Kodak NexPress Developer's Interface Reference

For System 14.0

Reference Specification

Version 14.0

April 2012

LEGAL

Eastman Kodak Company
2600 Manitou Road
Rochester, NY 14653 USA

Table of Contents

1 Introduction

The NexPress digital production color press is designed to handle a wide range of applications from short-run and quick turnaround, to variable data printing. It uses an open and standards-compliant architecture to integrate existing software applications. The NexPress digital production color press is driven by the NexPress front end controller. The NexPress front end is a scalable and open PDF-based product. It supports hot folder and JDF/JMF-based job submission and JMF-based job and press management.

Some of the key features of the NexPress front end are:

- Built upon Adobe® PDF-based workflow architecture
- Scalable hardware
- Accepts Job Definition Format (JDF) Job Tickets
- Supports hot folder submission
- Provides JDF/JMF interface for Job Ticket submission, queue management, and status messaging
- Supports common static page description language (PDL) formats including Adobe® PDF and Adobe® PostScript. Submission of a JDF Job Ticket requires PDF content.
- Supports variable data printing using ANSI Standard PPML/VDX.

1.1 Scope

This document is one of two used by third party software developers who design software controllers and output drivers that target the NexPress family of digital production color presses. The complete set of documents is:

- Kodak NexPress Developer's Interface Guide [KNDIG]
- Kodak NexPress Developer's Interface Reference [KNDIR] (This document)

The **Kodak NexPress Developer's Interface Guide** is the main user document. It references the [KNDIR], and provides descriptions of the various job submission, job management and press management operations. This guide also identifies the various content formats for the NexPress products, and describes the job submission methods and control interfaces using these content formats. Detail on the specific content formats is provided in the [KNDIR].

The **Kodak NexPress Developer's Interface Reference** provides technical detail on interfaces used in JDF and PPML/VDX submission and in control of the JDF portal. This technical companion document to the [KNDIG] is intended as a reference. The syntax and semantics of Device Capabilities, JDF Intent, JDF Process, JMF and PPML/VDX used by the NexPress front end is provided within this reference document.

These documents do not describe the interface between the NexPress front end and the NexPress print engine. This interface is not intended to be accessible to third-party workflow tool developers.

It is assumed that readers of this document have a working understanding of the Job Definition Format (i.e. JDF), the Adobe® Portable Document Format (i.e. PDF), and the PPML/VDX standard.

1.2 Glossary of Terms and Abbreviations

The table below explains terms and concepts used throughout this document, which may or may not be unique to the system environment described in this document.

Term	Definition
CGATS	Committee for Graphic Arts Technologies Standards – an ANSI accredited standards body that develops technical standards for the graphic arts industry. http://www.NPES.org
CIP4	International Cooperation for Integration of Processes in Prepress, Press, and Postpress. The consortium responsible for creating and maintaining the JDF Specifications. http://www.CIP4.org
Combined Process node	A JDF processes node definition that is an aggregate of several JDF processes. Such combined JDF process node definitions serve to model multiple function devices. See NexPress Combined Process Node.

Term	Definition
Component	Various versions or parts of semi-finished print products.
Device Capabilities; Device Capabilities File	The Device Capabilities file is a proprietary XML data file stored in the NexPress front end. It is named DevCaps.xml. This file provides a mechanism to inform client software applications of the system features and resources available on the *NexPress digital production color press*. Features include items such as output sheet delivery and finishing capabilities that are dependent on the press configuration. Resources include a complete characterization of all qualified substrates and their names, lists of installed fonts and ICC color profiles, and the names of defined Virtual Printer Hot Folders.
DFE	Digital Front End — The device that processes a print job (including impositioning and Ripping) and sends rasterized print data to the NexPress print engine for imaging onto media. It is referred throughout this document as the Kodak NexPress front end.
DIG	Developer's Interface Guide — A NexPress document that specifies the use of the various interfaces for submitting jobs and managing the press. Provides technical information for third-party application developers to successfully integrate their solution with the *Kodak NexPress digital production color press*.
Document	A meaningful unit of information. It may be represented in various ways including as a file, as a part of a file, or by printed output from a job.
DTD	Data Type Dictionary
Hot Folder	A hot folder is a file system directory that is monitored by a software process. Files placed into a hot folder are treated as input data and are processed by the software monitoring the hot folder. JDF uses this mechanism for file-based JDF job ticket submission.
Internal Job Queue	NexPress Job Queue containing jobs submitted through both the JDF Portal interface and Virtual Printer Hot Folders.
JDF	The Job Definition Format (JDF) is an evolving industry standard created and maintained by the International Cooperation for the Integration of Processes in Prepress, Press, and Postpress (CIP4). JDF is an extensible, XML–based format built on existing technologies such as the Portable Job Ticket Format (PJTF) from Adobe® and on the Print Production Format (PPF) from CIP3. JDF provides a container for a universal electronic job ticket for all kinds of print production data. A JDF structure can describe both the print product to be produced (intent) and the steps required to produce it (processes). The NexPress front end only consumes JDF process.
JDF device	As defined in the JDF Specification, a JDF device interprets JDF data, identifies a compatible process node definition, and employs a machine that executes it appropriately.
JDF job ticket	An XML data structure that specifies a print job. It may include both a print product description (product intent) and process definition (JDF Process). A JDF job ticket is comprised of one or more JDF nodes. Submission through the NexPress JDF Portal requires a JDF job ticket containing a single JDF process node.
JDF Portal	The NexPress front end's job submission interface that conforms to the JDF Specification and accepts JDF job tickets either by file-based submission to a JDF Hot Folder, or by a SubmitQueueEntry JMF message.
JDF Portal Job Queue	Queue containing only jobs submitted through the JDF Portal interface.
JDF process	A JDF node that specifies a unit of work or a step in the job workflow. Stitching, DigitalPrinting, Imposition, and LayoutPreparation are examples of JDF Processes. A JDF Process node often specifies the unit of work to be done by a JDF Device. The NexPress front end only supports a single combined process node that lists all the required JDF processes.

