ResourceSet>
[...]
```



→ **Partition Keys are unique within a PRODUCTION.**

Identifier: Partition Keys (2)

Extract from the XJDF Specification showing the full list of valid partition keys for the partition “Separation”.

Table 3-5: Part Element (Sheet 4 of 5)

Name	Data Type	Description
<i>Separation</i> ?	NMTOKEN	<p>Identifies the separation name.</p> <p>Values include:</p> <p><i>Composite</i> – Non-separated resource.</p> <p><i>Separated</i> – The resource is separated, but the separation definition is handled internally by the resource, such as a PDF file that contains SeparationInfo dictionaries.</p> <p><i>Cyan</i> – Process color.</p> <p><i>Magenta</i> – Process color.</p> <p><i>Yellow</i> – Process color.</p> <p><i>Black</i> – Process color.</p> <p><i>Red</i> – Additional process color.</p> <p><i>Green</i> – Additional process color.</p> <p><i>Blue</i> – Additional process color.</p> <p><i>Orange</i> – Additional process color.</p> <p><i>Spot</i> – Generic spot color. Used when the exact nature of the spot color is unknown.</p> <p><i>Varnish</i> – Varnish.</p> <p><i>none</i> – explicit reference to a skipped module (i.e., no separation). Note that “<i>none</i>” is spelled in lower case so that its value is identical to the pre-defined separation “<i>none</i>” in [PDF1.6].</p> <p>Note: Other values specify spot colors.</p>



XJDF Essentials

- Gray Boxes
- Common XML Structure
- Resources
- Extensibility
- Identifiers
- **Measurements, Units and Data Types**
- Folding Schemes and Pagination

Measurements, Units and Data Types



Default Units in XJDF

- XJDF specifies most values in default units
- All measurable quantities are stated in double precision.
- Processors SHOULD NOT specify a unit unless no default exists, such as when new Resources are defined
- **Overriding the default units that are defined is non-standard and MAY lead to undefined behavior**

Table 1.6: Units Used in XJDF

Measurement	Unit	Representation	Remarks
Angle	degree°	degree	—
Area	m ²	m2	Used for media, (e.g., in wide format printing).
Countable Objects	1	count	Countable objects, such as sheets, MAY be specified as "count".
Length	point (1/72 inch)	pt	Used for all except microscopic lengths (see below)
	micron (μ)	um	Used for microscopic lengths — where used (instead of points) it will be explicitly stated in the definition of the item. See Media/Thickness .
Line Screen	lpi	lpi	The lines per inch (lpi) for conventionally screened halftone, screened grayscale and screened monotone bitmap images.
Paper weight	g/m ²	gsm	Paper weight SHALL be provided in grams per square meter. See Appendix E North American and Japanese Media Weight Explained for details of calculating non-gsm paper weights.
Power (electrical)	kilowatt hour	kWh	Used to measure consumption of electricity. Note: Current power consumption (kW) MAY be provided in a ResourceInfo as 'rate of consumption' of electric power, i.e. kWh/h=kW.
Resolution	dpi	dpi	The dots per inch (dpi) for print output and bitmap image (TIFF, BMP, etc.) file resolution.
Screen Resolution	ppi	ppi	The pixels per inch (ppi) for screen display (e.g., softproof display and user interface display), scanner capture settings and digital camera settings.
Speed	unit/hour		Speed SHALL be specified in base units per hour. The base units that are used to represent speed SHALL be identical to the base unit.
Spot Resolution	spi	spi	For imaging devices such as filmsetters, platesetters and proofers, the fundamental imaging unit (e.g., one "on" laser or imaging head imaged unit). Note: Many imaging devices construct dots from multiple imaging spots, so dpi and spots per inch (spi) NEED NOT be equivalent.
Temperature	C° (Celsius)	C	degree centigrade
Volume (gas)	m ³ (cubic meter)	m3	Used to measure consumption of gas
Volume (liquid)	liter	l	—
Weight	gram	g	—



Length Measurements

The most significant measurement in XJDF is the length. All length values except microscopic lengths are defined in desktop publishing points (dtp points).

A point is defined by a 1/72 inch.

```
// Conversion dtp points (dtp) to millimeter (mm)
```

```
mm = dtp * 72.0 / 25.4
```

```
// Conversion millimeter (mm) to dtp points (dtp)
```

```
dtp = mm * 25.4 / 72.0
```



Length Measurements - Data Types

Data Type	Description	Sample
<code>double</code>	A double is a single value expression as used for width or height expressions.	"595.27"
<code>rectangle</code>	"Attributes of type rectangle are used to describe rectangular locations on the page, Sheet or other printable surface. A rectangle is represented as an array of four numbers - llx lly urx ury - specifying the lower-left x, lower-left y, upper-right x and upper-right y coordinates of the rectangle, in that order." The origin for positioning is always the lower left corner of the parent object.	"10.0 10.0 290.69 605.27"
<code>shape</code>	"A shape is represented as an array of three (positive or zero) numbers - x y z - specifying the width x, height y and depth z coordinates of the shape, in that order." A shape is typically used for three-dimensional objects such as a book block.	"280.69 595.27 1.26"
<code>XYPair</code>	"Attributes of type XYPair are used to describe sizes like @Dimensions [...]. They can also be used to describe positions on a page". A XYPair is typically used for two-dimensional definitions as for instance page size is.	"290.69 605.27"

Length Measurements

Length Expressions in an XJDF Document

- The attribute `//LayoutIntent/@Dimensions` is of datatype `XYPair`
- The attributes `//LayoutIntent/@FinishedDimensions` and `//Component/@Dimensions` are of datatype `shape`

