

Caldera JDF specification

[Changelog](#)

[Overview](#)

[Caldera 9.20](#)

[Caldera V10](#)

[Caldera V11](#)

[Caldera JDF Specification](#)

[JDF job \(process\)](#)

[JDF Node](#)

[Comments](#)

[ResourcePool](#)

[ResourceLinkPool](#)

[AuditPool](#)

[Caldera extension to JDF](#)

[JMF Messages](#)

[Commands](#)

[Queries](#)

[Persistent channels & Signals](#)

[JMF Examples](#)

[Job submission scenarios](#)

[Direct JDF submission by HTTP](#)

[Direct JDF submission in hotfolder](#)

[JMF job submission by HTTP](#)

[JMF job submission in hotfolder](#)

[MIME multipart packaging](#)

[Job submission as response to RequestQueueEntry JMF message](#)

[Active \(JMF-compliant\) RequestQueueEntry submission](#)

[Passive submission](#)

[Full working JDF example](#)

Changelog

19/09/2013	Initial version (Caldera9.20)
19/11/2013	Added FileSpecRef, RunList/@Pages, AuditPool/Modified. New chapter "Caldera extension to JDF". (9.20.131112 + BridgeMisc2)
22/11/2013	Added cal:Status/@FID. Changed cal:Status/contents tile and page to 0-based indexes. (9.20.131112 + BridgeMisc3)
04/12/2013	Caldera can now print CustomerID and CustomerOrderID. Print configuration in JDF can now be used in Caldera when using workflows. (9.20.131112 + BridgeMisc4)
Oct. 2014	Added JMF Queries. Added ComponentLink/@Amount and @ActualAmount (V10). Added persistent channels (Status & QueueStatus). Added cal:Status/thumbnail. Added MIME multipart messages.
23/03/2015	Added Layout resource (V10.1)
04/06/2015	Added GangName (V10.1)
04/04/2017	Added ReturnJMF support to SubmitQueueEntry command (V11.1)

Overview

JDF implementation in Caldera will be enhanced in future versions. This document will be completed as the work progresses.

When designing the JDF/JMF data, use the JDF validator tool at <http://jdfutility.cip4.org/JDFUtility/>. Check “Ignore Private extensions” option if the JDF/JMF ticket uses any <cal:...> nodes.

Caldera 9.20

This first version has limited JDF/JMF /support.

Quick Caldera 9.20 bridge feature overview:

- JMF unidirectionnal and bidirectionnal messaging, limited to job submission (HTTP and hotfolders), based on ICS JMF 1.4 and JDF Specification 1.4 (<http://www.cip4.org>)
- Support for single job JDF (DigitalPrinting) and multiple job JDF (ProcessGroup), based on ICS Base 1.4 and JDF Specification 1.4 (<http://www.cip4.org>)
- JDF job ticket contents:
 - Links to one or more source image files
 - Printing parameters (printer, media, resolution...) using Caldera extension to JDF format.
 - Customer information.
 - Comments.
 - Priority information (tags, due date).
- Fully automated job processing with status feedback
- Operator-driven manual job processing using the JobTracker module
- Detailed status feedback (AuditPool) using Caldera extension to the JDF format.

Caldera V10

V10 adds the following features:

- New JMF commands and queries
 - KnownMessages
 - KnownDevices
 - KnownSubscriptions
 - SubmissionMethods
 - Status
 - QueueStatus
 - StopPersistentChannel
 - RemoveQueueEntry

- AbortQueueEntry

JDF-related information can be displayed in Caldera in one of the following places:

- Annotations
- QRcode
- Free text in the crop marks.

To achieve that please insert one or more of the following tags in the text of the places listed above (please note that the QRcode editor allows to insert the tags by a contextual right-button mouse menu. This feature will be added to annotations and the free text in the future):

- **@(CustomerID)**: Customer ID, as specified in CustomerInfo/@CustomerID.
- **@(CustomerOrderID)**: Order ID, as specified in CustomerInfo/@CustomerOrderID.
- **@(JobTicket)**: This is the value of JDF/@JobID
- **@(Customer)**: Customer name, taken from CustomerInfo/Contact with ContactTypes = "Customer"
- **@(Sales)**: Sales name, taken from CustomerInfo/Contact with ContactTypes = "Sales".
- **@(DeviceID)**: Printer name as specified in Device/@DeviceID.
- **@(GangName)**: Gang name of the job, if specified during submission.

JDF tickets can now specify a printer and a quickprint configuration name in Device/@DeviceID resource.

Caldera V11

- Nexio is now able to send SubmissionMethod and ReturnQueueEntry JMF messages (mainly in relation with QueueSubmissionParams/@ReturnJMF feature) (in V11.1)
- JDF tickets can now contain partial print configuration when a quickprint configuration is selected using @DeviceID (V11.1)
- Completed JDF tickets returned via ReturnURL / ReturnJMF can now contain the print configuration of the job in the cal:Status node in AuditPool (V11.1)

Caldera JDF Specification

The following specification uses the XPath XML format description, which is ParentNode/ChildNode/@AttributeName. In the description tables below, nodes and attributes in bold are required. Attributes not in bold can be however required in some cases. If so, it is detailed in the value description.

JDF job (process)

JDF Node

A JDF ticket must have one root JDF node. It can have child JDF nodes in some cases (see JDF/@Type below).

JDF/@ID	Unique ID of JDF node. Used to refer to the JDF node.
JDF/@JobID	Optional, required in root node. Internal job ID used by the Producer of the JDF (MIS?)
JDF/@JobPartID	Optional, required in root node. Identifiant of the node within the job. Internal ID of Producer.
JDF/@Status	Status of the node <ul style="list-style-type: none">• Waiting: Node can be executed, has not been tested.• Ready: Node can be executed and was tested (ressources and parameters are available and tested)• InProgress: node is executing• Completed: Node executed and finished correctly.• Aborted: Node aborted (error or manual abort) Other values exist, but are not used by Caldera.
JDF/@Type	Node type <ul style="list-style-type: none">• "Combined": Process node that covers more than one Process, when a single device can have several roles (listed in JDF/@Types). Should not have nested JDF nodes (?)• "ProcessGroup": intermediate level, describes a group of Processes (nested JDF nodes)• "Product": Highest level, describes completely a product, can have nested ProcessGroups and/or Processes (nested JDF nodes)• Other value: Process job type, lowest level in a job hierarchy. Must NOT contain nested JDF nodes. The

	values to be used are those of @Types. Example: "DigitalPrinting" (currently the only type recognized by Caldera). Required resource nodes vary according to this value.
JDF/@Types	Optional (@Type=ProcessGroup), Required (@Type=Combined), Ignored else. List of values used for @Type. Example: "Rendering Screening DigitalPrinting Cutting". The order defines the order of execution. Currently checked only for "DigitalPrinting"
JDF/@Version	Version of JDF ICS. Optional for child JDF nodes, required for root node. Currently "1.4"
JDF/@ICSVersions	Required in root node. <ul style="list-style-type: none"> • "Base_L1-1.4": conformance level 1 • "Base_L2-1.4": conformance level 2
JDF/@xmlns	Currently " http://www.CIP4.org/JDFSchemas_1_1 ". Required in root JDF node, optional elsewhere.
JDF/@xmlns:xsi	Required in root node. "http://www.w3.org/2001/XMLSchema-instance"
JDF/@xmlns:cal	Caldera specific namespace. Required in root node if the JDF uses Caldera namespace. "http://www.caldera.com/jdf"
JDF/@DescriptiveName	Optional, but required in root JDF node . Human-readable one line string, job title.
JDF/@Activation	Optional, default value is "Active". <ul style="list-style-type: none"> • "Active": job is active. Only value that allows the job to be run. • "Held": job is held. • other: Interpreted as "Inactive", job will not be run.
JDF/AuditPool	Contains audit information (errors, messages, log, notifications...)
JDF/ResourcePool	Container for resources.
JDF/ResourceLinkPool	Describes input and output resources of the node.

JDF job of "DigitalPrinting" type must link at least one of following resources:

- RunList (Input)
- DigitalPrintingParams (Input)
- Component (Output)

```

<?xml version="1.0" encoding="UTF-8"?>
<JDF ID="JDF0343322" Category="DigitalPrinting" Status="Ready"
    Type="Process" Version="1.4"
    xmlns="http://www.CIP4.org/JDFSSchema_1_1"
    xmlns:cal="http://www.caldera.com/jdf"
    DescriptiveName="My JDF job" >
  <ResourcePool>...</ResourcePool>
  <ResourceLinkPool>...</ResourceLinkPool>
  <AuditPool>...</AuditPool>
</JDF>

```

Comments

Comments can be added at any place. Comments must not be nested. Comments that are displayed in Remote Browser are only those placed directly in nodes that represent a displayed item, that is:

- Comment placed directly in a JDF node which is displayed.
- Comment directly in a resource which is displayed (RunList?)
- Comment placed directly in LayoutElement or FileSpec which is displayed.

For usage in Caldera, a job can define a Comment with name of “Tags” to define tags that can be searched in Remote Browser GUI. The tags are simply words separated by spaces.

Another item specific to Caldera is the “DueDate” comment, which defines a priority. The comment contents is a time in ISO8601:2004 format.

Comment/@Name	A name that defines the usage of a comment. “Description”, “Instruction”, “JobDescription” (only in JDF or CustomerInfo nodes), “OperatorText”, “UserText”. Specific to Caldera: “Tags”
Comment/	text of the comment

Example of a comment:

```

<Comment Name="JobDescription">Stickers for Tartenpion&Co</Comment>
<Comment Name="Tags">STICKERS URGENT</Comment>
<Comment Name="DueDate">2013-04-17T08:00:00Z</Comment>

```

ResourcePool

RunList: Ordered set of LayoutElement nodes. Allows to create a virtual document spread over

multiple files, or produce multiple documents stored in the same file. Can have nested RunList elements. In that case the order is defined by attribute defined in @PartIDKey of the root RunList node.