Term	Definition
JDF resources	JDF elements (such as Media, MediaIntent, or Component) that are input to or output from a JDF process or product intent node. JDF resources are always defined as sub-elements of the ResourcePool. The JDF device may consume physical resources that are input to JDF processes (e.g., substrate described by a Media resource input to a digital printing process).
JMF	Job Messaging Format — JDF Specification: *"JMF is a subset of JDF that handles communication among JDF controllers and equipment on the shop floor…. JMF can be used to establish a queue, discover the capabilities of a JDF-enabled device, determine the status of a device, e.g., "RIPing", "Idle", and so on."*
Job	A unit of work that can processed separately. A job within a printing system may be a part of a larger job from the perspective of a print shop. JDF specifies what needs to be produced in a job and how it is to be accomplished. Typically, a job consists of work done on one or more files, using a set of resources, to produce output.
Media Catalog	A data resource managed by the NexPress front end that contains a profile entry for each substrate, or medium, supported by this *NexPress digital production color press*. Each profile entry describes physical characteristics of a supported medium, and the medium name.
NCF	NexPress Capabilities Format. See also: Device Capabilities
NexPress digital production color press	The Kodak NexPress print engine and front end (DFE).
NexPress print engine	The printing device component of the NexPress digital production color press
NexPress Combined Process node	A JDF-Combined Process node (value of the ***Types*** attribute is *Combined*) that conforms to the restrictions identified in the ***Kodak NexPress Developer's Interface Reference.*** The *NexPress digital production color press* Combined Process Node includes LayoutPreparation, Imposition, ColorSpaceConversion, Interpreting, Rendering, Screening, and DigitalPrinting. Beginning with Release 10.1, only DigitalPrinting is required in JDF/@Types, other parameters will use default settings if not specified.
NexPress Conforming Content Files	Content files assembled from composite pages and specified in reader order, including Adobe® PDF documents for static jobs and ANSI CGATS.20 (PPML/VDX) for variable data jobs. NexPress Conforming Content Files may also represent a sequence of pre-imposed sheet surfaces for both static PDF jobs and variable PPML/VDX jobs; each PDF or PPML page definition is a pre-imposed layout to be printed 1-up onto a sheet surface.
NexPress front end	The DFE processor of the *NexPress digital production color press*
Page	A unit of content that is normally imaged onto a rectangular area of a surface of an output medium. The page is normally the unit of output to which medium selection, imposition, and a variety of other output processing options are applied.
PDF	Adobe® Portable Document Format
PDF/VT	Portable Document Format / Variable Transactional
PDL	Page Description Language (i.e. Adobe® PostScript).
PDM (VDX)	*Portable Digital Master version 2.01* – an early draft of what is now the fully balloted and accredited ANSI CGATS.20-2002 PPML/VDX standard. This is the optimized Structured Document Format used by earlier versions of the NexPress front end for representing the page content of a VDP job. Like PPML/VDX, this also is a PDF-based format that contains PPML data and is used to encode the many Recipient Instances of a VDP job in a highly optimized way. PDM support by the NexPress front end will be deprecated in the future.

Term	Definition
PJTF	Portable Job Ticket Format — Developed by Adobe®, this is is the internal data format for representing job ticket control information in the NexPress front end. JDF job tickets are converted to PJTF upon submission.
PPML	Personalized Print Markup Language — Name of the XML-based variable data mark-up language developed by the Print On Demand initiative (PODi). PPML provides semantics for object-level reusability of graphical content. The PDF-based PPML/VDX and PDM formats use PPML data to describe Instance documents and their pages.
PPML/VDX	Personalized Print Markup Language/Variable Data eXchange — The CGATS.20 ANSI standard PDF and PPML-based file format for representing variable content documents. A Structured Document Format that may be device and workflow-independent. It separates the variable data creative design process from the details of the print manufacturing process. This standard was accredited after the first release of the NexPress digital production color press. PDM is closely related to this standard. Like PDM, it is PDF-based and uses PPML data to describe Recipient Instances and their pages. PPML data is embedded within one of the PDF files of the set files that comprise an Instance.
Recipient Instance	The pages of a VDP content data stream that belong to a particular recipient. Represented in the PPML data of a PPML/VDX instance as a JOB element.
RIP	Raster Image Processor – converts vector and sampled image data interpreted from page content formats such as PDF, PostScript, and PPML/VDX into raster image data that is used to drive the print imaging apparatus of the digital printer.
SDK	Software Developer's Kit — A *collection of documentation and tools intended to enable* third party application developers to successfully integrate their variable data solutions (using ANSI PPML/VDX) with the *NexPress digital production color press*.
Sheet	A unit of a medium, typically a sheet of paper or an area on a continuous roll of roll-fed paper, on which output is printed. Sheets may be cut and folded after printing. For example, a sheet with no folds may contain one or two Pages, one per imaged side; a brochure with one fold may contains four Pages, one located on each side of the fold per side.
Structure component	A partial component of a print product such as the cover pages, body pages, dust cover, or insert pages of a book. Note that a book may be an assembly of structure components.
Structured Document Format	Page content definition format that is structured such that the data that makes up graphical page definitions are defined independent of each other (e.g. Adobe® PDF and ANSI PPML/VDX). The pages of a document using Structured Document Format may be indexed and the data can be efficiently accessed in random fashion.
URI	Uniform Resource Identifier as defined in RFC 2396: Uniform Resource Identifiers (URI)
URL	Uniform Resource Locator as defined in RFC 2396: Uniform Resource Identifiers (URI)
VDP	Variable Data Printing — The concept of customizing each print (or Instance) of a run of a digital print job.
VDX	Short term for PPML/VDX. See PPML/VDX
Virtual Printer Hot Folder	An active operating system folder that provides a mechanism to submit print job data for execution by the NexPress front end. Each Virtual Printer Hot Folder has a pre-assigned Virtual Printer Job Ticket that specifies processing requirements for the PDF, PDM2, PostScript, or PPML/VDX data submitted into it.

Term	Definition
VPJTT	Virtual Printer Job Ticket Template – Specifies settings required by NexPress front end to process content files submitted into a Virtual Printer Hot Folder. The NexPress Client software is used to specify its settings.
XML	Extensible Markup Language

1.3 Document Conventions

This section identifies the standard conventions used in this document. Following these conventions will help ensure consistency throughout the document.