```
<xjdf:ProductList>
  <xjdf:Product Amount="2500" DescriptiveName="My Folded Leaflet">
    <xjdf:Intent Name="LayoutIntent">
      <xjdf:LayoutIntent Dimensions="841.88 595.27"
        Sides="TwoSidedHeadToHead"
        FinishedDimensions="280.69 595.27 0.0"/>
    </xjdf:Intent>
    [...]
  </xjdf:Product>
</xjdf:ProductList>
<xjdf:ResourceSet Name="Component" Usage="Output">
  <xjdf:Resource ExternalID="COMP_ID_FOLDED">
    <xjdf:Component Dimensions="280.69 595.27 1.26"
      MediaRef="MEDIA-ID" />
  </xjdf:Resource>
</xjdf:ResourceSet>
```



Length Measurements

Length Expressions how specified in the Specification

Here, the corresponding definitions of the LayoutIntent attributes from the XJDF Specification.

Table 6-31: LayoutIntent Element (Sheet 1 of 2)

Name	Data Type	Description
<i>Bleed</i> ?	double	Bleed of the artwork in points. The value of 0 means NO bleed. A negative value indicates bleed is needed but the value is unknown.
<i>Dimensions</i> ?	XYPair	Specifies the width (X) and height (Y) in points, respectively, of the trimmed and unfolded (flat) product. For example, <i>Dimensions</i> for a Z-fold is the unfolded dimensions, while <i>@FinishedDimensions</i> is the folded dimensions if known. Use <i>Dimensions</i> if <i>@FinishedDimensions</i> is not known. <i>@Dimensions</i> is provided for the rare case that <i>@FinishedDimensions</i> does not unambiguously define the finished product, due to complex folding schemes. If both values are specified, <i>@FinishedDimensions</i> takes precedence
<i>FinishedDimensions</i> ?	shape	Specifies the width (X), height (Y) and depth (Z) in points, respectively, of the finished product Component after all finishing operations, including folding, trimming, etc. If the Z coordinate is 0, it SHALL be ignored. Only <i>@FinishedDimensions</i> SHOULD be specified if both <i>@FinishedDimensions</i> and <i>@Dimensions</i> are known



XJDF Essentials

- Gray Boxes
- Common XML Structure
- Resources
- Extensibility
- Identifiers
- Measurements, Units and Data Types
- **Folding Schemes and
Pagination**

Folding Schemes and Pagination



Folding and Pagination

- Folding and Pagination are major tasks in a print production.
- **Folding** describes **how** a sheet or a final product has to be **folded** (e. g. zig-zag fold) and **pagination** defines how the single pages have to be **ordered and positioned**.

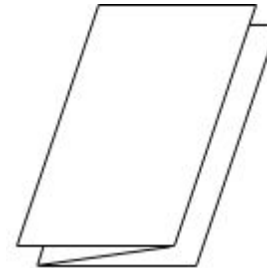


Figure: Sketch of a F6-2 Fold

Product Parts

- Each print product is being described as a set of one or multiple product parts
- A simple flyer or a folded leaflet consists of just one single product part.
- A complex product consists of at least two or even more product parts

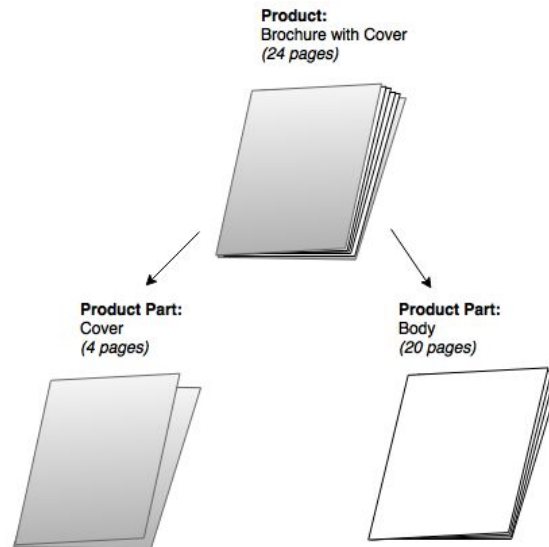
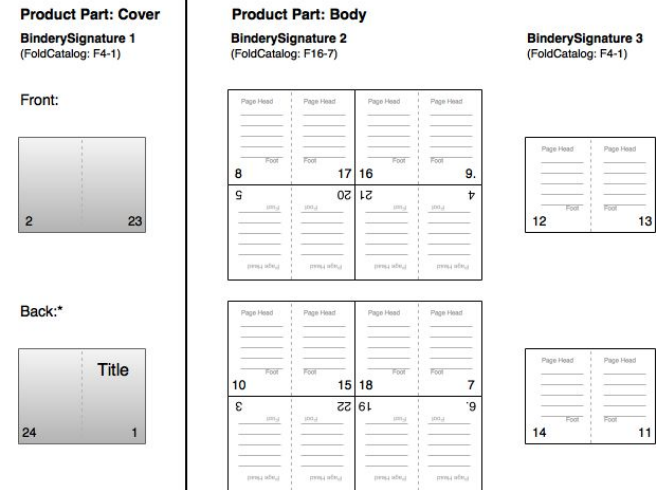


Figure: Product parts of a 24-page brochure

Bindery Signatures

- Each product part has to be separated into one or several BinderySignatures
- A BinderySignature represents both sides of an explicit folding signature with one or more pages.
- Separation abilities of a 16-pages brochure:
 - 1 x 16 pages
 - 2 x 8 pages
 - 4 x 4 pages
 - ...combinations are also valid



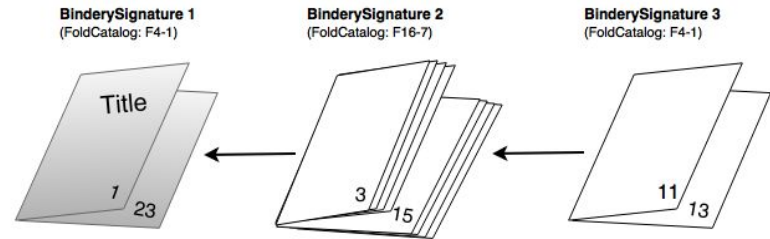
* BinderySignatures have been turned by flipping horizontally

Figure: BinderySignatures of a 24-paged brochure

→ A BinderySignature is conceptually the smallest physical component in which all print products are being separated.

Assembling of a Brochure

- “BinderySignature 1” has four pages and represents the cover.
- “BinderySignature 2” has sixteen pages and contains pages 3 to 10 and 15 to 22.
- “BinderySignature 3” has four pages and contains pages in the middle, 11 to 14.
- **All three BinderySignatures as sticked one in another from the outside to the inside.**



Definition of BinderySignatures in XJDF

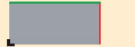








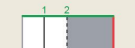


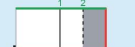
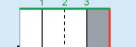
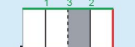


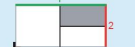

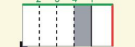
- All BinderySignatures are part of the same ResourceSet.
- BinderySignatures are partitioned at least by its BinderySignatureID.
- The BinderySignatureID partition key is the unique identifier of a BinderySignature within the production workflow.
- FoldCatalog attribute is a fold catalog identifier which refers to a predefined standard fold schema in the XJDF fold catalog.