RunList/@Class	"Parameter". Required only in root RunList node. Other classes currently ignored.
RunList/@ID	Required only in root RunList node. Unique RunList identifier.
RunList/@Status	Required in root RunList node. Optional in children. "Available" = resource can be used. All other values currently interpreted as resource can not be used.
RunList/@Directory	Optional. Allows to use relative URL in FileSpec nodes. Must be absolute URI. Note: The Directory attribute has no effect with FileSpecRef. The FileSpec object referred to by FileSpecRef must have complete URL.
RunList/@DocCopies	Optional. Default = 1. Specifies desired number of copies of the whole RunList.
RunList/@Pages	Optional. Selection of pages to load/print. The parameter is a list of pages or ranges separated by spaces. A range is defined by ~ character. Default = "0 ~ -1" (all pages). Warning, page numbering starts at 0 (which is the first page). Unselected pages are not loaded in Caldera at all. Ranges and single pages can be mixed (example: "0 2 ~ 5 7 8"). Note: pay attention to spaces in ranges. The character ~ must be surrounded by spaces. See example in FileSpecRef definition. See important note below the table.
RunList/@PartIDKeys	Optional, only used with partitioned resources (in which case this is required), in root RunList node. Indicates the attribute of children nodes to identify the parts. Use "Run".
RunList/@Run	Used only in ordered partitioned RunList nodes, to identify a run. See examples below.
RunList/@DescriptiveName	Optional. Human-readable one line string.
RunList/LayoutElement	Child element containing the file. Can have more than one of those (unordered list)

Important note: Currently there is a limitation when specifying variable number of copies per page in a multi-page document. Be sure to always have only one FileSpec node pointing to the same document and use FileSpecRef to refer to it. Having two FileSpec in a job pointing to the

same file can cause the bridge to count printed copies badly in some cases.

LayoutElement: Contained in RunList to specify a file.

LayoutElement/FileSpec	Child element to specify the file (see below). Can contain more than one of these (unordered list)
LayoutElement/FileSpecRef	Reference to a file (FileSpec). Used when the same file is referenced more than once in the job (to select specific pages for example)

FileSpec: Specify external file. Currently Containers, Compression and aliases not supported. There should be exactly one FileSpec in a LayoutElement.

FileSpec can also be used directly in ResourcePool (without being linked in ResourceLinkPool). In that case it serves only as reference for FileSpecRef nodes in linked RunList and must have @ID, @Class and @Status attributes.

FileSpec/@URL	Location of the file. Can be absolute http/file or relative file URI. Relative path to RunList/@Directory (only file://).
FileSpec/@UserFileName	Optional. User-friendly file name. If not specified, the name is extracted from the URL.
FileSpec/@CheckSum	Optional. MD5 checksum of the file. Currently not verified.
FileSpec/@Class	"Parameter". Must be specified ONLY when placed directly in ResourcePool (for use with FileSpecRef).
FileSpec/@ID	Unique resource identifier to be specified in rRef attribute of FileSpecRef. Must be specified ONLY when placed directly in ResourcePool (for use with FileSpecRef).
FileSpec/@Status	"Available" = resource can be used. All other values currently interpreted as resource can not be used. Must be specified ONLY when placed directly in ResourcePool (for use with FileSpecRef).

Single file example:

```
<ResourcePool>
  <RunList Class="Parameter" Directory="file://blah/blah/" ID="RL001"
    Status="Available">
    <LayoutElement><FileSpec URL="file1.pdf"/></LayoutElement>
```

```

    </RunList>
</ResourcePool>

```

Multiple ordered files example with cal:PrintConfig:

```

<ResourcePool>
  <RunList Class="Parameter" ID="RL001" PartIDKeys="Run"
    Status="Available">
    <RunList Run="1">
      <LayoutElement><FileSpec URL="http://blah/file1.pdf"/></LayoutElement>
    </RunList>
    <RunList Run="2">
      <LayoutElement><FileSpec URL="http://blah/givefile.php?file2.pdf">
      </LayoutElement>
    </RunList>
    <DigitalPrintingParams Class="Parameter" ID="DPP221" Status="Available">
      <cal:PrintConfig>
        <modename>CMYK</modename>
        <res_id idx="0" res="200">200x200</res_id>
      </cal:PrintConfig>
    </DigitalPrintingParams>
  </RunList>
</ResourcePool>

```

FileSpecRef: Reference to a FileSpec resource. Must be placed in a LayoutElement. This is used when you need to reference the same file multiple times, for example to select specific pages in a multi-page document.

FileSpecRef/@rRef	ID of a FileSpec resource.
-------------------	----------------------------

Example of page selection in a multipage document: (expecting 2 copies of page 1,2 and 3 and 1 copy of page 4)

```

<ResourcePool>
  <FileSpec ID="FSPEC001" Class="Parameter" Status="Available"
    URL="file:///input/1234.pdf"/>

  <RunList Class="Parameter" ID="RL001" Status="Available">
    <RunList DocCopies="2" Pages="0 ~ 2">
      <LayoutElement>
        <FileSpecRef rRef="FSPEC001"/>
      </LayoutElement>
    </RunList>
    <RunList Pages="3">
      <LayoutElement>

```

```

        <FileSpecRef rRef="FSPEC001"/>
    </LayoutElement>
</RunList>
</RunList>
</ResourcePool>

```

Layout: (Added in Caldera 10.1) Resource allowing to define a Compose document using individual pages and input files from RunList resource. Individual objects are placed by specifying a rotation (steps of 90 degrees only!), scaling and displacement. Currently rotations not aligned on 90 degrees and using axis symmetry (flip) are not supported and will cause the job to be set in error.

Compatibility note: JDF specification (1.5) does not allow to link multiple Layout resources in a single ResourceLinkPool. However Caldera does support that possibility for user's convenience. Every linked Layout will be treated as separate compose document.

Layout defines a document surface to hold the individual items and a list of ContentObject items, each corresponding to a page of a file. All dimensions are specified in millimeters. Each ContentObject is linked to a selected page of the RunList by specifying the "Ord" index. The Ord index takes into account partitioned RunLists (@PartIDKeys = "Run").

Example: RunList contains 2 files with following page specifications:

- file1: Pages="0 1 3" (we want to print only page 1, 2 and 4 of file1)
- file2: (file2 has 2 pages, "Pages" not specified thus we want to print all pages)

The "Ord" attribute in ContentObject references the following pages:

- Ord=0: page 1 of file1
- Ord=1: page 2 of file1
- Ord=2: page 4 of file1
- Ord=3: page 1 of file2
- Ord=4: page 2 of file2

Layout/@Class	"Parameter"
Layout/@ID	Unique ID.
Layout/@Status	"Available"
Layout/@SurfaceContentsBox	"0 0 <doc width> <doc height>" Defines the size of the compose document (in mm). The parameter defines a rectangle "lower-left-x lower-left-y upper-right-x upper-right-y". However lower left x and y should be 0. This parameter is not

	used to position the document, only to set its size. Thus, if the parameter does not start with 0 0, the real size is obtained by subtracting the lower left point to the upper-right one.
Layout/@DescriptiveName	Optional layout name. This allows to set the name of the compose document that will be created. If not specified, JDF/DescriptiveName is used. If that one is not defined either, the document will be named "Layout".
Layout/ContentObject/@CTM	"a b c d e f" Content transformation matrix. See explanations below.
Layout/ContentObject/@Ord	Index of matching page from the RunList (see above)

Content Transformation Matrix: Matrix to define geometric transformation to place all objects.

The first four parameters "a b c d" define a 2x2 matrix with rotation and scale. The latter two parameters "e f" define a displacement vector (e=dx f=dy). The final position of an object is obtained by applying the CTM to all 4 corners of an object. The initial position of the object is lower left point = (0,0), upper right point = (page width at 100%, page height at 100%). The operation is done by multiplying the CTM by the vector which define a point. In what follows (x1,y1) are coordinates of a point before transformation, (x2, y2) are coordinates of the point after transformation. The coordinate system has the origin (0,0) as the lower left point, x axis increasing to the right and y axis increasing to the top. All displacements and dimensions are in millimeters.

$$\begin{bmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{bmatrix} \times \begin{pmatrix} x1 \\ y1 \\ 1 \end{pmatrix} = \begin{pmatrix} x2 \\ y2 \\ 1 \end{pmatrix}$$

$$x2 = a \times x1 + c \times y1 + e$$

$$y2 = b \times x1 + d \times y1 + f$$

The following table lists the most common cases. The resulting object is placed with its lower left corner in point defined by (X, Y). W and H are respectively the original (unscaled) image width and height. Xsc and Ysc are X and Y scale (1 = 100%, 0.5 = 50%, 2 = 200%...)

No rotation	CTM = "Xsc 0 0 Ysc X Y"
Rotation 90 degrees clockwise	CTM = "0 -Xsc Ysc 0 X (Y+Xsc*W)"
Rotation 180 degrees	CTM = "-Xsc 0 0 -Ysc (X+Xsc*W) (Y+Ysc*H)"

Rotation 90 degrees counter-clockwise	CTM = "0 Xsc -Ysc 0 (X+Ysc*H) Y"
---------------------------------------	----------------------------------

Example: Definition of a document of 300x200 mm, with 3 objects: 2 copies of page 1 and one copy of page 2. Every page is 50x40 mm. The first object is the page 1, placed with no transformations in the lower left corner of the document. The second object is the page 1 again, this time rotated 90 degrees clockwise, placed in the right upper corner of the document. The third object is page 2, scaled at 50%, placed in the left upper corner of the document.

```
<ResourcePool>
  <Layout Class="Parameter" ID="Lay01" Status="Available"
    SurfaceContentsBox="0 0 300.0 200.0" DescriptiveName="My compose">
    <ContentObject CTM="1 0 0 1 0 0" Ord="0"/>
    <ContentObject CTM="0 -1 1 0 260.0 200.0" Ord="0"/>
    <ContentObject CTM="0.5 0 0 0.5 0 180" Ord="1"/>
  </Layout>
</ResourcePool>
```

DigitalPrintingParams: Resource to define RIP parameters. Currently allows to pass the Caldera PrintConfig data. In the future we could support some of JDF attributes (Collate...)