1) PDF operators, PDF keywords, the names of keys in PDF dictionaries, and other predefined names are written in a bold sans serif type font; for example, the **ID** key.
2) Operands of PDF operators or values of dictionary keys are written in an italic sans serif font, for example the */Catalog* key.
3) PPML and JDF element and XML element names in general are written in a bold sans serif type font, for example the **DOCUMENT** element.
4) Values of attributes of XML elements are written in an italic sans serif font. For example: *LineArt.*
5) Attribute names of PPML and XML elements are written in a bold italic sans serif font. For example, the ***ProcessColorModel*** attribute of the **ColorColorantControl** element.
6) In some cases the Xpath notation may also be used when showing the relationship of an XML element attribute or sub-element to an element. For example: ***ColorantControl/@ProcessColorModel***.
7) Placeholders for normally variable information are written in an italic serif font. For example: "The first value specifies the number of *columns* of page cells and the second value specifies the number of *rows* of page cells in the multi–up grid".
8) Within examples, use of bold font has no technical significance and is used for emphasis only.

1.4 Constraints

1) Similar to the CIP4 Interoperability Conformance Specification (ICS) documents, all NexPress front end supported elements, sub-elements, and attributes are presented in this document in a tabular format with the following column headings: "Name or Value", "Manager", "Worker", and "Description". Please refer to [BICS] for how to read ICS documents if further understanding of notations and format is needed.
2) NexPress front end also conforms to the size limits for attribute values as specified in [BICS].
3) Refer to [JDF] for additional detail on specific JDF references mentioned in this document. Information from the JDF Specification will not be duplicated in this document whenever possible. This will help in maintenance of this document and help to eliminate any discrepancies with [JDF].
4) NexPress front end, System 10.1 uses JDF 1.3.

1.5 References

[1] Adobe Portable Document Format Reference Manual, Version 1.6, 2004, Adobe Systems Incorporated, [PDF]

[2] Adobe Portable Job Ticket Format Technical Note #5620, Version 1.1, April 1999, Adobe Systems Incorporated, http://partners.adobe.com/asn/developer/PDFS/TN/5620.pdf, [PJTF]

[3] JDF Specification, Release 1.3, September 2005, www.cip4.org, [JDF]

[4] XML Specification, Version 1.0, February 1998, www.w3.org, [XML]

[5] PPML Functional Specification, Version 2.0, 27-March-2002, Print On Demand Initiative (PODi), [PPML]

[6] ANSI CGATS.20:2002 Graphic Technology, Variable printing data exchange using PPML and PDF (PPML/VDX), Approval dated 8-July-2002, American National Standards Institute, Inc. available from NPES, [PPML/VDX]

[7] Application Notes for CGATS.20-2002, Prepared by CGATS SC6/TF2, August 2004, http://www.npes.org/standards/tools.html, [PPML/VDX-AN]

[8] Base Interoperability Conformance Specification (ICS), Version 1.0, December 2004, www.cip4.com, [BICS]

[9] Kodak NexPress Developer's Interface Guide, for System 9.0, Version 2.0, January 2007, Eastman Kodak Company, [KNDIG]

[10] Kodak NexPress Developer's Interface Reference, for System 9.0, Version 2.0, January 2007, Eastman Kodak Company, [KNDIR]

[11] ISO 16612-2 Graphic Technology – Variable Data Exchange Part 2: Using PDF/X-4 and PDF/X5 (PDF/VT-1 and PDF/VT-2) , First Edition, 15-August-2010. [PDF/VT]

2 Differences between Software Releases

2.1 Changes in System 14.0

- New Dry ink support: white, gold, neon pink
- Deprecate NexPress DryInk silver, renamed to NexPress DryInk pearlescent

2.2 Changes between System 12.0 and 12.1

2.2.1 JDF

- Matte Clear Dry Ink support
  - Media/@NXP:ClearCoat[Front|Back] = matteClearEasy
  - RenderingParams/@NXP:MTCLProcessColorantThreshold
  - RenderingParams/@NXP:MTCLThresholdApplication
- Silver Dry Ink support
- Custom Color Dry Ink support

2.3 Changes between System 11.2 and System 12.0

- Light Black dry ink support
- HD inks support
- Support for PDF/VT-1 documents

2.4 Changes between System 11.0/11.1 and System 11.2

2.4.1 JDF

- Red Fluorescing clear dry ink support:

2.5 Changes between System 10.1 and System 11.0/11.1

2.5.1 JDF

- Dimensional DryInk clear support:
  - Media/@NXP:ClearCoat[Front|Back] = raiseAllSimple
  - ColorantControl/@HotOffsetCompensation
  - RenderingParams/@NXP:DMCLProcessColorantThreshold
  - RenderingParams/@NXP:DMCLThresholdApplication
- Color management for RGB colorspaces may no longer be disabled.

2.5.2 Device Capabilities

- SupportedEngineSpeedList indicating the engine speeds for which the resource is supported has been added to:

- Printer
- PrinterMedium
- CurrentLoadedFifthColor
- SubstrateSupply
- DeliveryTypeDetails
- Delivery

- Printer includes PrinterSerialNumber to identify the engine by serial number.

2.6 Changes between System 10.1 and System 10.2

2.6.1 JDF

- Support for M700e finishing options
- M700e Resource Check warnings and errors are reported in JobPhase/@Comment for queued jobs.

2.7 Changes between System 10.0 and System 10.1

2.7.1 JDF

- **LocatorTTL (See Section 6.3) –**
  A multicast locator was added for use by the Creo Prinergy Workflow Controller.
- **JDF Combined Types (See Section 5.1) –**
  Restrictions have been removed that required a specific set of types in the JDF Process combined node. Now only DigitalPrinting is required.
- Most of the configuration files in \CDFE_CONFIG\InitialConfig\Printready for use by Heidelberg PrintReady version 2.1 have been removed.
- Use of the JDFStorage directory for completed JDF tickets has been deprecated.
- **Additional inks for Separations (See Section 5.3.1.1) –**
  Dry ink values were defined for micr, raised clear, and xd clear.
- **Additional ClearCoat options (See Section 5.12) –**
  Values added to support iqFlood, iqImage, xdPhoto, and xdGraphic.
- **Additional DeviceInfo status reported (See Section 3.2.1.8.1) –**
  DeviceCondition and StatusDetails are reported for press condition "NeedsAttention".
- **Resource Check details added to JobPhase (See Section 3.2.1.8.1.2) –**
  Resource Check warnings and errors are reported in JobPhase/@Comment for queued jobs.
- **Resource Query supported (See Sections 3.1.2 and 3.2.1.7) –**
  Reports descriptions and availability of media and dry ink resources.
- **Job processing controls defined in JDF (See Section 5.13.1) –**
  Custom attribute NXP:StopPoints created to stop job during processing.
- References to "NexStation" have been replaced with "NexPress".
- **Removed HDM:Shutdown (See Section 3.1.1) –**
  Removed JMF commands HDM:Shutdown and HDM:CheckFolderAccess. "Shutdown" is equivalent to HDM:Shutdown, and continues to be supported.