```
[...]  
<xjdf:ResourceSet Name="BinderySignature" Usage="Input">  
  <xjdf:Resource>  
    <xjdf:Part BinderySignatureID="BS-1" ProductPart="COVER-ID" />  
    <xjdf:BinderySignature FoldCatalog="F4-1"/>  
  </xjdf:Resource>  
  <xjdf:Resource>  
    <xjdf:Part BinderySignatureID="BS-2" ProductPart="BODY-ID" />  
    <xjdf:BinderySignature FoldCatalog="F16-7"/>  
  </xjdf:Resource>  
  <xjdf:Resource>  
    <xjdf:Part BinderySignatureID="BS-3" ProductPart="BODY-ID" />  
    <xjdf:BinderySignature FoldCatalog="F4-1"/>  
  </xjdf:Resource>  
</xjdf:ResourceSet>  
[...]
```








The XJDF Fold Catalog

- XJDF comes with a predefined standardized Fold Catalog
- The XJDF fold catalog describes the most common fold schemes
- All folding schemes in the XJDF fold catalog are chronologically grouped by the number of finished pages.

F2-1 	F4-1 2x1  ↑1/2	F4-2 2x1  ↓1/2	F6-1 3x1  ↑1/3 ↓1/3	F6-2 3x1  ↓1/3 ↑1/3
F6-3 3x1  ↑1/4 ↑1/2	F6-4 3x1  ↑1/3 ↑1/3	F6-5 3x1  ↑2/3 ↓1/3	F6-6 3x1  ↑3/4 ↓1/4	F6-7 3x1  ↑1/4 ↓1/4
F6-8 3x1  ↑2/3 ↑1/3	F8-1 4x1  ↑1/2 ↑1/4	F8-2 4x1  ↑1/2 ↓1/4	F8-3 4x1  ↑1/4 ↓1/4 ↑1/4	F8-4 4x1  ↑1/4 ↑1/2 ↓1/4
F8-5 4x1  ↑1/4 ↑1/4 ↑1/4	F8-6 4x1  ↑3/4 ↓1/4 ↓1/4	F8-7 2x2  ↑1/2 + ↑1/2	F10-1 5x1  ↑1/5 ↓1/5 ↑1/5 ↓1/5	F10-2 5x1  ↑4/5 ↓1/5 ↓1/5 ↓1/5

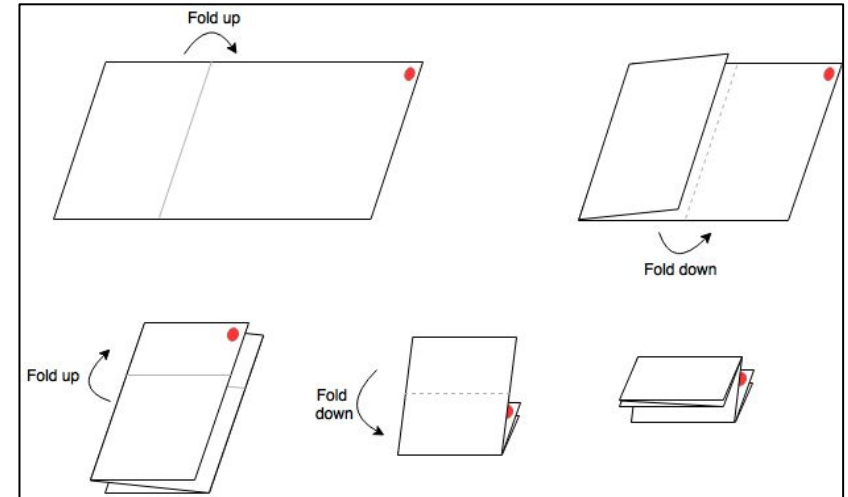
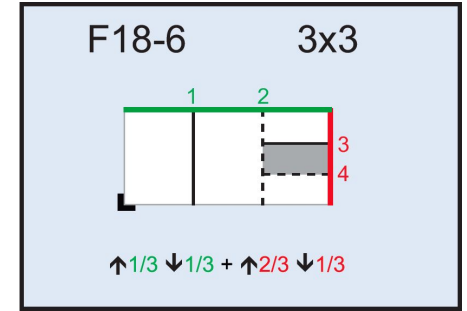
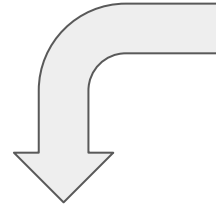


Legend of the XJDF Fold Catalog

Sign	Name	Description
	Fold Up	The bottom section is folded that way, that it remains upon the top section. When the "Fold up" is drawn horizontally, the left section remains above the right section after folding.
	Fold Down	The bottom section is folded that way, that it remains below the top section. When the "Fold down" is drawn horizontally, the left section remains under the right section after folding.
	Final Format	The final format of the folded signature after all folds has been applied. The gray area also shows the final position of the folded signature after the last fold.
1, 2, 3	Fold Order	The numbers define the chronological order of the folds.
	Lay	The lay of the open sheet.
	Open Format	The open sheet format of the signature. The green (upper) line marks the length of the open sheet whereas the red (right) one marks the width of the open sheet.

Interpretation of the Folding Scheme

- Here, a sequence of a “F18-6” fold schema is shown step by step.
- The red dot on the upper right corner of the sheet indicates the fixed point during the whole folding process.



Collecting Sample (1)

A Real-World Sample

- Here a real-world XJDF Collecting Sample targeted to a “Collecting Stitching Trimming” Gray Box.
- The 24-paged magazine is defined process neutral in ProductList.
- The Input- and Output Resources are the aggregation of all three Processes.
- Internal intermediate resources are NOT being specified.