DigitalPrintingParams/@Class	"Parameter"
DigitalPrintingParams/@ID	Unique ID.
DigitalPrintingParams/@Status	"Available"
DigitalPrintingParams/cal:PrintConfig	Caldera specific PrintConfig node.

Device: Define the device to use (typically a printer). If the device is not specified, it must be defined otherwise in the Rip (as workflow parameter...)

Device/@Class	"Implementation"
Device/@ID	Resource ID
Device/@DeviceID	Device name. This is the server name in Caldera. The server name can contain a hostname if the server is configured as remote on the Bridge's computer (new in V10). In that case the format is "hostname::printer name". MIS can retrieve list of devices with KnownDevices JMF query (new in V10). Starting from V10.1 the DeviceID can contain a name of a print configuration. In that case the format is "hostname::printer:configuration".
Device/@Status	"Available"

Component: The component resource is an output resource, representing the product of the job (printed images).

Component/@Class	"Quantity"
Component/@ID	Resource ID
Component/@Status	"Unavailable" ?
Component/@ComponentType	"FinalProduct"

CustomerInfo: Information about customers. This resource is more or less just a specific comment. The values do not interfere with job execution. The information can be visualised in Remote Browser. A job should have a unique CustomerInfo.

CustomerInfo/@ID	Unique ID
CustomerInfo/@Class	"Parameter"
CustomerInfo/@Status	"Available"
CustomerInfo/@CustomerID	Product-specific customer ID. This value can be printed in Caldera annotations and/or crop marks by inserting "@(CustomerID)" in the text.
CustomerInfo/@CustomerOrderID	Product-specific customer order ID. This value can be printed in Caldera annotations and/or crop marks by inserting "@(CustomerOrderID)" in the text.
CustomerInfo/Contact	Nodes describing the customer info. See below. Multiple nodes possible.

Contact:

Contact/@ContactTypes	String specifying contact type. Valid values: "Accounting", "Administrator", "Agency", "Approver", "ArtReturn", "Author", "Billing", "Customer", "Delivery", "DeliveryCharge", "Designer", "Editor", "Illustrator", "Owner", "Photographer", "Pickup", "Sender", "Supplier", "SurplusReturn", "TelephoneSanitizer". Whatever is in here is just displayed in RemoteBrowser. Can have multiple values (space separated). To have this displayed in JobTicket in Caldera, use "Customer" and "Sales".
------------------------------	---

Contact/Address	only one node possible
Contact/Company	only one node possible
Contact/Person	only one node possible
Contact/ComChannel	multiple nodes possible

Address:

Address/@City	
Address/@Country	
Address/@CountryCode	
Address/@PostalCode	
Address/@Street	
Address/@Region	
Address/@ExtendedAddress	

Company:

Company/@OrganizationName	Name of the company
----------------------------------	---------------------

Person:

Person/@FirstName	
Person/@FamilyName	
Person/@JobTitle	
Person/@Languages	
Person/@NamePrefix	
Person/@NameSuffix	
Person/Address	only one node possible
Person/ComChannel	multiple nodes possible

ComChannel:

ComChannel/@ChannelType	"Phone", "Email", "Fax", "WWW", "JMF", "PrivateDirectory", "InstantMessaging"
ComChannel/@Locator	Locator of this type of channel. URL syntax is recommended (mailto:..).

ResourceLinkPool

RunListLink: Link to a RunList resource

RunListLink/@Usage	Usage of the RunLink. <ul style="list-style-type: none"> • "Input": resource is an input • "Output": resource is an output. Currently ignored (Could be used later to output ripped files for archival).
RunListLink/@rRef	Reference to the ID of the RunLink.

LayoutLink: Link to a Layout resource (**Added in Caldera 10.1**)

LayoutLink/@Usage	Usage of the resource. <ul style="list-style-type: none"> • "Input": resource is an input • "Output": resource is an output. Currently ignored (Could be used later to output automatic layouts).
LayoutLink/@rRef	Reference to the ID of the Layout resource.

DigitalPrintingParamsLink: Link to DigitalPrintingParams resource.

DigitalPrintingParamsLink/@Usage	"Input"
DigitalPrintingParamsLink/@rRef	Reference to a DigitalPrintingParams

DeviceLink: Link to a Device resource.

DeviceLink/@Usage	"Input" (despite the fact a printer is an output device, this printer definition is the input of the process)
DeviceLink/@rRef	Reference to a Device resource ID.

ComponentLink: Link to Component resource. It can be used to specify the amount to print, considering that all the pages and copies specified by RunList represent an amount of “1”.

ComponentLink/@Usage	“Output”
ComponentLink/@rRef	ID of linked Component resource.
ComponentLink/@Amount	Expected amount (number of copies) of the document to print. “Document” is a set of input files with the list of pages/copies described by a RunList resource. (new in V10)
ComponentLink/@ActualAmount	Total number of document copies that was actually printed. (new in V10)

CustomerInfoLink: Link to a CustomerInfo resource.

CustomerInfoLink/@Usage	“Input”
CustomerInfoLink/@rRef	ID of linked CustomerInfo resource.

Example: ProcessGroup resource linking using common resource (2 separate print jobs)

```
<JDF ID="mainJDF01" Type="ProcessGroup"...>
  <ResourcePool>
    <DigitalPrintingParams ID="DPP221".../>
  </ResourcePool>
  <JDF ID="J1" Type="DigitalPrinting"...>
    <ResourcePool>
      <RunList ID="RL01".../>
    </ResourcePool>
    <ResourceLinkPool>
      <DigitalPrintingParamsLink Usage="Input" rRef="DPP221"/>
      <RunListLink Usage="Input" rRef="RL01"/>
    </ResourceLinkPool>
  </JDF>
  <JDF ID="J2" Type="DigitalPrinting"...>
    <ResourcePool>
      <RunList ID="RL02".../>
    </ResourcePool>
    <ResourceLinkPool>
      <DigitalPrintingParamsLink Usage="Input" rRef="DPP221"/>
      <RunListLink Usage="Input" rRef="RL02"/>
    </ResourceLinkPool>
  </JDF>
</JDF>
```

```

        </ResourceLinkPool>
    </JDF>
</JDF>

```

AuditPool

Common attributes to all AuditPool elements:

@AgentName	Optional. Product name.
@AgentVersion	Optional. Product version.
@TimeStamp	Time of event or element creation. Time format is ISO8601:2004 "YYYY-MM-DDTHH:MM:SSZ". The "Z" can be replaced by time zone offset (+01:00, -02:00 etc...). In that case the time is local. If Z is used, the time is UTC.

Created: This node should be written when the JDF is created. The node has no additional attributes.

ResourceAudit: Information about printing and errors for individual input files. This is a link between JDF and a Caldera job. The behavior of the node is specific to Caldera if <cal:Status> is present. There is always only one occurrence of ResourceAudit with unique @FileSpecURL, server_name and job_id values. If the cal:Status node does not contain job_id node, the status (probably an error) is relative to an event that occurred before printing (opening the file, ...)

ResourceAudit/RunListLink	The link to a RunList resource (see the ResourceLinkPool for details)
ResourceAudit/RunListLink/cal:Status/*	Caldera-specific information with JobExport Info data. The format is described in separate specification (Caldera JobExport Info Specification)
ResourceAudit/RunListLink/cal:Status/@FileSpecURL	Complete URL of the file to which this node is relative.

ProcessRun: Information about running the process. This element logs the end of a process run. Any subsequent elements belong to the next run.

ProcessRun/@End	Date and time at which the process ended.
------------------------	---

ProcessRun/@EndStatus	Status of the process at the end of run. <ul style="list-style-type: none"> • Aborted • Completed
ProcessRun/@Start	Date and time at which the process started.

Modified: Information about a modification of the JDF job. Should be created each time a node somewhere in the ResourceLink is modified. This is inserted for example when the Bridge creates links for thumbnails into FileSpec/cal:Thumb node.

Modified/@XPath	XPath of modified node in the JDF data.
Modified/Comment	Human readable log information about the modification.

AuditPool example:

```
<AuditPool>
  <Created AgentName="Flow+" AgentVersion="4.0"
    TimeStamp="2013-04-17T08:48:16Z"/>
  <ResourceAudit AgentName="Caldera" AgentVersion="9.20"
    TimeStamp="2013-04-17T10:24:11Z">
    <RunListLink Usage="Input" rRef="RL001">
      <cal:Status FileSpecURL="http://myip/myfile.pdf">
        <job_state>Error</job_state>
        <job_error>Unknown file format</job_error>
      </cal:Status>
    </RunListLink>
  </ResourceAudit>
  <ProcessRun AgentName="Caldera" AgentVersion="9.20"
    End="2013-04-17T10:24:11Z" Start="2013-04-17T09:31:41Z"
    EndStatus="Aborted" TimeStamp="2013-04-17T10:24:11Z">
  </ProcessRun>
</AuditPool>
```

Caldera extension to JDF

The Bridge completes the JDF data with the following information at various moments:

- File thumbnail (created at the moment a file is loaded in Caldera)
- Information about printed jobs in Caldera
- Information about the total number of copies printed of a resource.

cal:Thumb: This node is created in a FileSpec node of a resource as soon as the file has been loaded in Caldera. In case of a multi-page file there is one thumbnail by page. A thumbnail is kept during the lifetime of a job in the Bridge, it is destroyed when the job is deleted.

cal:Thumb/@URL	URL of the thumbnail using the file:// scheme. Useable locally on the Bridge computer.
cal:Thumb/@RemoteURL	URL of the thumbnail using the http:// scheme, thus accessible by remote systems.
ca:Thumb/@page	If the resource is multipage, this designates the page represented by the thumbnail (1st page = 0)

cal:Info: This unique node is created in a FileSpec node of a resource at two possible moments:

- When a multi-page resource is opened
- When number of printed occurrences of the resource increases

In case of multi-page resources, each single page is counted separately.

cal:Info/@Pages	Number of pages in the resource file. This is written only if the resource has at least 2 pages. Omitted else.
cal:Info/PrintedCopies	Total number of printed copies of the resource.
cal:Info/PrintedCopies/@page	If multi-page resource, this specifies to which page the count is related to (0 = first page). Omitted if single-page document.