2.7.2 Device Capabilities

- **SecuredColorant (See Section 9.2.90) –**
  Identify colorants in the fifth station that support secure printing (such as MICR).

- **DeliveryTypeDetails for Finishing devices (See Section 9.2.22) –**
  List of finishers using the DFA interface in Delivery/DeliveryType/@DeliveryTypeDetails have been replaced by the single identifier "DFA".

2.8 Changes between System 9.0 and System 10.0

2.8.1 JDF

- JDFHotFolder is available as a network share whenever the JDF Portal Queue is open.
- PDF content files accompanying JMF SubmitQueueEntry in 3-part MIME can now be spooled directly to a file to eliminate the filesize restrictions of earlier releases. To enable this behavior, the content-file part must be identified within the MIME header using "Content-Disposition: Attachment" and "Content-Type: application/pdf".
- **RenderingParams/NXP:CLCDDIQParams (See Section 5.14.2) –**
  "DeviceDefault" is now interpretated as "Gradation" for the following attributes within RenderingParams/NXP:CLCDDIQParams: HTTextBW, HTGraphicsBW, HTImageBW, HTTextCMYK, HTLineCMYK, HTGraphicsCMYK, HTImageCMYK.
- **QueueSubmissionParams/@URL (See Section 3.1.1.6.1) –**
  Support for "ftp:" added to QueueSubmissionParams/@URL.

2.9 Changes between System 8.5 and System 9.0

2.9.1 JDF

- Postscript support has been added for content files in a JDF runlist. In previous releases only PDF was supported.
- Content-Type in the HTTP header of a JMF response will use application/vnd.cip4-jmf+xml if this type is used in the request.
- When the JDF queue is closed or blocked, any JDF Tickets dropped onto the JDF hot folder are deleted. In the previous release, these files remained in the hot folder directory until the queue was opened.
- **RunList resource – Layout Element: FileSpec (See Section 5.15.1.1) –**
  Support was added for URI scheme "FTP:" in **RunList/Layout/FileSpec/@URL**. This allows content files on an FTP server to be referenced within a JDF JobTicket.
- **RunList resource (See Section 5.13) –**
  Runlist/@Directory is supported. It can be used to identify a base path for FILE: URLs containing a relative path. Previously relative paths had to be located within the JDF Hotfolder.
- **Runlist/@Directory** can also be used to designate a search path for VDX PDL content. In this case the URL must be either a local path (drive letter required if not on the C: drive) or a UNC path. For example:

```
<RunList Directory="/parentDirectory/theDirectory"  …/>
<RunList Directory="d:/parentDirectory/theDirectory" …/>
<RunList Directory="//servername/sharename" …/>
```

- **JMF Node – Worker to Manager: Message Response (See Section 3.2.1) –**
  Whenever applicable, failed JMF commands return one of the standard exception codes defined in the JDF 1.3 specification.
- **Media resource: Location (See Section 5.12.1) –**
  Restrictions on media definition have been loosened so that **Location/@LocationName** is not required in some jobs. If omitted from the JDF, LocationName for body and cover will be determined using media partitions.
- **Media resource: Location (See Section 5.12.1) –**
  Additional values in **Media/Location/@LocationName** are supported.

- **JMF Node – Manager to Worker: QueueFilter (See Section 3.1.1.9) –**
  QueueFilter is now supported on incoming JMF commands and queries.
- **JMF Node – Worker to Manager: Message Response (See Section 3.2.1.1) –**
  Persistent JMF messages that are sent to registered subscribers will include **Notification/@JobID** and **Notification/@NXP:QueueEntryID** when these values are known.
- **Message Response: ResponseTypeObj – Submission Methods (See Section 3.2.1.3) –**
  JMF KnownMessages response includes SubmissionMethods query.
- **Device resource (See Section 5.5) –**
  When **Device/@DeviceID** is supplied in a JDF ticket, it is matched against the ***SenderID*** reported by the device in JMF responses.
- **NodeInfo resource (See Section 5.13) –**
  JDF/NodeInfo was moved to a resource in the JDF 1.3 specification. It is used to designate ***@TargetRoute*** for the completed JDF. The deprecated notation continues to be supported as well.
- **JDF Process Node (See Section 5.2) –**
  Support was added for **JDF/@Activation.** A value of "Held" in a job ticket will set the initial state of the submitted job to "Held".
- **DigitalPrintingParams resource (See Section 5.7) –**
  DigitalPrintingParams/@NXP:ColorFlow has been added to identify jobs that should adhere to the ColorFlow policy of the press.
- **RenderingParams resource (See Section 5.13.1) –**
  RenderingParams with selected proprietary attributes has been added.
- **ColorSpaceConversionParams resource: ColorSpaceConversionOp (See Section 5.4.1) –**
  ColorSpaceConversionOp/@ConvRGBGrayToBlack and **ColorSpaceConversionOp/@RGBGrayToBlackThreshold** have been added to control use of black colorant in RGB Graphics images.
- **NodeInfo resource (See Section 5.13) –**
  Proprietary attribute **NodeInfo/@NXP:DeleteWhenDone** has been added to the JDF ticket so that hotfolder jobs can designate whether to be automatically removed upon completion or abort.
- **Documentation added for Portal Configuration File (See Section 6) –**
  The JDF Portal Configuration file is not new, but the configuration file was not previously referenced in this document. For System 9.0, usage of the configuration value bDeleteWhenDone has been enhanced to control auto-delete behavior when submitted JDF hotfolder jobs reach completion or are aborted.
- **Device/ModelName (See Section 3.2.1.8.1.1 and Section 5.5) –**
  This attribute has been defined to facilitate rejection of a JDF ticket for situations in which the JDF was written using device-specific attributes and values that may be interpreted incorrectly on other JDF device. No action is taken in System 9.0 from this attribute.