```

<xjdf:XJDF JobID="JOB-1234" Types="Collecting Stitching Trimming"
  xmlns:xjdf="http://www.CIP4.org/JDFSchema_2_0">
  <xjdf:AuditPool>
    <xjdf:Created AgentName="MIS" AgentVersion="1.0" TimeStamp="2016-09-26T18:15:47Z" />
  </xjdf:AuditPool>
  <xjdf:ProductList>
    <xjdf:Product Amount="1000" DescriptiveName="My Magazine A4" IsRoot="true">
      <xjdf:Intent Name="BindingIntent">
        <xjdf:BindingIntent BindingOrder="Collecting" BindingSide="Left"
          BindingType="SaddleStitch" ChildRefs="COVER-ID BODY-ID"/>
      </xjdf:Intent>
    </xjdf:Product>
    <xjdf:Product ID="COVER-ID" IsRoot="false">
      <xjdf:Intent Name="ColorIntent">
        <xjdf:ColorIntent>
          <xjdf:SurfaceColor Surface="Front" ColorsUsed="Cyan Magenta Yellow Black" />
          <xjdf:SurfaceColor Surface="Back" ColorsUsed="Cyan Magenta Yellow Black" />
        </xjdf:ColorIntent>
      </xjdf:Intent>
      <xjdf:Intent Name="LayoutIntent">
        <xjdf:LayoutIntent Sides="TwoSidedHeadToHead" Pages="4"
          FinishedDimensions="841.88 595.27 0.0"/>
      </xjdf:Intent>
      <xjdf:Intent Name="MediaIntent">
        <xjdf:MediaIntent MediaQuality="135_MATT"/>
      </xjdf:Intent>
    </xjdf:Product>
    <xjdf:Product ID="BODY-ID" IsRoot="false">
      <xjdf:Intent Name="ColorIntent">
        <xjdf:ColorIntent>
          <xjdf:SurfaceColor Surface="Front" ColorsUsed="Black" />
          <xjdf:SurfaceColor Surface="Back" ColorsUsed="Black" />
        </xjdf:ColorIntent>
      </xjdf:Intent>
      <xjdf:Intent Name="LayoutIntent">
        <xjdf:LayoutIntent Sides="TwoSidedHeadToHead" Pages="20"
          FinishedDimensions="841.88 595.27 0.0"/>
      </xjdf:Intent>
      <xjdf:Intent Name="MediaIntent">
        <xjdf:MediaIntent MediaQuality="80_MATT"/>
      </xjdf:Intent>
    </xjdf:Product>
  </xjdf:ProductList>

```

Collecting Sample (2)

```

<xjdf:ResourceSet Name="Assembly" Usage="Input">
  <xjdf:Resource>
    <xjdf:Assembly BinderySignatureIDs="BS-1 BS-2 BS-3" Order="Collecting" />
  </xjdf:Resource>
</xjdf:ResourceSet>
<xjdf:ResourceSet Name="BinderySignature" Usage="Input">
  <xjdf:Resource>
    <xjdf:Part BinderySignatureID="BS-1" ProductPart="COVER-ID" />
    <xjdf:BinderySignature FoldCatalog="F4-1"/>
  </xjdf:Resource>
  <xjdf:Resource>
    <xjdf:Part BinderySignatureID="BS-2" ProductPart="BODY-ID" />
    <xjdf:BinderySignature FoldCatalog="F16-7"/>
  </xjdf:Resource>
  <xjdf:Resource>
    <xjdf:Part BinderySignatureID="BS-3" ProductPart="BODY-ID" />
    <xjdf:BinderySignature FoldCatalog="F4-1"/>
  </xjdf:Resource>
</xjdf:ResourceSet>
<xjdf:ResourceSet Name="Component" Usage="Input">
  <xjdf:Resource ExternalID="COMP-BS-1">
    <xjdf:Part BinderySignatureID="BS-1" />
    <xjdf:Component Dimensions="852.68 603.67 1.42" MediaRef="MEDIA-COVER-ID" />
  </xjdf:Resource>
  <xjdf:Resource ExternalID="COMP-BS-2">
    <xjdf:Part BinderySignatureID="BS-2" />
    <xjdf:Component Dimensions="852.68 603.67 3.34" MediaRef="MEDIA-BODY-ID" />
  </xjdf:Resource>
  <xjdf:Resource ExternalID="COMP-BS-3">
    <xjdf:Part BinderySignatureID="BS-3" />
    <xjdf:Component Dimensions="852.68 603.67 0.92" MediaRef="MEDIA-BODY-ID" />
  </xjdf:Resource>
</xjdf:ResourceSet>

```

[Continued on next row...]

```

<xjdf:ResourceSet Name="Component" Usage="Output">
  <xjdf:Resource ExternalID="COMP-MAGAZINE">
    <xjdf:AmountPool>
      <xjdf:PartAmount Amount="1050" />
    </xjdf:AmountPool>
    <xjdf:Component Dimensions="841.88 595.27 5.26" />
  </xjdf:Resource>
</xjdf:ResourceSet>
<xjdf:ResourceSet Name="StitchingParams" Usage="Input">
  <xjdf:Resource>
    <xjdf:StitchingParams StitchType="Saddle" StapleShape="Eyelet"/>
  </xjdf:Resource>
</xjdf:ResourceSet>
<xjdf:ResourceSet Name="TrimmingParams" Usage="Input">
  <xjdf:Resource>
    <xjdf:TrimmingParams Height="841.88" Width="595.27"/>
  </xjdf:Resource>
</xjdf:ResourceSet>
<xjdf:ResourceSet Name="Media">
  <xjdf:Resource ID="MEDIA-COVER-ID">
    <xjdf:Media MediaType="Paper" Thickness="150" Weight="135" ISOPaperSubstrate="PS3" />
  </xjdf:Resource>
  <xjdf:Resource ID="MEDIA-BODY-ID">
    <xjdf:Media MediaType="Paper" Thickness="90" Weight="80" ISOPaperSubstrate="PS3" />
  </xjdf:Resource>
</xjdf:ResourceSet>
<xjdf:ResourceSet Name="NodeInfo" Usage="Input">
  <xjdf:Resource>
    <xjdf:NodeInfo Start="2016-09-26T22:00:00Z" />
  </xjdf:Resource>
</xjdf:ResourceSet>
</xjdf:XJDF>

```

Product Description

Introduction

XJDF Essentials

Product Description

Process Description

ICS

XJMF Messaging

Authoring Process



...coming soon



ICS

Introduction

XJDF Essentials

Product Description

Process Description

ICS

XJMF Messaging

Authoring Process



ICS - Interoperability Conformance Specifications

ICS Documents describe specific standardized interfaces in a print production workflow. Samples of ICS Documents are:

- MISCS
(**MIS** to **C**onventional **P**rinting - **S**heet-Fed ICS)
- MISFin
(**MIS** to **F**inishing ICS)
-



ICS - GrayBoxes

Technically, each ICS Document describes one or multiple GrayBoxes. It describes the minimal processes required in addition to the appropriate input and output resources.

6	Conformance Tables – Gray Boxes	9
6.1	Abstract GB MISFin	9
6.2	GB <i>BoxMaking</i>	12
6.3	GB <i>HardcoverFin</i>	13
6.4	GB <i>InsertFin</i>	17
6.5	GB <i>SheetFin</i>	18
6.6	GB <i>SoftcoverFin</i>	19
6.7	GB <i>StitchFin</i>	22



Definition of ICS conformity in an XJDF Document

XJDF Documents which conform to a certain Gray Box of an ICS Document also have to specify the attributes @ICSVersions and @Category:

```
<xjdf:XJDF JobID="JOB-1234" Types="Collecting Stitching Trimming"
  Category="MISFin.StitchFin" ICSVersions="MISFin_L1-2.0"
  xmlns:xjdf="http://www.CIP4.org/JDFSchema_2_0">
  [...]
</xjdf:XJDF>
```

Format values:

@Category: <ICSName>.<GrayBoxName>

@ICSVersions: <ICSName>_L<ICSLevel>-<ICSVersion>



Systems are NOT compatible to XJDF but rather to ICS Documents !!

- **While XJDF is just a common foundation, the real standardization is carried out by the ICS Documents**
- **Standard XJDF interfaces are specified in ICS Documents. XJDF is the overall clamp.**



XJMF Messaging

Introduction

XJDF Essentials

Product Description

Process Description

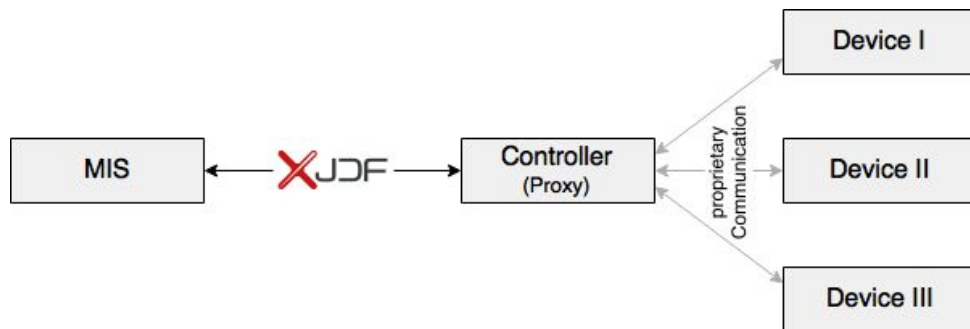
ICS

XJMF Messaging

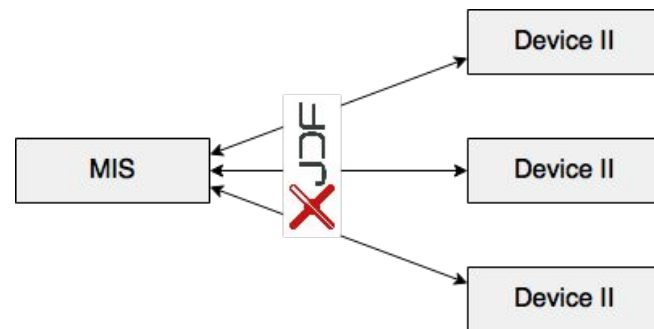
Authoring Process



System Architectures



System Architecture WITH Controller



System Architecture WITHOUT Controller



Typical use of XJMF

XJMF is the messaging system coming with XJDF in order to have a communication infrastructure between the MIS, the Controllers and the Devices.

Typical use cases for XJMF include but are not limited to:

- System setup
- Dynamic status and error tracking for Jobs and Devices
- Pipe control
- Device setup and Job changes
- Queue handling and Job submission
- Device Capability description



Message Families

Each XJMF Message belongs to one of the four message families:

- **Query Message** - Receiving information without changing the state
- **Command Message** - Receiving information AND changing the state
- **Signal Message** - Asynchronous respond to a Query Subscription
- **Response Message** - Synchronous respond to a Query or Command



XJMF Query Message

Receiving information without changing the state. The KnownDevices Query Message requests information about the Devices that are controlled by a Controller.

Further specific samples of XJMF Query Messages are:

- QueryNotification
(Query for Notifications of a devices or controller)
- QueryStatus
(Queries the status of a device, controller or job)
- QueryKnownMessages
(Queries list of all messages supported)

```
<xjdf:XJMF xmlns:xjdf="http://www.CIP4.org/JDFSschema_2_0">
  <xjdf:Header ID="XJMF_ID" DeviceID="DEVICE_ID_MIS"
    Time="2016-11-10T21:35:14+02:00" />
  <xjdf:QueryKnownDevices>
    <xjdf:Header ID="MESSAGE_ID" DeviceId="DEVICE_ID_MIS"
      Time="2016-11-10T21:35:14+02:00" />
    <xjdf:DeviceFilter DeviceDetails="Brief" />
  </xjdf:QueryKnownDevices>
</xjdf:XJMF>
```



XJDF Response Message

Synchronous respond to a Query or Command. This message is a response message to the KnownDevices Query.

Further specific samples of XJMF Response Messages are:

- ResponseStatus
(Status details of a device or controller)
- ResponseShutDown
(Response of the appropriate command)

```
<xjdf:XJMF xmlns:xjdf="http://www.CIP4.org/JDFSschema_2_0">
  <xjdf:Header ID="XJDF_ID" DeviceID="DEVICE_ID_CONTROLLER"
    Time="2016-11-11T11:51:14+02:00" />
  <xjdf:ResponseKnownDevices ReturnCode="0">
    <xjdf:Header ID="42" refID="MESSAGE_ID"
      DeviceID="DEVICE_ID_CONTROLLER"
      Time="2016-11-11T11:51:14+02:00" />
    <xjdf:Device DeviceID="PRESS-01" />
    <xjdf:Device DeviceID="PRESS-02" />
    <xjdf:Device DeviceID="PRESS-03" />
  </xjdf:ResponseKnownDevices>
</xjdf:XJMF>
```



XJMF Command Message

Receiving information AND changing the state. The following message for instance shuts down a device or controller (StandBy Mode).

Further specific samples of XJMF Query Messages are:

- **CommandWakeUp**
(Wakes up a device that is in standby mode)
- **CommandSubmitQueueEntry**
(Submits a Job in order to be executed)