Example:

```
<RunList Class="Parameter" ID="RL0001" Status="Available">
  <LayoutElement>
    <FileSpec URL="file:///input/1234.pdf">
      <cal:Thumb page="0" URL="file:///thumb/HwGVq3qBQCUR.jpg"
        RemoteURL="http://192.168.1.1/thumb.php?HwGVq3qBQCUR.jpg"/>
      <cal:Thumb page="1" URL="file:///thumb/9o7c6oLxnMGT.jpg"
        RemoteURL="http://192.168.1.1/thumb.php?9o7c6oLxnMGT.jpg"/>
      <cal:Info Pages="2">
        <PrintedCopies page="0">1</PrintedCopies>
        <PrintedCopies page="1">1</PrintedCopies>
      </cal:Info>
    </FileSpec>
  </LayoutElement>
</RunList>
```

cal:PrintConfig: This node is located in DigitalPrintingParams node of ResourcePool. Its contents describes precisely print parameters in Caldera specific format (described in a separate specification: PrintConfigXML.pdf).

The contents of cal:PrintConfig is identical to an exported print configuration from a print module in Caldera. The recommended way to populate the cal:PrintConfig node is as follows:

- Open the Caldera print module of the printer that will be used to print the job.
- Set all the parameters as needed.
- Save the configuration (floppy disk icon).
- Open the Configuration management window (button “...” to the right of the floppy disk icon)
- Select newly created configuration and export it (button Export).
- That will create a XML file which contents can be used to fill the cal:PrintConfig node.



Starting V10.0 build 150318 it is now possible to specify directly a name of a configuration in the JDF Device resource (as part of the DeviceID) instead of embedding the cal:PrintConfig node (DeviceID="Printer name:my configuration name").

Starting V11.1, if cal:PrintConfig is included in JDF ticket at the same time a print preset is selected with DeviceID, the preset is then selected and data from PrintConfig is merged with it, so customers are allowed to modify some aspects of the print configurations based on a preset. Few constraints exist in that case:

- The size of the image can be only specified by the scale, using PrintConfig/img/scale_x_phys and scale_y_phys parameters. The size of the image can not be set using PrintConfig/img/w and PrintConfig/img/h values, because the priority is always accorded to the scale.
- When changing the image scale, PrintConfig/img/scale100_phys and PrintConfig/img/full parameters should be both set to false because of parameter priorities.
- When specifying different scales in X and Y directions, the config should specify PrintConfig/img/keep_ratio set to "false" to allow that.
- Any print parameters should be specified using the values and never "idx" attributes. Values of the idx attributes are used only as fallback when the value is not recognized, and the indexes could vary between different Caldera versions.

Before V11.1, when both cal:PrintConfig and DeviceID with preset name were present, it was always cal:PrintConfig which has been used and the preset name was ignored.

cal:Status: This is a print log from Caldera, located in an AuditPool/ResourceAudit node. One cal:Status node is equivalent to a Caldera print job. The contents is specified in a separate

specification (JobExport_Info_Spec.pdf). Keep in mind that there is exactly one cal:Status node per resource file and specific Caldera job id. That is, if an existing Caldera job is printed again, the previous information is replaced by an updated one.

If a Caldera job contains more than one resource files (from the same or different JDF jobs), cal:Status node corresponding to that job is repeated for each of the resources.

cal:Status/@FileSpecURL	Complete URL of the file resource.
cal:Status/@FID	Internal Bridge file identifier. 1-based counter of files in the job. Used internally by the Bridge to identify a file in unique way.
cal:Status/contents	Contents of the Caldera print job. See below.
cal:Status/thumbnail	Http-based URL to a thumbnail of the Caldera job.
cal:Status/PrintConfig	Node with actual print configuration of the print job. Present only if the feature has been enabled in Nexio config (V11.1)
cal:Status/*	Other nodes: Caldera-specific information about the print job. See JobExport_Info_Spec.pdf.

The cal:Status/contents node in cal:Status which is specific to the Bridge (thus not described in the JobExport_Info_Spec.pdf specification). It allows to understand how the job resource has been used. It allows to tell if the resource was tiled and/or composed/nested.

contents/.../item	The “item” node represents the resource.
contents/.../item/@page	If the item is multi-page, the @page attribute specifies which page has been printed. First page = 0.
contents/.../item/@npages	If the item is multi-page, the @npages attribute specifies the total number of pages.
contents/.../other	The “other” node signifies that there are other resources (from the same job or a different one) in the given container (compose). That acts as indication that the job media/ink consumption are not an exact amount for the given resource.
contents/.../compose	This is a container for Compose or nesting
contents/.../tiling	This is a tiling container.
contents/.../tiling/@tile	The tile number represented by a tiling container. First tile = 0.

contents/.../tiling/@ntiles	Total number of tiles.
contents/.../tiling/@id	The ID of a tiling. Allows to group tiles belonging to one poster.

Example: Demonstrates a 2 page document that has been composed, then tiled and printed together using the nesting, together with other files.

```

<contents>
  <compose>
    <tiling tile="0" ntiles="2" id="309149B7-A850-448B-8807-9970659208C3">
      <compose>
        <item page="0" npages="2"/>
        <item page="1" npages="2"/>
      </compose>
    </tiling>
    <tiling tile="1" ntiles="2" id="309149B7-A850-448B-8807-9970659208C3">
      <compose>
        <item page="0" npages="2"/>
        <item page="1" npages="2"/>
      </compose>
    </tiling>
    <other/>
  </compose>
</contents>

```

JMF Messages

JMF/@SenderID	ID of the sender of the message
JMF/@Version	JDF/JMF Version. "1.4"
JMF/@TimeStamp	Date/time when the message was created
JMF/@xmlns	"http://www.CIP4.org/JDFSchemas_1_1"
JMF/@ICSVersions	"JMF_L1-1.4"
JMF/@ResponseURL	Response is sent here when received over unidirectional channel (hotfolder). Shouldn't be used in bidirectional channel (HTTP). Must be generated by the sending agent explicitly with explicit filename and starting with file://. If the path-part (after file://) does not start with /, the path is considered as relative to the hotfolder. The file must not be directly in the hotfolder and all the directories must be created in advance. (Subdirectories in the hotfolder are accepted)

JMF on bidirectional channel (HTTP):

```
<?xml version="1.0" encoding="UTF-8"?>
<JMF xmlns="http://www.CIP4.org/JDFSchemas\_1\_1" SenderID="FlowPlus320334"
    Timestamp="2013-04-17T10:24:11Z" Version="1.4">
    ...
</JMF>
```

JMF message over unidirectional channel (hotfolder):

```
<?xml version="1.0" encoding="UTF-8"?>
<JMF xmlns="http://www.CIP4.org/JDFSchemas\_1\_1" SenderID="FlowPlus320334"
    Timestamp="2013-04-17T10:24:11Z" Version="1.4"
    ResponseURL="file:///some/path/temp100392092.jmf">
    ...
</JMF>
```

Commands

Command is a message that can affect the state of the controller. The Response is generated immediately after receiving the Command.

Command/@ID	Message ID. Used as refID in Response etc...
Command/@Type	Command type.

SubmitQueueEntry Command. The submitter must retrieve the new QueueEntryID from the response (Response/QueueEntry/@QueueEntryID). Response/QueueEntry/@JobID is copied from JDF/@JobID.

Command/QueueSubmissionParams/@URL	URL of the JDF file. "file://", "http://" or "https://"
Command/QueueSubmissionParams/@ReturnURL	URL where to send the JDF file after the job is completed or aborted. The URL must support directly JDF input (no JMF). Must be explicit URL if file://
Command/QueueSubmissionParams/@ReturnJMF	URL where to send ReturnQueueEntry command after the job is completed or aborted. The URL must accept a MIME package with both JMF command and JDF ticket referenced by "cid" URL scheme. See "Job submission scenarios" later in the doc. (Added in Caldera 11.1)
Command/QueueSubmissionPar	If the job submission is triggered by a

ams/@refID	RequestQueueEntry command, this attribute is REQUIRED and must have the same value as Command/@ID that triggered the job submission.
Command/QueueSubmissionParams/@Hold	<p>“true”: The job is inserted in “Held” status (even if the Bridge is configured to not hold jobs)</p> <p>“false” (default): The job is inserted in “Waiting” status (if the Bridge is configured to hold jobs automatically, the job is inserted in “Held” status even if Hold=“false”)</p>
Command/QueueSubmissionParams/@GangName	<p>Sets the gang name to which the job belongs. This allows to sort jobs in image bar and to nest them together. Specified value should not contain any space. (New in Caldera 10.1)</p> <p>Note: Please note that using the JMF SubmitQueueEntry command is the only way to assign a job to a gang. It is not possible when the JDF ticket is submitted directly.</p>
Command/QueueSubmissionParams/@GangPolicy	<p>Specifies the ganging policy (New in Caldera 10.1):</p> <ul style="list-style-type: none"> • “Gang”: Add job to the gang specified by GangName. This is default if a GangName is specified. • “NoGang”: Ignore the GangName, no ganging is performed. This is the default when no GangName is specified. • “GangAndForce”: Currently identical to “Gang”.

RequestQueueEntry Command. This command is sent in the opposite direction than usual messages (Device=>MIS). It requests a new job(s) to be submitted. This is currently used to request job submission from systems that are not able to do that spontaneously (WebShop).