2.9.2 Device Capabilities

- **Device Capabilities Elements: SubstrateType (See Section 9.2.102) –**
  New SubstrateType members: Plain, SingleCoated, DoubleCoated, Recycled, Texture, Film, Label, Vellium, and Bond.
- **Device Capabilities Elements: SurfaceType (See Section 9.2.104) –**
  New SurfaceType members: Normal, Tabbed, and Punched.
- **Device Capabilities Elements: ColorType (See Section 9.2.12) –**
  Substrate ColorName replaced by ColorType enumeration.
- **Device Capabilities Elements: ColorList (See Section 9.2.10) –**
  New element ColorList.
- **Device Capabilities Elements: PrinterPlatform (See Section 9.2.76) –**
  NexPress_M700 added to PrinterPlatform.
- **Device Capabilities Elements: SystemMedium (See Section 9.2.106) –**
  ColorType added to SystemMedium.

- **Device Capabilities Elements: MinSize and MaxSize (See Sections 9.2.60 and 9.2.54) –**
  PixelRectangle replaced by individual Width and Height members in definition of MinSize and MaxSize.
- **Device Capabilities Elements: FeederType (See Section 9.2.26) –**
  New element FeederType to characterize SubstrateSupply in the P1 engine.
- **Device Capabilities Elements: FontType (See Section 9.2.33) –**
  New FontType member PostScriptTTF.
- **Device Capabilities Elements: DeliveryTypeDetails (See Section 9.2.22) –**
  New DeliveryTypeDetails members HighCapacity, ProofWaste, and Invalid.

2.10 Changes between System 8.4 and System 8.5

The primary development goal for the NexPress JDF Portal in System 8.4 was to support Base ICS Level 2. The Base ICS (Interoperability Conformance Specification)[BICS] is published by CIP4. In System 8.5, development focused on aspects of ICS Level 3 support that were thought to have the greatest benefit for users of the NexPress.

2.10.1 JDF

- Support was added for MIME packaging of content files. MIME packaged submissions using a MIME file to the JDF Portal Hot Folder, or a 3-part MIME with JMF SubmitQueueEntry are both supported.
- The location of the JDF Portal Hot Folder has changed to \\<servername>\HotFolder\JDFHotfolder.
- The completed JDF file can be sent to a web server by specifying "HTTP://servername:PORT#/" in the URI for ***TargetRoute*** or ***ReturnURL***.
- Support is extended for PDF version 1.6 as produced by Adobe® Acrobat® 7.
- To assist developers who are integrating a controller with the JDF Portal, all JMF communications received and sent by the NexPress front end are logged. Only JMF messages and MIME-encoded JDF job tickets are written to the log; received PDF content is not saved.
- **RunList resource: Disposition (See Section 5.15.2) –**
  Support for **FileSpec/Disposition** was added to provide automatic deletion of content files. By default, the content file is preserved.
- **JMF: QueueSubmissionParams / *@Disposition* (See Section 3.1.1.6.1) –**
  If supplied the job will not be automatically removed from the queue when status becomes completed or aborted. No attribute values are supported.
- **JMF: QueueSubmissionParams / *@ReturnURL* (See Section 3.1.1.6.1) –**
  Http support added.
- **JMF: QueueSubmissionParams / *@URL* (See Section 3.1.1.6.1) –**
  "cid:", "file:", and "http:" schemes supported.
- **JMF: Message Query (See Section 3.1.1.9) –**
  Support added for SubmissionMethods; it returns the hot folder location.
- **JMF: StatusResponse: JobPhase(See Section 3.2.1.8.1.2) –**
  Job phase *FailedTestRun* reported for JDF failed verification.
- **JDF Product Intent: JDF Product Intent Node / JDF (See Section 4.2)** –
  NexPress Combined Process Node. A root JDF intent node may be submitted to the NexPress for processing if the JDF intent includes a NexPress Combined Process Node. The job ticket is constructed entirely from the JDF Process; the JDF intent is ignored.
- **JDF Process: JDF Process Node (See Section 5.2) –**
  The NexPress Process Node can be anywhere in the submitted JDF ticket, and any JDF Product nodes WILL BE ignored in favor of the NexPress Combined Process node.
- **JDF Process: NodeInfo / TargetRoute (See Section 5.13) –**
  The completed JDF file can be sent to a web server by specifying "HTTP://servername:PORT#".
- **JDF Process: ScreeningParams Resource (See Section 5.15.2) –**
  The NexPress JDF Portal does not enumerate screen names, however legacy screen enumerations continue to be

supported. Supported ScreeningID values are listed in the NexPress Device Capabilities File. Supported ScreeningParams values are "Classic", "Optimum", "Line", "Supra", and "Stochastic".

2.10.2 Device Capabilities

- **Device Capabilities Elements: DeliveryTypeDetails (See Section 9.2.22) –**
  Additional delivery devices identified. "BookletMaker" replaced with "WatkissVarioBookletMaker", "WatkissVarioBookletMaker WithSpinemaster", "WatkissGen2BookletMaker", "WatkissGen2BookletMaker WithSpinemaster", "Coverbind", and "NonDfa".
- **Device Capabilities Elements: ExternalControllerType (See Section 9.2.25) –**
  Replaced "SPIRE" with "CREO_PODS".
- **Device Capabilities Elements: I2Pplatform (See Section 9.2.36)**
- **Device Capabilities Elements: Printer (See Section 9.2.70) –**
  I2Pplatform added.
- **Device Capabilities Elements: ScreenType (See Section 9.2.89) –**
  New screen types "Stochastic" and "None".

3 JMF Reference

The JDF Portal processes JMF and JDF content using a policy of Best Effort. With few exceptions, an omitted attribute or an incorrect value will not fail submission. All received JMF and JDF must, however, be properly formed XML. A JDF ticket or JMF request will be summarily rejected if XML parsing fails.

This chapter documents the NexPress support for JMF as referenced in the [KNDIG]. Use of JMF for the NexPress front end is provided in [KNDIG] and not here.