```
<xjdf:XJMF xmlns:xjdf="http://www.CIP4.org/JDFSschema_2_0">
  <xjdf:Header ID="XJMF_ID" DeviceID="DEVICE_ID_MIS"
    Time="2016-11-11T13:03:38+02:00" />
  <xjdf:CommandShutDown>
    <xjdf:Header ID="MESSAGE_ID" DeviceID="DEVICE_ID_MIS"
      Time="2016-11-11T13:03:38+02:00" />
    <xjdf:ShutDownCmdParams ShutDownType="StandBy"/>
  </xjdf:CommandShutDown>
</xjdf:XJMF>
```



XJMF Signal Message

Asynchronous respond to a Query Subscription. This message returns status details from a device, proxied by a controller:

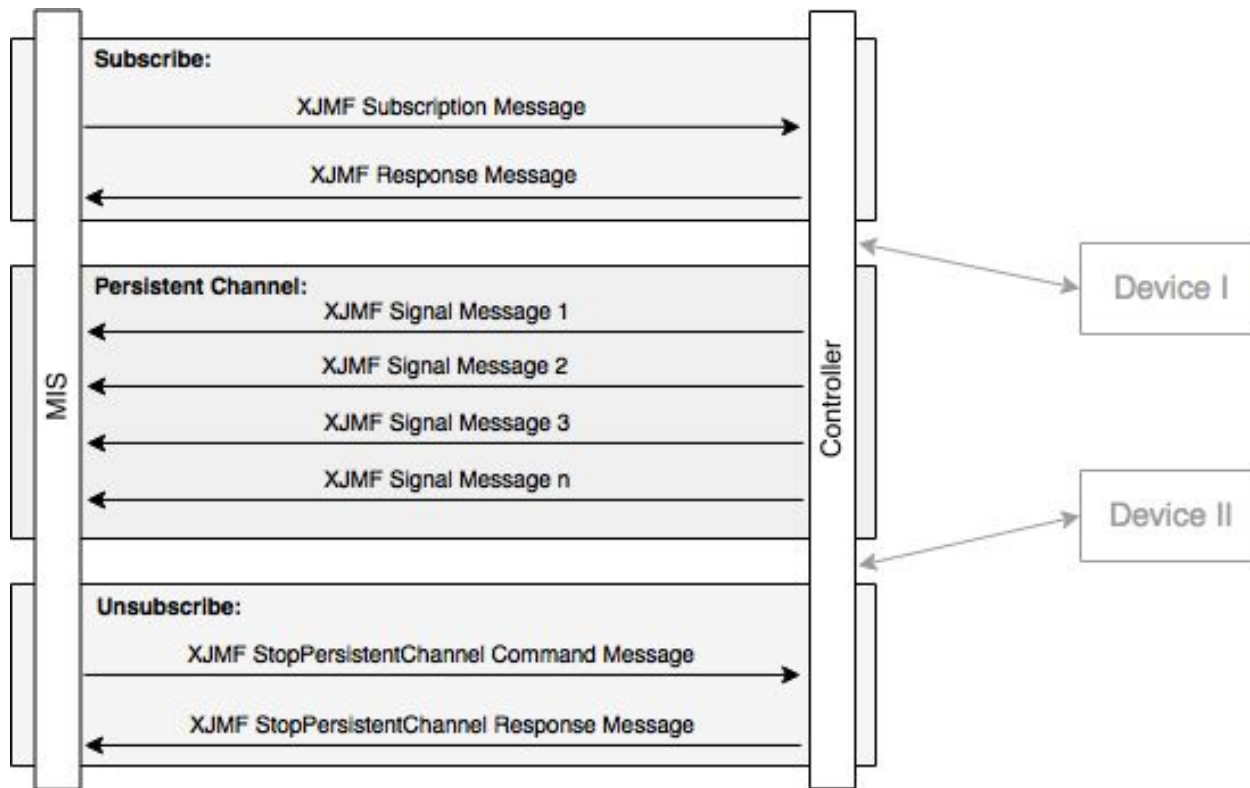
Further specific samples of XJMF Query Messages are:

- **SignalNotification**
(Publish Notification messages)
- **SignalResource**
(Publish Resource consumptions)

```
<xjdf:XJMF xmlns:xjdf="http://www.CIP4.org/JDFSchema_2_0">
  <xjdf:Header ID="XJMF_ID" DeviceID="DEVICE_ID_CONTROLLER"
    Time="2016-11-12T12:11:39+02:00" />
  <xjdf:SignalStatus ReturnCode="0">
    <xjdf:Header ID="1234" refID="MESSAGE_ID"
      DeviceID="DEVICE_ID"
      Time="2016-11-12T12:11:39+02:00" />
    <xjdf:DeviceInfo DeviceID="DEVICE_ID" Speed="12500"
      ProductionCounter="5425" Status="Production" />
  </xjdf:SignalStatus>
</xjdf:XJMF>
```



Concept of Subscriptions



Subscription

Subscribe to Status Details

The Subscription Element in the Query Message, makes a Query to a Subscription. When receiving, a new Persistent Channel has to be created on the addressed controller or device.

```
<xjdf:XJMF xmlns:xjdf="http://www.CIP4.org/JDFSchemas_2_0">
  <xjdf:Header ID="XJMF_ID" DeviceID="DEVICE_ID_MIS"
    Time="2016-11-15T13:22:01+02:00" />
  <xjdf:QueryStatus>
    <xjdf:Header ID="MESSAGE_ID" DeviceID="DEVICE_ID_MIS"
      Time="2016-11-15T13:22:01+02:00" />
    <xjdf:StatusQuParams EmployeeInfo="true" JobDetails="true" />
    <xjdf:Subscription RepeatTime="30"
      URL="http://mis.example.org/xjmf" />
  </xjdf:QueryStatus>
</xjdf:XJMF>
```



Subscription

Unsubscribe to Status Details

The StopPersistentChannel Command Message removes the Persistent Channel for a specific subscriber and unsubscribes a subscriber from a controller or device:

```
<xjdf:XJMF xmlns:xjdf="http://www.CIP4.org/JDFSchema_2_0">
  <xjdf:Header ID="XJMF_ID" DeviceID="DEVICE_ID_MIS"
    Time="2016-11-15T15:26:59+02:00" />
  <xjdf:CommandStopPersistentChannel>
    <xjdf:Header ID="MESSAGE_ID" DeviceID="DEVICE_ID_MIS"
      Time="2016-11-15T15:26:59+02:00" />
    <xjdf:StopPersChParams DeviceID="DEVICE_ID_MIS"
      URL="http://mis.example.org/xjmf" />
  </xjdf:CommandStopPersistentChannel>
</xjdf:XJMF>
```



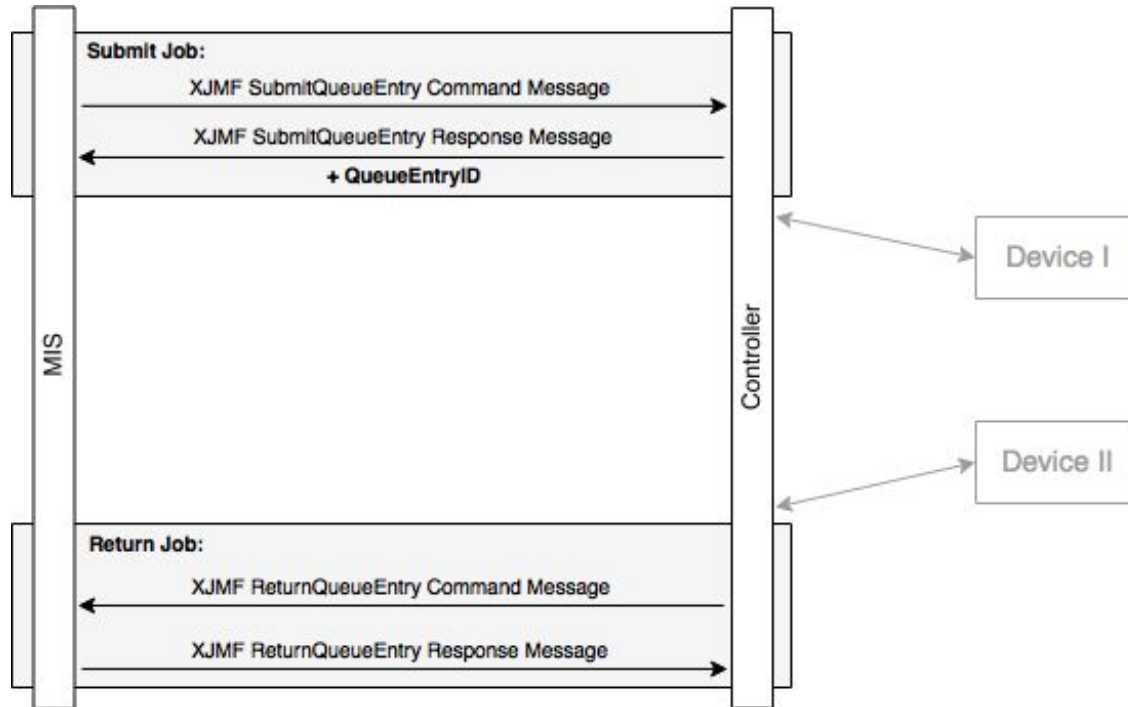
Job Submission

The following XJMF Command Messages are for job submission and organization:

- **CommandSubmitQueueEntry**
A Job is submitted to a queue in order to be executed.
- **CommandReturnQueueEntry**
Returns a Job to the controller originally submitted the job.
- **CommandResubmitQueueEntry**
Replaces a queue entry without affecting the entry's parameters.
- **CommandModifyQueueEntry**
Modifies the properties of one or more QueueEntry elements.
- **CommandRequestQueueEntry**
A new Job is requested by the Device



Job Submission



Job Submission

Sample: Submit Job

XJMF Command:

```
<xjdf:XJMF xmlns:xjdf="http://www.CIP4.org/JDFSchema_2_0">
  <xjdf:Header ID="XJMF_ID" DeviceID="DEVICE_ID_MIS"
    Time="2016-11-12T12:11:39+02:00" />
  <xjdf:CommandSubmitQueueEntry>
    <xjdf:Header ID="MESSAGE_ID" DeviceID="DEVICE_ID_MIS"
      Time="2016-11-12T12:11:39+02:00" />
    <xjdf:QueueSubmissionParams
      URL="http://mis.example.org/xjdf/job-1234.xjdf"
      ReturnJMF="http://mis.example.org/xjmf" Priority="42"/>
  </xjdf:CommandSubmitQueueEntry>
</xjdf:XJMF>
```



XJMF Response:

```
xjdf:XJMF xmlns:xjdf="http://www.CIP4.org/JDFSchema_2_0">
  <xjdf:Header ID="XJMF_ID" DeviceID="DEVICE_ID_CONTROLLER"
    Time="2016-11-12T12:11:39+02:00" />
  <xjdf:ResponseSubmitQueueEntry ReturnCode="0">
    <xjdf:Header ID="1234" refID="MESSAGE_ID"
      DeviceID="DEVICE_ID_CONTROLLER"
      Time="2016-11-12T12:11:39+02:00"/>
    <xjdf:QueueEntry QueueEntryID="1" Priority="42"
      Status="Waiting" />
  </xjdf:ResponseSubmitQueueEntry>
</xjdf:XJMF>
```



Job Submission

Sample: Return Job

XJMF Response:

```
<xjdf:XJMF xmlns:xjdf="http://www.CIP4.org/JDFSchema_2_0">
  <xjdf:Header ID="XJMF_ID" DeviceID="DEVICE_ID_CONTROLLER"
    Time="2016-11-12T12:11:39+02:00" />
  <xjdf:ResponseReturnQueueEntry ReturnCode="0">
    <xjdf:Header ID="1234" refID="MESSAGE_ID"
      DeviceID="DEVICE_ID_CONTROLLER"
      Time="2016-11-12T12:11:39+02:00" />
  </xjdf:ResponseReturnQueueEntry>
</xjdf:XJMF>
```

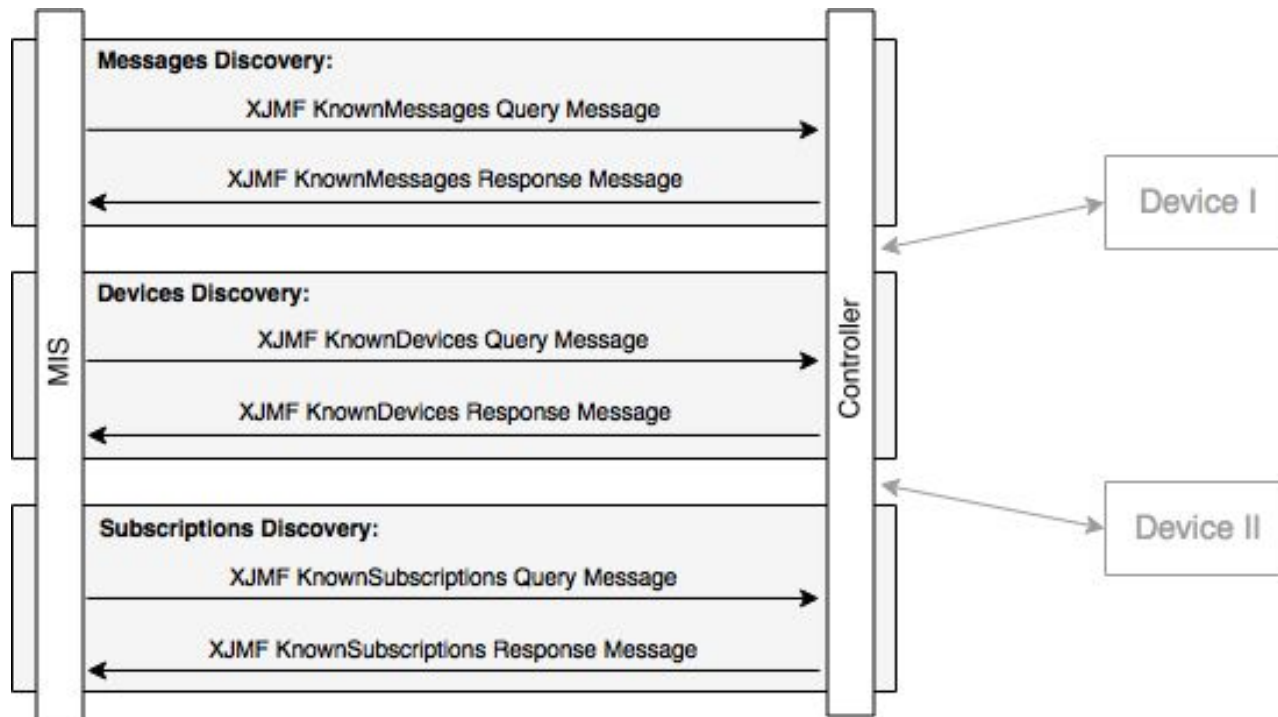
XJMF Command:

```
<xjdf:XJMF xmlns:xjdf="http://www.CIP4.org/JDFSchema_2_0">
  <xjdf:Header ID="XJMF_ID" DeviceID="DEVICE_ID_CONTROLLER"
    Time="2016-11-12T12:11:39+02:00" />
  <xjdf:CommandReturnQueueEntry>
    <xjdf:Header ID="MESSAGE_ID" DeviceID="DEVICE_ID_CONTROLLER"
      Time="2016-11-12T12:11:39+02:00" />
    <xjdf:ReturnQueueEntryParams QueueEntryID="1"
      URL="http://controller.example.org/xjdf/job-1234.xjdf" />
  </xjdf:CommandSubmitQueueEntry>
</xjdf:XJMF>
```



XJMF Bootstrapping

“The XJMF Handshake”



Authoring Process

Introduction

XJDF Essentials

Product Description

Process Description

ICS

XJMF Messaging

Authoring Process



The CIP4 Authoring Process

Everyone is able to participate at the development of the CIP4 Specifications - only a Visitor Account is required.

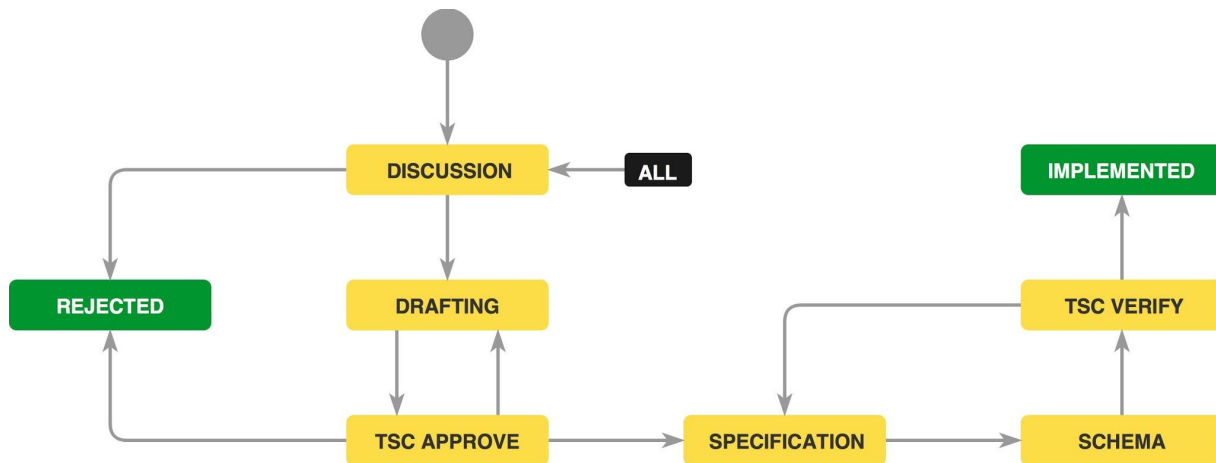
All discussions are lead openly in CIP4s JIRA System (<http://jira.cip4.org>):

Specification	Link
XJDF Specification	https://jira.cip4.org/projects/XJDF
JDF Specification	https://jira.cip4.org/projects/JDF
PrintTalk Specification	https://jira.cip4.org/projects/PTK
...	...



The Spec Editing Process

The Spec Editing Process is an integrated process from the raw idea to the implementation in one of CIP4s Specifications:



CIP4 Meetings

- **Web Conferences**
Scheduled biweekly
(<https://confluence.cip4.org/display/PUB/Calendar>)
- **InterOp Meetings**
Twice a year - USA, Europe, Japan
(<https://confluence.cip4.org/display/PUB/InterOp>)
- **Face2Face Meetings**
Individual Workgroup Meetings

Activity

All **Comments** Work Log History Activity

Stefan Meissner added a comment - 5 days ago - [edited](#)

#XJDF WG 2016-11-10:
Amounts do not belong to NodelInfo. Amounts are part of the appropriate (physical) Input and Output Resources.
New Issue [XJDF-861](#) for waste types.



The XJDF Book



This presentation is based on the **XJDF Book**, which is available here:



<http://ricebean.net/xjdf>





XJDF Book

<http://ricebean.net/xjdf>

The technical introduction to XJDF.



Thank you.