Command/RequestQueueEntryParams/@QueueURL	Controller URL to which JMF messages should be sent. This actually allows to not configure the Controller URL in advance, and use the one supplied here dynamically.
--	--

Response: Message sent after receiving a query or a command. In a bidirectional channel (HTTP), the response is sent as the body of a http answer to a http request. In unidirectional channel (hotfolder) it is sent according to the JMF/@ResponseURL value.

Response/@refID	ID of the message this response concerns. If not used, the response is for the whole JMF message (parse error...)
Response/@Type	The same value as Command/@Type or Query/@Type.
Response/@ReturnCode	<ul style="list-style-type: none"> • 0: success (default, optional) • 1: general error

	<ul style="list-style-type: none"> • 2: internal error • 3: XML parse error • 5: query/command not implemented • 6: invalid params • 7: insufficient params • 105: unknown queue entry • 106: request failed because entry is running • 113: request failed because entry in required state • 114: request failed, job is in final state (PendingReturn, Completed or Aborted) • 120: URL cannot be resolved
--	--

The response can contain a Notification node in the case of error (for ex. parse error, etc...). The Notification node has the same attributes as an AuditPool item (@TimeStamp, @AgentName, @AgentVersion) and the following:

Response/Notification/@Class	"Error" or "Warning"
Response/Notification/Comment	Error message.

Response to a successful "SubmitQueueEntry" command: The Response node must contain a QueueEntry node on successful operation. The node can contain Notification in case of error. The response does NOT confirm job execution, only that the job has been queued.

Response/QueueEntry/@JobID	The same value as JDF/@JobID, for reference.
Response/QueueEntry/@QueueEntryID	ID generated by the Bridge, representing the newly created queue entry.
Response/QueueEntry/@Status	Initial status of the job. This depends on the bridge configuration and QueueSubmissionParams/@Hold parameter. <ul style="list-style-type: none"> • "Waiting": job is added active • "Held": job is hold, must be activated manually.
Response/QueueEntry/@GangName	If the job was submitted belonging to a gang, the gang name is displayed here (New in Caldera 10.1)

RemoveQueueEntry Command: Remove an entry from the queue. This command requires the entry to be in the following status:

- Waiting
- Held
- Completed

- Aborted

Any other queue entry status causes an error.

Command/RemoveQueueEntryParams/ QueueFilter	Filter to define a set of queue entries to remove. For details see QueueFilter of QueueStatus query.
--	--

Example:

```
<JMF xmlns="http://www.CIP4.org/JDFSSchema_1_1" SenderID="FlowPlus01"
      TimeStamp="2013-04-17T10:24:11Z" Version="1.4">
  <Command ID="C001" Type="RemoveQueueEntry">
    <RemoveQueueEntryParams>
      <QueueFilter>
        <QueueEntryDef QueueEntryID="435"/>
      </QueueFilter>
    </RemoveQueueEntryParams>
  </Command>
</JMF>
```

RemoveQueueEntry Response: The response has no additional attributes. Error or warning can be appended if the command failed (even partially).

AbortQueueEntry Command: Abort a running entry. This command requires the entry to be in the following status:

- Waiting
- Held
- InProgress
- Suspended (not used in Caldera)

Any other queue entry status causes an error.

Command/AbortQueueEntryParams/ @EndStatus	Status to which the entry will transit. Can be either "Aborted" (default) or "Completed"
Command/AbortQueueEntryParams/ QueueFilter	Filter to define a set of queue entries to abort. For details see QueueFilter of QueueStatus query.

Example:

```
<JMF xmlns="http://www.CIP4.org/JDFSSchema_1_1" SenderID="FlowPlus01"
      TimeStamp="2013-04-17T10:24:11Z" Version="1.4">
  <Command ID="C001" Type="AbortQueueEntry">
    <AbortQueueEntryParams EndStatus="Aborted">
      <QueueFilter>
        <QueueEntryDef QueueEntryID="435"/>
      </QueueFilter>
    </AbortQueueEntryParams>
  </Command>
</JMF>
```

```

        </QueueFilter>
    </RemoveQueueEntryParams>
</Command>
</JMF>

```

AbortQueueEntry Response: The response has no additional attributes. Error or warning can be appended if the command failed (even partially).

Queries

Query/@ID	Message ID. Used as refID in Response etc...
Query/@Type	Query type.

KnownMessages Query: ask for the list of supported JMF messages. This message has no additional parameters. Added in Caldera V10.

Example:

```

<JMF xmlns="http://www.CIP4.org/JDFSSchema_1_1" SenderID="FlowPlus01"
    TimeStamp="2013-04-17T10:24:11Z" Version="1.4">
    <Query ID="C001" Type="KnownMessages" />
</JMF>

```

KnownMessages Response: The response has one MessageService entry per supported message.

Response/MessageService/@Type	Type of the message
Response/MessageService/@JMFRole	<ul style="list-style-type: none"> “Receiver”: message is understood by the Bridge “Sender”: message is generated by the Bridge
Response/MessageService/@Query	If “true”, the message is a query (“false” is the default)
Response/MessageService/@Command	If “true”, the message is a command (“false” is the default)

Example:

```

<JMF xmlns="http://www.CIP4.org/JDFSSchema_1_1" SenderID="CalderaBridge"
    Version="1.4" TimeStamp="2014-01-16T16:23:41+01:00">
    <Response ID="R001" Type="KnownMessages" refID="C001">
        <MessageService JMFRole="Receiver" Command="true"
            Type="SubmitQueueEntry"/>
    </Response>
</JMF>

```

```

    <MessageService JMFRole="Sender" Command="true" Type="RequestQueueEntry"/>
    <MessageService JMFRole="Receiver" Query="true" Type="KnownMessages"/>
  </Response>
</JMF>

```

KnownDevices Query: ask for the list of devices known by Bridge.

Query/DeviceFilter/@DeviceDetails	<ul style="list-style-type: none"> • “None” (default): Returns only @DeviceID and @DeviceStatus. • “Brief”: same as “None” • “Details”: Insert Device nodes, without capabilities • “Capability”: Insert DeviceCap nodes with detailed device capabilities.
Query/DeviceFilter/Device/@DeviceID	Return only information about devices that matches the DeviceID. Multiple Device nodes possible.
Query/DeviceFilter/Device/@ModelName	Return only information about devices that matches the model name. Multiple Device nodes possible.

Example:

```

<JMF xmlns="http://www.CIP4.org/JDFSSchema_1_1" SenderID="FlowPlus01"
    TimeStamp="2013-04-17T10:24:11Z" Version="1.4">
  <Query ID="C001" Type="KnownDevices">
    <DeviceFilter DeviceDetails="Details"/>
  </Query>
</JMF>

```

KnownDevices Response: The response has one DeviceList entry and as many DeviceInfo entries as known devices.

Response/DeviceList/DeviceInfo/@DeviceStatus	Status of the device. Currently always “Running”
Response/DeviceList/DeviceInfo/Device/@DeviceID	Device ID to use as @DeviceID in Device resource. This corresponds to the server name in Caldera and can eventually contain a host name or IP address if the server is handled remotely (“host::server name”)
Response/DeviceList/DeviceInfo/Device/@ModelName	The model name. This correspond to Caldera model name.
Response/DeviceList/DeviceInfo/Device/DeviceCap	When @DeviceDetails=“Capability”, this node is returned. It contains detailed information about the

	device, such as list of profiles, list of loadings and list of extra inks. The format is not detailed here, please request when needed.
--	---

Example:

```
<JMF xmlns="http://www.CIP4.org/JDFSSchema_1_1" SenderID="CalderaBridge"
      Version="1.4" TimeStamp="2014-01-16T16:23:41+01:00">
  <Response ID="R001" Type="KnownDevices" refID="C001">
    <DeviceList>
      <DeviceInfo DeviceStatus="Running">
        <Device DeviceID="Designjet-L26500-A" ModelName="Designjet-L26500"/>
      </DeviceInfo>
      <DeviceInfo DeviceStatus="Running">
        <Device DeviceID="Designjet-L26500-B" ModelName="Designjet-L26500"/>
      </DeviceInfo>
      <DeviceInfo DeviceStatus="Running">
        <Device DeviceID="192.168.0.123::Lambda" ModelName="Lambda"/>
      </DeviceInfo>
    </DeviceList>
  </Response>
</JMF>
```

SubmissionMethods Query: ask for supported methods to submit jobs. This message has no additional parameters. Added in Caldera V10.

Example:

```
<JMF xmlns="http://www.CIP4.org/JDFSSchema_1_1" SenderID="FlowPlus01"
      TimeStamp="2013-04-17T10:24:11Z" Version="1.4">
  <Query ID="C001" Type="SubmissionMethods" />
</JMF>
```

SubmissionMethods Response: The response has one SubmissionMethods entry.

Response/SubmissionMethods/@Packaging	Supported packaging. Currently "MIME" (Bridge can accept MIME multipart/related content)
Response/SubmissionMethods/@URLSchemes	List of supported URL schemes. Currently this is "file http" which means Caldera can understand URLs starting with file:// and http://

Example:

```
<JMF xmlns="http://www.CIP4.org/JDFSSchema_1_1" SenderID="CalderaBridge"
      Version="1.4" TimeStamp="2014-01-16T16:23:41+01:00">
  <Response ID="R001" Type="SubmissionMethods" refID="C001">
    <SubmissionMethods Packaging="MIME" URLSchemes="file http"/>
  </Response>
```

</JMF>

Status: query information about jobs

Query/StatusQuParams/@JobID	Return information only relative to jobs with given JobID
Query/StatusQuParams/@JobPartID	Return information only relative to jobs with given JobPartID
Query/StatusQuParams/@QueueEntryID	Return information only relative to job with given queue entry ID.
Query/StatusQuParams/@DeviceDetails	<ul style="list-style-type: none">• “None” (default): Returns only DeviceInfo node. Does not return jobs.• “Brief”: Provide job details (JobPhase nodes)• other: ignored values, as “Brief”
Query/StatusQuParams/@JobDetails	Used only if @DeviceDetails is not “None”. <ul style="list-style-type: none">• “None” (default): return basic job information (@JobID, @JobPartID, @Amount and @PercentCompleted)• “Brief”: Provide all available job information• “Full”: Like “Brief” with an URL to retrieve the full JDF contents.• other: ignored, as “Brief”.