The NexPress front end and NexPress workflow tools support only the attributes, values and sub-elements of the resources identified below. Any attributes, values or sub-elements of the resource other then those defined below will be ignored.

3.1 JMF Node – Manager to Worker

For this section, many tables have their columns as Manager and Worker. The Manager in this case is sending a JMF message and the Worker is receiving the JMF message. The Manager is the Client Application accessing the NexPress front end and the Worker is the NexPress front end responding to the Client Application. "Client Application" is used in the broadest terms to indicate any entity accessing the NexPress JDF interface. The notation is similar to that in the Base ICS [BICS], except a value of "r" in the "Worker" column indicates the NexPress requires the attribute, element, or value; "r?" indicates the field is supported and will be used if supplied.

Name or Value	Manager	Worker	Description
DeviceID	**w?**	**!r**	Data Type: *string* Not read or written by NexPress for JMF nodes
SenderID	**w**	**!r**	Data Type: *string* Not read by NexPress for JMF nodes
TimeStamp	**w?**	**r?**	Data Type: *dateTime*
Version	**w**	**r?**	Data Type: *JDFJMFVersion*
1.0			
1.1			
1.2	**w?**	**r?**	JDF spec version 1.2
1.3	**w?**	**r?**	JDF spec version 1.3
xmlns	**w**	**r?**	Data Type: *URI*
http://www.CIP4.org/JDFSchema_1_1	**w**	**r?**	
Message	**w**	**r**	Abstract element. One message supported. See individual Message section below for each message family.
Command	**w?**	**r?**	Message should be either command or query
Query	**w?**	**r?**	Message should be either command or query

3.1.1 Message – Command

Name or Value	Manager	Worker	Description
ID	**w**	**r**	Data Type: *ID* Value established by Producer.

Name or Value	Manager	Worker	Description
Type	**w**	**r**	Data Type: *NMTOKEN* Only one command ***Type*** value allowed.
AbortQueueEntry	**w?**	**r?**	
CloseQueue	**w?**	**r?**	
HoldQueue	**w?**	**r?**	
HoldQueueEntry	**w?**	**r?**	
OpenQueue	**w?**	**r?**	
ResumeQueue	**w?**	**r?**	
RemoveQueueEntry	**w?**	**r?**	
StopPersistentChannel	**w?**	**r?**	
SubmitQueueEntry	**w?**	**r?**	
SuspendQueueEntry	**w?**	**r?**	
Shutdown	**w?**	**r?**	Resets the NexPress JDF Portal.
CommandTypeObj	**w←**	**r**	Abstract element. Not all ***Type*** values listed require this element. See **CommandTypeObj** elements below for required elements.
xsi:type	**w?**	**r?**	Data Type: *NMTOKEN* NexPress does not do schema validation on received JMF/JDF even when this value is supplied.

3.1.1.1 CommandTypeObj – **AbortQueueEntry**

Name or Value	Manager	Worker	Description
QueueEntryDef	**w**	**r**	See **QueueEntryDef** below.
QueueFilter	**w?**	**r?**	Specify queue contents returned in the response. See **QueueFilter** element below.

3.1.1.2 CommandTypeObj – **HoldQueueEntry**

Name or Value	Manager	Worker	Description
QueueEntryDef	**w**	**r**	See **QueueEntryDef** below.
QueueFilter	**w?**	**r?**	Specify queue contents returned in the response. See **QueueFilter** element below.

3.1.1.3 CommandTypeObj – **RemoveQueueEntry**

Name or Value	Manager	Worker	Description
QueueEntryDef	**w**	**r**	See **QueueEntryDef** below.
QueueFilter	**w?**	**r?**	Specify queue contents returned in the response. See **QueueFilter** element below.

3.1.1.4 CommandTypeObj – **ResumeQueueEntry**

Name or Value	Manager	Worker	Description

Name or Value	Manager	Worker	Description
QueueEntryDef	**w**	**r**	See **QueueEntryDef** below.
QueueFilter	**w?**	**r?**	Specify queue contents returned in the response. See **QueueFilter** element below.

3.1.1.5 CommandTypeObj – **StopPersistentChannel**

Name or Value	Manager	Worker	Description
StopPersChParams	**w**	**r**	See **StopPersChParams** below.

3.1.1.5.1 StopPersChParams

Name or Value	Manager	Worker	Description
URL	**w**	**r**	Same as ***URL*** set in query to establish the persistent channel.
ChannelID	**w**	**r**	Data Type: *NMTOKEN* Same as ***ID*** used in query to establish the persistent channel

3.1.1.6 CommandTypeObj – **SubmitQueueEntry**

Name or Value	Manager	Worker	Description
QueueSubmissionParams	**w**	**r**	See **QueueSubmissionParams** below.
QueueFilter	**w?**	**r?**	Specify queue contents returned in the response. See **QueueFilter** element below.

3.1.1.6.1 QueueSubmissionParams

Name or Value	Manager	Worker	Description
Disposition	**w?**	**r?**	If supplied, the job will not be automatically removed from the queue when status becomes completed or aborted. No attribute values are supported.
Priority = 50	**w**	**r?**	Data Type: *integer* NexPress only adjusts order of jobs not yet submitted to the Internal Queue
1 - 100	**w?**	**r?**	Default Value: 50.
ReturnURL	**w?**	**r?**	Data Type: *URL* "file:" and "http:" schemes supported.
URL	**w**	**r?**	Data Type: *URL* "cid:", "file:", "ftp:", and "http:" schemes supported.

3.1.1.7 CommandTypeObj – **SuspendQueueEntry**

Name or Value	Manager	Worker	Description
QueueEntryDef	**w**	**r**	See **QueueEntryDef** below.
QueueFilter	**w?**	**r?**	Specify queue contents returned in the response. See **QueueFilter** element below.