Example:

```
<JMF xmlns="http://www.CIP4.org/JDFSSchema_1_1" SenderID="FlowPlus01"
      Timestamp="2013-04-17T10:24:11Z" Version="1.4">
  <Query ID="C001" Type="Status">
    <StatusQuParams DeviceDetails="Brief" JobDetails="Brief"/>
  </Query>
</JMF>
```

Status response: List jobs, sorted by devices

Response/DeviceInfo/@DeviceID	Device ID handling the jobs. Response can have multiple DeviceInfo nodes, one per device.
Response/DeviceInfo/@DeviceStatus	“Running”. Currently no other value is returned.

Response/DeviceInfo/JobPhase/@JobID	
Response/DeviceInfo/JobPhase/@JobPartID	
Response/DeviceInfo/JobPhase/@QueueEntryID	
Response/DeviceInfo/JobPhase/@Amount	Number of currently printed documents as defined by JDF resources.
Response/DeviceInfo/JobPhase/@PercentCompleted	"100" means the job is completed (requested amount has been achieved)
Response/DeviceInfo/JobPhase/@Status	Status of the JDF job. The same as JDF/@Status.
Response/DeviceInfo/JobPhase/@Activation	Activation of the job. The same as JDF/@Activation
Response/DeviceInfo/JobPhase/@URL	URL to full JDF tree of the job. Returned JDF has @Activation = "Informative". The JDF represent a snapshot of the job at the moment the URL is called.

Example:

```
<JMF xmlns="http://www.CIP4.org/JDFSchema_1_1" SenderID="CalderaBridge"
  Version="1.4" TimeStamp="2014-07-23T12:12:17+02:00">
  <Response ID="R001" Type="Status" refID="C001">
    <DeviceList>
      <DeviceInfo DeviceStatus="Running" DeviceID="CalderaJet">
        <JobPhase JobID="small" JobPartID="1" Amount="1"
          PercentCompleted="100" Status="Completed" Activation="Active"
          QueueEntryID="425"/>
        <JobPhase JobID="small" JobPartID="3" PercentCompleted="0"
          Status="Waiting" Activation="Held" QueueEntryID="428"/>
      </DeviceInfo>
    </DeviceList>
  </Response>
</JMF>
```

QueueStatus: query information about device queues and jobs.

Query/QueueFilter	optional, to specify device/job filters
Query/QueueFilter/@QueueEntryDetails	Specifies detail of requested information: <ul style="list-style-type: none"> • "None": Do not return QueueEntry elements. Only Queue element is filled.

	<ul style="list-style-type: none"> • “Brief” (default): Return Queue and QueueEntry elements • “JobPhase”: Insert JobPhase elements in QueueEntry nodes (see “Status” JMF query).
Query/QueueFilter/@JobID	List only jobs with specified JobID
Query/QueueFilter/@JobPartID	List only jobs with specified JobPartID
Query/QueueFilter/@OlderThan	List only jobs with submission time older or equal to specified timestamp
Query/QueueFilter/@NewerThan	List only jobs with submission time newer or equal to specified timestamp
Query/QueueFilter/@GangNames	Space-separated list of gang names. List only jobs belonging to specified gangs. (New in Caldera 10.1)
Query/QueueFilter/Device	Allow to filter on specific device (optional). Multiple Device nodes are allowed.
Query/QueueFilter/Device/@DeviceID	Show only jobs from specified device’s queue
Query/QueueFilter/Device/@ModelName	Show only jobs from queues of devices with specified model name.
Query/QueueFilter/QueueEntryDef	Allow to filter on specific queue entry (optional). Note: Only one QueueEntryDef node is allowed currently.
Query/QueueFilter/QueueEntryDef/@QueueEntryID	Queue entry ID to show information about.

Example:

```

<JMF xmlns="http://www.CIP4.org/JDFSschema_1_1" SenderID="FlowPlus01"
      TimeStamp="2013-04-17T10:24:11Z" Version="1.4">
  <Query ID="C001" Type="QueueStatus">
    <QueueFilter QueueEntryDetails="Brief">
      <Device DeviceID="my_printer"/>
    </QueueFilter>
  </Query>
</JMF>

```

QueueStatus response: Lists current queue entries, in order of processing priorities, classified by device.

Response/Queue/@DeviceID	Device ID. Multiple Queue nodes possible, one by device.
Response/Queue/@Status	"Running". Currently no other status is returned.
Response/Queue/QueueEntry/@JobID	JobID of the job
Response/Queue/QueueEntry/@JobPartID	JobPartID of the job
Response/Queue/QueueEntry/@QueueEntryID	Queue entry ID
Response/Queue/QueueEntry/@Status	Queue entry status. Warning, this is not the same as JDF/@Status. Possible values: "Held", "Waiting", "Running", "Suspended", "PendingReturn", "Completed", "Aborted", "Removed"
Response/Queue/QueueEntry/@SubmissionTime	Timestamp of job submission
Response/Queue/QueueEntry/@GangName	If the job was submitted belonging to a gang, the gang name is displayed here (New in Caldera 10.1)
Response/Queue/QueueEntry/JobPhase	See "Status" JMF query for details.

Example:

```
<JMF xmlns="http://www.CIP4.org/JDFSSchema_1_1" SenderID="CalderaBridge"
      Version="1.4" TimeStamp="2014-07-22T15:55:42+02:00">
  <Response ID="RE001" Type="QueueStatus" refID="C001">
    <Queue DeviceID="CalderaJet" Status="Running">
      <QueueEntry JobID="MyJob1" JobPartID="1" QueueEntryID="425"
Status="Completed" SubmissionTime="2014-07-22T14:51:50+02:00"/>
      <QueueEntry JobID="Job2" JobPartID="1" QueueEntryID="426"
Status="Completed" SubmissionTime="2014-07-22T14:55:07+02:00"/>
    </Queue>
  </Response>
</JMF>
```

Persistent channels & Signals

Status and QueueStatus queries can be used as persistent channels that can be subscribed. The Bridge will then issue a "Signal" message when something new happened, as defined by the subscription. A signal is identical to a response, except that the "Response" node is replaced by "Signal". A persistent channel is identified by @ID of the query used to subscribe. A signal is then issued with @refID identical to that ID.

Subscription is performed by appending a “Subscription” node to the Query element, as defined in the following table:

Query/Subscription/@URL	The URL where the signals are to be sent.
--------------------------------	---

When a query carries the Subscription node, the query is not performed immediately. Instead, when an event matches the filter defined by the query (Status or QueueStatus), the Signal is emitted to given URL, acting as if the query was requested at that moment.

When a subscription is done, the caller should check if it was successful by checking Response/@Subscribed attribute, which is set to “true” if it worked. In the opposite case an error will be inserted as Notification.

Currently the Bridge supports only less-reliable persistent channels (type “FireAndForget”), which means the signal reception is not acknowledged and the signal could be lost if the receiver is not ready to receive it.

A subscriber should remove the subscription (using StopPersistentChannel query) channel when done, because the channel is never closed automatically.

Example: In this example the caller subscribes to QueueStatus message for printer “my_printer”. The persistent channel ID is “C001” and it has to be kept and used to stop the channel.

```
<JMF xmlns="http://www.CIP4.org/JDFSSchema_1_1" SenderID="FlowPlus01"
      TimeStamp="2013-04-17T10:24:11Z" Version="1.4">
  <Query ID="C001" Type="QueueStatus">
    <QueueFilter QueueEntryDetails="Brief">
      <Device DeviceID="my_printer"/>
    </QueueFilter>
    <Subscription URL="http://myhost/path"/>
  </Query>
</JMF>

<JMF xmlns="http://www.CIP4.org/JDFSSchema_1_1" SenderID="CalderaBridge"
      Version="1.4" TimeStamp="2014-07-22T15:55:42+02:00">
  <Response ID="RE001" Type="QueueStatus" refID="C001" Subscribed="true"/>
</JMF>
```

Later, when a queue entry changes its status, the caller will get a signal like this. The signal can be identified by its @refID which is the channel ID.:

```
<JMF xmlns="http://www.CIP4.org/JDFSSchema_1_1" SenderID="CalderaBridge"
      Version="1.4" TimeStamp="2014-07-22T15:55:42+02:00">
  <Signal ID="SG001" Type="QueueStatus" refID="C001">
    <Queue DeviceID="my_printer" Status="Running">
```

```

        <QueueEntry    JobID="MyJob1"    JobPartID="1"    QueueEntryID="425"
Status="Completed" SubmissionTime="2014-07-22T14:51:50+02:00"/>
    </Queue>
</Response>
</JMF>

```

KnownSubscriptions Query: Lists all known subscriptions (persistent channels)

Query/SubscriptionFilter/	Optional filter. If omitted, all channels are listed.
Query/SubscriptionFilter/@ChannelID	List only specified channel
Query/SubscriptionFilter/@DeviceID	List only channels specific to specified device
Query/SubscriptionFilter/@JobID	List channels specific to specified JobID
Query/SubscriptionFilter/@JobPartID	List channels specific to specified JobPartID
Query/SubscriptionFilter/ @MessageTypes	List channels with specific types. This is a space-separated list with one or more of following values: "Status", "QueueStatus".
Query/SubscriptionFilter/@URL	List channels subscribed with specified URL.

KnownSubscription reponse: list of persistent channels.

Response/SubscriptionInfo	One node per persistent channel
Response/SubscriptionInfo/@ChannelID	Persistent channel ID
Response/SubscriptionInfo/@Family	"Signal"
Response/SubscriptionInfo/@SenderID	Sender ID used to emit the signal
Response/SubscriptionInfo/ @MessageType	"Status" or "QueueStatus"
Response/SubscriptionInfo/Subscription	Copy of the Subscription node used to subscribe to the channel.

StopPersistentChannel Command: Delete a persistent channel.

Command/StopPersChParams/@ChannelID	The ID of the channel to be stopped.
Command/StopPersChParams/@JobID	Stop all channels relative to specific JobID

Command/StopPersChParams/@JobPartID	Stop all channels relative to specific JobPartID
Command/StopPersChParams/@MessageType	Stop all channels of this type ("Status" or "QueueStatus")
Command/StopPersChParams/@QueueEntryID	Stop all channels relative to specified queue entry.
Command/StopPersChParams/@URL	Stop all channels that have been subscribed with this URL.

JMF Examples

New job command and response:

```

<JMF xmlns="http://www.CIP4.org/JDFSSchema\_1\_1" SenderID="FlowPlus01"
    TimeStamp="2013-04-17T10:24:11Z" >
  <Command ID="C001" Type="SubmitQueueEntry">
    <QueueSubmissionParams URL="http://myip/givejob.php?job10023"
      ReturnURL="http://myip/jobcompleted.php" />
  </Command>
</JMF>

<JMF xmlns="http://www.CIP4.org/JDFSSchema\_1\_1" SenderID="RIP01"
    TimeStamp="2013-04-17T10:25:28Z">
  <Response ID="R004" refID="C001" Type="SubmitQueueEntry">
    <QueueEntry JobID="JDF0034" QueueEntryID="1289829839283"
      Status="Waiting"/>
    <Queue DeviceID="blahblah" Status="Running">
  </Response>
</JMF>

```

JMF response if error:

```

<JMF xmlns="http://www.CIP4.org/JDFSSchema\_1\_1" SenderID="RIP01"
    TimeStamp="2013-04-17T10:25:28Z">
  <Response ID="R004" refID="C001" Type="SubmitQueueEntry"
    ReturnCode="120" >
    <Notification Class="Error">
      <Comment>URL cannot be resolved</Comment>
    </Notification>
  </Response>
</JMF>

```

Job submission scenarios

JDF jobs can currently be submitted by:

- Placing a JDF file in a hotfolder
- Sending a JDF file by HTTP (unidirectional channel)
- Placing a JMF SubmitQueueEntry message into a hotfolder (unidirectional channel)
- Sending a JMF SubmitQueueEntry message by HTTP (bidirectional channel)

Direct JDF submission by HTTP

The MIS can send directly JDF jobs to the Bridge HTTP interface (as defined by Base URL, HTTP port and HTTP Base Path in the Bridge configuration). The job resources (input files) should be referenced by http:// URIs in the JDF file. The bridge will retrieve it when the job is activated. There is no possibility to send job status back to the MIS in this case (see JMF submission below)

Direct JDF submission in hotfolder

The MIS can send directly JDF files into the Bridge hotfolder (as defined by hotfolder Path in the bridge configuration). The job resources (input files) should be referenced by file:// URIs, which can be absolute path (file:///path...) or relative path to the hotfolder (file://dir_in_hotfolder...). Usual scenarios are exported disk from the MIS which is mounted remotely on the bridge computer, or files written by the MIS in a sub-directory in the hotfolder prior to writing the JDF file). There is no possibility to send job status back to the MIS in this case (see JMF submission below).

Note: The hotfolder implementation of the bridge implies the following limitation: the hotfolder MUST be local on the computer running the bridge. However it can be remotely exported to other computers (the MIS). This restriction is due to the low-level system file monitoring.

JMF job submission by HTTP

The MIS can send a JMF SubmitQueueEntry to submit a job. The JDF job is retrieved by the Bridge at URL specified in QueueSubmissionParams@URL parameter in the JMF message. The resource retrieval is the same as in direct JDF submission. The JMF message can specify an URL (QueueSubmissionParams@ReturnURL or @ReturnJMF) to which the job feedback is to be sent after completion. The bridge will send the JDF file with completed AuditPool section to this URL.

When @ReturnJMF is specified, the URL must respond to SubmissionMethods JMF query and must specify Packaging="MIME" in the answer. The URL must accept a MIME multipart/related

package with ReturnQueueEntry JMF command as first part and the JDF as another part (linked by “cid” URL scheme). Any other method of returning job by ReturnQueueEntry is currently not supported. Added in Caldera 11.1.

JMF job submission in hotfolder

This scenario is rare, since the JMF messaging is used mainly with http communication. The MIS can write a JMF SubmitQueueEntry into the hotfolder to submit a job. The JDF job and job resources (input files) must be kept accessible by file:// URI scheme. The JDF file is specified by QueueSubmissionParams@URL parameter in the JMF message. The resource retrieval is the same as in direct JDF submission. The JMF message can specify an URL (QueueSubmissionParams@ReturnURL) to which the job feedback is to be written after completion. The bridge will write the JDF file with completed AuditPool section to this URL (which should use file:// scheme and should define an unique file name)

Note: The hotfolder implementation of the bridge implies the following limitation: the hotfolder MUST be local on the computer running the bridge. However it can be remotely exported to other computers (the MIS). This restriction is due to the low-level system file monitoring.

MIME multipart packaging

Starting from Caldera V10, the Bridge supports MIME multipart/related messages. Such message can contain JMF, JDF and binary resources. Content-ID header field is used to identify a specific part, which can be referenced to using “cid:” URL scheme.

Currently only base64 and 8bit content encoding is supported. The JMF/JDF parts should use 8bit content encoding (Content-Transfer-Encoding header) and binary data should always use base64 encoding. JMF parts should have Content-Type set to application/vnd.cip4-jmf+xml and JDF parts should have application/vnd.cip4-jdf+xml. In any case, the main message should be specified in the first part.

Important: Any header blocks must end with exactly one blank line. Any additional blank lines are considered to be part of the part body. Absence of blank line will cause the body to be considered as header and will be ignored. The boundary used to encapsulate the parts must be unique and must not appear anywhere in the content of any part.

Example: This example illustrates a MIME-packed job submission using JMF “SubmitQueueEntry” query with the JDF and a PDF file packaged as parts in the MIME message.

MIME-Version: 1.0

Content-Type: multipart/related; boundary=myuniqueboundary

--myuniqueboundary
Content-Type: application/vnd.cip4-jmf+xml
Content-Transfer-Encoding: 8bit

```
<?xml version="1.0" encoding="UTF-8"?>
<JMF xmlns="http://www.CIP4.org/JDFSschema_1_1" SenderID="MIS"
      Timestamp="2013-04-17T10:24:11Z" Version="1.4"
      ResponseURL="file:///some/path/my_jmf_response.jmf">
  <Command ID="C001" Type="SubmitQueueEntry">
    <QueueSubmissionParams URL="cid:myJDFpart"
      ReturnURL="file:///some/path/my_jdf_return.jdf" />
  </Command>
</JMF>
```

--myuniqueboundary
Content-Type: application/vnd.cip4-jdf+xml
Content-ID: <myJDFpart>
Content-Transfer-Encoding: 8bit

```
<?xml version="1.0" encoding="UTF-8"?>
<JDF Type="DigitalPrinting" ID="my JDF job" JobID="jdf1" JobPartID="1"
      Status="Waiting" Activation="Active">
  <ResourcePool>
    <RunList Class="Parameter" ID="RL01" Status="Available">
      <LayoutElement>
        <FileSpec URL="cid:myPDFfile" UserFileName="Flower.pdf"/>
      </LayoutElement>
    </RunList>
    <DigitalPrintingParams Class="Parameter" ID="DPP01" Status="Available"/>
    <Component Class="Quantity" ID="COMP01" Status="Unavailable"
      ComponentType="FinalProduct"/>
  </ResourcePool>
  <ResourceLinkPool>
    <RunListLink Usage="Input" rRef="RL01"/>
    <DigitalPrintingParamsLink Usage="Input" rRef="DPP01"/>
    <ComponentLink Usage="Output" rRef="COMP01"/>
  </ResourceLinkPool>
</JDF>
```

--myuniqueboundary
Content-Type: image/pdf
Content-ID: <myPDFfile>
Content-Transfer-Encoding: base64

zQzdHp3fNnGGAZmO+8/4WcDOrnbLPBlMqeLPud5MALhHHhgGwmftB2Of8+M17OTixexdbxFhhgBnK
PTu+bAaP9aOzizfFnGASEM2NcYspiaYL2LreIMuCjM7sYBofsv2d7mZaQzIgzYVxiwCWGGGAGGGG
AGGGGAGGGGUEM5/G5gEsmIv2f95zSK0aK+riOplMA2EydzXF9u76QhtZxhhgBo5YqN/O6t3F2a/yY
OUEpxPpDMoZlPY6Hjf8ANc4YubO42WMA6yghnRrhBoeWL5wa7+KqtDqr2cCo5nFdujtsrzXgSp07
mwhG+2yvRmYo9075s2zwRTolQgwCg82ELo26Wbo9075s4wwH/9k=