3.1.1.8 QueueEntryDef

Name or Value	Manager	Worker	Description
QueueEntryID	**w**	**r**	Data Type: *string* Refers to specific queue entry

3.1.1.9 QueueFilter

Name or Value	Manager	Worker	Description
MaxEntries	**w?**	**r?**	Data Type: *integer* Maximum number of entries to be returned in message response. If not specified, all matching entries are returned.
OlderThan	**w?**	**r?**	Data Type: *dateTime* Return only queue entries with a submission time older than or equal to that specified. If omitted, all entries matching other filter options are returned.
NewerThan	**w?**	**r?**	Data Type: *dateTime* Return only queue entries with a submission time newer than or equal to that specified. If omitted, all entries matching other filter options are returned.
QueueEntryDetails = Brief	**w?**	**r?**	Data Type: *enumeration* Specifies the amount of information provided in matching queue entries for the returned queue.
None	**w?**	**r?**	Do not fill in QueueEntry elements in the queue.
Brief	**w?**	**r?**	Returned QueueEntry elements omit job phase information.
JobPhase	**w?**	**r?**	Returned QueueEntry elements include in phase information in addition to that returned by "Brief".
JDF	**w?**	**r?**	Returned QueueEntry elements include the original JDF ticket in addition to all information returned by "JobPhase".
StatusList	**w?**	**r?**	Data Type: *enumerations* Return only queue entries with a status matching one of the entries in specified StatusList. If omitted, all entries matching other filter options are returned.
Running	**w?**	**r?**	
Waiting	**w?**	**r?**	
Held	**w?**	**r?**	
Removed	**w?**	**r?**	
Suspended	**w?**	**r?**	
Completed	**w?**	**r?**	
Aborted	**w?**	**r?**	
QueueEntryDef	**w?**	**r?**	Return only queue entries matching this list of one or more queue entry IDs. See **QueueEntryDef** above. If omitted, all entries matching other filter options are returned.

3.1.2 Message – Query

Name or Value	Manager	Worker	Description
ID	**w**	**r?**	Data Type: *ID* Required only with subscription. Used in refID of response
QueryTypeObj	**w←**	**r**	Abstract element. The NexPress only supports this element for ***Type***=Status. See **QueryTypeObj** element below.
Subscription	**w?**	**r**	Only supported for ***Type***=Status. See **Subscription** below.
Type	**w←**	**r**	Data Type: *NMTOKEN* Only one query ***Type*** value allowed.
KnownMessages	**w?**	**r?**	
QueueStatus	**w?**	**r?**	
Resource	**w?**	**r?**	Request status of consumable resources. See below.
Status	**w?**	**r?**	See below.
SubmissionMethods	**w?**	**r?**	Provides location of the JDF hot folder.
NXP:DeviceCapabilities	**w?**	**r?**	Requests content of the Device Capabilities File.
xsi:type	**w?**	**r?**	Data Type: *NMTOKEN* NexPress does not do schema validation on received JMF/JDF even when value is supplied.
QueueFilter	**w?**	**r?**	Only applicable for queries that always return a queue (i.e. QueueStatus). See **QueueFilter** element above.

3.1.2.1 Subscription

Name or Value	Manager	Worker	Description
URL	**w**	**r**	Data Type: *URL* Scheme must be "http:". Port number is optional. If the receiving web server does not properly handshake subscriptions, the subscribed channel will be closed after 3 consecutive failed attempts.

3.1.2.2 QueryTypeObj – **Resource**

Name or Value	Manager	Worker	Description
ResourceQuParams	**w?**	**r?**	Only required for non-default values. See **ResourceQuParams** below.

3.1.2.2.1 ResourceQuParams

Name or Value	Manager	Worker	Description
Classes = Consumable	**w?**	**r?**	Data Type: *enumeration*
Consumable	**w?**	**r?**	Query for the status of consumable resources. Default Value.

Name or Value	Manager	Worker	Description
Exact = false	**w?**	**r?**	Data Type: *enumeration*
false	**w?**	**r?**	Details for queried resources is NOT requested. Default Value.
true	**w?**	**r?**	Response should provide details for the queried resources.

3.1.2.3 QueryTypeObj – **Status**

Name or Value	Manager	Worker	Description
StatusQuParams	**w?**	**r?**	Only required for non-default values. See **StatusQuParams** below.

3.1.2.3.1 StatusQuParams

Name or Value	Manager	Worker	Description
DeviceDetails = None	**w?**	**r?**	Data Type: *enumeration*
None	**w?**	**r?**	Minimal attributes are returned in **DeviceInfo**. Default Value.
Brief	**w?**	**r?**	Additional **DeviceInfo** attributes are returned. Includes attributes ***Manufacturer*** and ***DeviceType***.
Full	**w?**	**r?**	Maximum available**DeviceInfo** is returned. Includes a **Device** sub-element.
Details	**w?**	**r?**	Equivalent to DeviceDetails="Full"
QueueInfo = false	**w?**	**r?**	Data Type: *boolean*
true	**w?**	**r?**	A **Queue** element containing a list of **QueueEntry** is returned.
false	**w?**	**r?**	No **Queue** element or **QueueEntry** is returned. Default Value.
JobDetails = None	**w?**	**r?**	Data Type: *enumeration*
None	**w?**	**r?**	Minimal **JobPhase** elements in **Response/DeviceInfo**. Default Value.
Brief	**w?**	**r?**	**JobPhase** includes additional attributes ***QueueEntryID*** and ***StartTime***.
Full	**w?**	**r?**	**JobPhase** also includes **JDF** job ticket.
JobID	**w?**	**r?**	Data Type: *string* If supplied, returned queue is filtered to only provide details of job matching JDF/@JobID. Note that JDF/@JobID is optional in a submitted JDF.
QueueEntryID	**w?**	**r?**	Data Type: *string* If supplied, returned queue is filtered to only provide details of job that has been assigned QueueEntryID. If StatusQuParams/@QueueEntryID is defined, then StatusQuParams/@JobID is ignored.

3.2 JMF Node – Worker to Manager

For this section, many tables have their columns as Manager and Worker. The Worker in this case is sending a JMF message and the Manager is receiving the JMF message. The Manager is the Client Application accessing the NexPress front end and the Worker is the NexPress front end responding to the Client Application. "Client Application" is used in the broadest terms to indicate any entity accessing the NexPress JDF interface. The notation is similar to that in the Base ICS [BICS], except a value of "w" in the "Worker" column indicates the NexPress will write the attribute, element, or value; "w←" indicates the field is supported and will be written when appropriate; "!w" indicates the field is not supported.