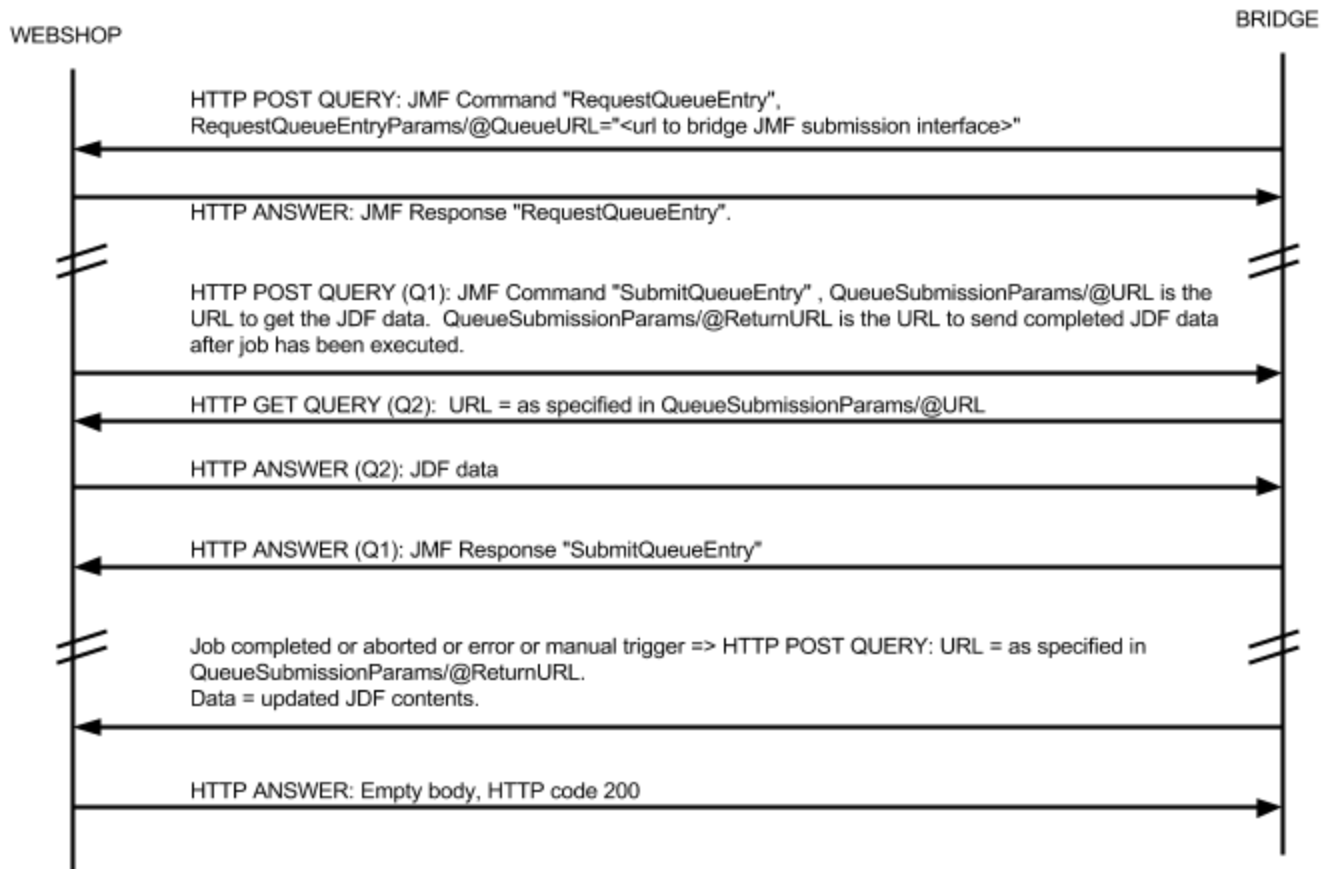
--myuniqueboundary--

Job submission as response to RequestQueueEntry JMF message

Systems that are not able to spontaneously submit new jobs must allow using the RequestQueueEntry JMF message. This message tells to the MIS that the bridge is awaiting new jobs if available. There are two ways to submit the jobs: Active and Passive. Only the active way complies to the JMF/JDF specification. There is no setting to choose between the two ways. The bridge will recognise automatically which one is used by analysing the answer to the RequestQueryEntry message. The RequestQueryEntry message is sent only by HTTP, never in hotfolder)

Active (JMF-compliant) RequestQueueEntry submission

The MIS receives a RequestQueueEntry message. It acknowledges the bridge, then sends new jobs to the QueueURL as set in the RequestQueueEntry. This requires unfiltered channels in both directions (bridge must be able to reach the MIS and the MIS must be able to reach the bridge)



Example:

1. Bridge requests new jobs (by user via Remote Browser) to preconfigured URL:

```

<JMF xmlns="http://www.CIP4.org/JDFSSchema_1_1" SenderID="RIP01"
      Timestamp="2013-04-17T10:24:11Z" >
  <Command ID="C001" Type="RequestQueueEntry">
    <RequestQueueEntryParams QueueURL="http://ip_of_rip/controller" />
  </Command>
</JMF>

```

2. WebShop response:

```

<JMF xmlns="http://www.CIP4.org/JDFSSchema_1_1" SenderID="WebShop01"
      Timestamp="2013-04-17T10:25:28Z">
  <Response ID="R004" refID="C001" Type="RequestQueueEntry" />
</JMF>

```

3. WebShop send jobs to the @QueueURL URL. The value of

Command/QueueSubmissionParams/@refID should be equal to Command/@ID of the RequestQueueEntry command:

```
<JMF xmlns="http://www.CIP4.org/JDFSSchema_1_1" SenderID="WebShop01"
      TimeStamp="2013-04-17T10:24:11Z" >
  <Command ID="ID0001" Type="SubmitQueueEntry">
    <QueueSubmissionParams URL="http://myip/givejob.php?job10023"
      ReturnURL="http://myip/jobcompleted.php" refID="C001" />
  </Command>
  <Command ID="ID0002" Type="SubmitQueueEntry">
    <QueueSubmissionParams URL="http://myip/givejob.php?job10025"
      ReturnURL="http://myip/jobcompleted.php" refID="C001" />
  </Command>
</JMF>
```

In the case WebShop is not interested on job feedback (@ReturnURL), it is possible to produce only one SubmitQueueEntry Command, referencing unique ProcessGroup JDF file.

Passive submission

The MIS receives a RequestQueueEntry message, and it sends the jobs as response on the same channel. This way of job submission is not JMF/JDF compliant but is a workaround when partial filtering has to be kept. It requires an unfiltered channel in one direction (the bridge must be able to contact the MIS).

Full working JDF exemple

Job with two source files (file1.pdf and file2.pdf), requesting two copies of each. Job is in progress (2 copies of file1.pdf have been printed).

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

<JDF ID="JDF0343322" JobID="JDF0034" JobPartID="1" Status="InProgress"
Activation="Active" Type="DigitalPrinting"
  Version="1.4" ICSVersions="Base_L1-1.4" DescriptiveName="Test Job"
  xmlns="http://www.CIP4.org/JDFSSchema_1_1"
  xmlns:cal="http://www.caldera.com/jdf"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <Comment Name="JobDescription">
    This is a Caldera test JDF job
  </Comment>

  <Comment Name="Tags">URGENT GLOSSY</Comment>
```

<Comment Name="DueDate">2013-04-17T08:00:00Z</Comment>

<ResourcePool>

<RunList Class="Parameter" ID="RL001" Status="Available" DocCopies="2"
Directory="file:///path_to_remote_MIS_storage/sources" DescriptiveName="Source
files" >

<LayoutElement>

<FileSpec URL="file1.pdf" UserFileName="Tile 1"/>

<FileSpec URL="file2.pdf" UserFileName="Tile 2"/>

</LayoutElement>

</RunList>

<DigitalPrintingParams Class="Parameter" ID="DPP001" Status="Available">

<cal:PrintConfig>

<modename>CMYK</modename>

<res_id res="200">200x200</res_id>

</cal:PrintConfig>

</DigitalPrintingParams>

<CustomerInfo ID="CI001" Class="Parameter" Status="Available"

CustomerID="993844544" CustomerOrderID="32e9932">

<Contact ContactTypes="Customer">

<Address City="Strasbourg" Country="France" PostalCode="67201"

Street="1 rue des Freres Lumiere"/>

<Company OrganizationName="Caldera"/>

<ComChannel ChannelType="Phone" Locator="tel:(+33)388210000"/>

<Person FirstName="Georges" FamilyName="Onetwothree"

JobTitle="technician">

<ComChannel ChannelType="Email"

Locator="mailto:georges@caldera.com"/>

</Person>

</Contact>

</CustomerInfo>

<Device ID="DEV001" Class="Implementation" DeviceID="Epson SC-S70600"
Status="Available"/>

<Component Class="Quantity" ComponentType="FinalProduct" ID="C0432"
Status="Unavailable"/>

</ResourcePool>

<ResourceLinkPool>

```

    <DigitalPrintingParamsLink Usage="Input" rRef="DPP001"/>
    <RunListLink Usage="Input" rRef="RL001"/>
    <CustomerInfoLink Usage="Input" rRef="CI001"/>
    <DeviceLink Usage="Input" rRef="DEV001"/>
    <ComponentLink Usage="Output" rRef="C0432"/>

</ResourceLinkPool>

<AuditPool>

    <Created AgentName="FlowPlus" AgentVersion="4.0"
        TimeStamp="2013-04-17T08:00:00Z"/>

    <ResourceAudit TimeStamp="2013-06-10T12:00:00Z" AgentName="CalderaBridge"
AgentVersion="9.20">
        <RunListLink Usage="Input" rRef="RL001">
            <cal:Status
FileSpecURL="file:///path_to_remote_MIS_storage/sources/file1.pdf">
                <job_id>5</job_id>
                <job_name>file1.pdf</job_name>
                <job_state idx="3">Finished</job_state>
                <job_error/>
                <server_name>Epson_SC-S70600</server_name>
                <print_width unit="Inchs">62.551389</print_width>
                <print_height unit="Inchs">25.255556</print_height>
                <media_width unit="Inchs">64.000000</media_width>
                <media_height unit="Inchs">25.373611</media_height>
                <media_usage_ratio>0.972818</media_usage_ratio>
                <nb_printed>2</nb_printed>
                <begin_time timestamp="1365084525">Thu Apr 4 16:08:45
2013</begin_time>
                <print_time_sec>7</print_time_sec>
                <op_time timestamp="1365084532">Thu Apr 4 16:08:52
2013</op_time>
                <ink_cons unit="ml" total="4.561729">
                    <ink name="Cyan" short="C" unit="ml">1.492432</ink>
                    <ink name="Magenta" short="M" unit="ml">1.502096</ink>
                    <ink name="Yellow" short="Y" unit="ml">1.567201</ink>
                    <ink name="Black" short="K" unit="ml">0.000000</ink>
                </ink_cons>
            </cal:Status>
        </RunListLink>
    </ResourceAudit>

</AuditPool>

</JDF>

```