Name or Value	Worker	Manager	Description
DeviceID	**!w**	**r?**	Data Type: *string* Not used by NexPress in JMF Messaging.
SenderID	**w**	**r?**	Data Type: *string* <servername>
TimeStamp	**w**	**r?**	Data Type: *dateTime*
Version	**w**	**r?**	Data Type: *JDFJMFVersion*
1.3	**w**	**r?**	
xmlns	**w**	**r?**	Data Type: *URI*
http://www.CIP4.org/JDFSchema_1_1	**w**	**r?**	
Message	**w**	**r?**	Abstract element. One message supported. See individual Message section below for each message family.
Response	**w?**	**r?**	
Signal	**w?**	**r?**	

3.2.1 Message – Response

Name or Value	Worker	Manager	Description
ID	**w**	**r?**	Data Type: *ID* Value established by NexPress.
refID	**w**	**r?**	Data Type: *NMTOKEN* Same as ***ID*** from query or command.
ReturnCode	**w?**	**r?**	Data Type: *integer* May be omitted for value "0", Success. See [JDF] Appendix D for list of JMF Return Codes.
Subscribed	**w←**	**r?**	Data Type: *boolean* Provided only with responses for queries.
true	**w?**	**r?**	Query initiates a persistent channel. The channel is identified by ***Response@refID***.
false	**w?**	**r?**	No persistent channel established.
Type	**w**	**r?**	Data Type: *NMTOKEN* Same as ***Type*** set in query or command.

Name or Value	Worker	Manager	Description
AbortQueueEntry	**w?**	**r?**	
CloseQueue	**w?**	**r?**	
HoldQueue	**w?**	**r?**	
HoldQueueEntry	**w?**	**r?**	
OpenQueue	**w?**	**r?**	
Resource	**w?**	**r?**	
ResumeQueue	**w?**	**r?**	
RemoveQueueEntry	**w?**	**r?**	
StopPersistentChannel	**w?**	**r?**	
SubmissionMethods	**w?**	**r?**	
SubmitQueueEntry	**w?**	**r?**	
SuspendQueueEntry	**w?**	**r?**	
HDM:Shutdown	**w?**	**r?**	
KnownMessages	**w?**	**r?**	
QueueStatus	**w?**	**r?**	
Status	**w?**	**r?**	
NXP:DeviceCapabilities	**w?**	**r?**	
xmlns:xsi	**w**	**r?**	Data Type: *URI*
http://www.w3.org/2001/XMLSchema-instance	**w?**	**r?**	
xsi:type	**w**	**r?**	Data Type: *NMTOKEN* Value is generated through a concatenation of "Response" and the ***Type*** value, e.g. *ResponseHoldQueue*
Notification	**w?**	**r?**	Returned only when an improper **Command** or **Query** message is received. NexPress may respond to an invalid message without including **Notification** element. See **Notification** below. NOTE: Only SubmitQueueEntry consistently uses **Notification** on error.
ResponseTypeObj	**w?**	**r**	Abstract element. Not all ***Type*** values listed require this element. See **ResponseTypeObj** below.

3.2.1.1 Notification

Name or Value	Worker	Manager	Description
AgentName	**w**	**r?**	Data Type: *string*
AgentVersion	**w**	**r?**	Data Type: *string* NexPress software build version
Class	**w**	**r?**	Data Type: *enumeration*
Error	**w**	**r?**	
Error	**w?**	**r?**	Provides additional information for common errors. See **Error** below.

Name or Value	Worker	Manager	Description
JobID	**w?**	**r?**	JDF/@JobID of job originating Notification response. Omitted if undefined. Useful in signal notification of persistent channels.
NXP::QueueEntryID	**w?**	**r?**	QueueEntryID of job originating Notification response. Omitted if undefined. Useful in signal notification of persistent channels.
TimeStamp	**w**	**r?**	Data Type: *dateTime*
Type	**w**	**r?**	Data Type: *NMTOKEN*
Error	**w**	**r?**	
Comment	**w**	**r?**	Data Type: *telem* Free-form text describing failure

3.2.1.1.1 Error

Name or Value	Worker	Manager	Description
ErrorID	**w**	**r?**	Data Type: *string* Error code for condition; defined in [JDF] Appendix D.

3.2.1.2 ResponseTypeObj – AbortQueueEntry, CloseQueue, HoldQueue, HoldQueueEntry, OpenQueue, ResumeQueue, RemoveQueueEntry, SuspendQueueEntry

Name or Value	Worker	Manager	Description
Queue	**w**	**r?**	See **Queue** below.

3.2.1.3 ResponseTypeObj – SubmissionMethods

Name or Value	Worker	Manager	Description
HotFolder	**w**	**r?**	Data Type: *URL* Location of hot folder.
Packaging	**w**	**r?**	Data Type: *enumerations* List of packaging methods supported.
MIME	**w**	**r?**	JDF Portal accepts MIME Multipart/Related packaging of JMF and content files within JDF ticket.
URLSchemes	**w**	**r?**	Data Type: *NMTOKENS* List of schemes supported for retrieval of content files within JDF ticket. In the current release, all the following schemes are supported within a JDF ticket.
file	**w**	**r?**	file scheme. Note: The DFE must have unauthenticated access to referenced content files. Windows 2003 Server imposes new requirements on access not applicable in Windows 2000 Server..

Name or Value	Worker	Manager	Description
ftp	**w**	**r?**	FTP (File Transfer Protocol)
http	**w**	**r?**	HTTP (Hypertext Transport Protocol)

3.2.1.4 ResponseTypeObj – SubmitQueueEntry

Name or Value	Worker	Manager	Description
QueueEntry	**w**	**r?**	**QueueEntry** just submitted. Use to obtain assigned ***QueueEntryID***. See **QueueEntry** below.
Queue	**w**	**r?**	See **Queue** below.

3.2.1.5 ResponseTypeObj – KnownMessages

Name or Value	Worker	Manager	Description
MessageService	**w**	**r?**	Abstract element. See **MessageService** below.

3.2.1.5.1 MessageService

Name or Value	Worker	Manager	Description
Command = false	**w←**	**r?**	Data Type: *boolean* Identifies a command message.
false	**w?**	**r?**	Message is not a command. Default Value.
true	**w?**	**r?**	Message is supported as a command.
Persistent = false	**w←**	**r?**	Data Type: *boolean* Indicates whether Persistent Channel/Signal subscription is supported for this message.
false	**w?**	**r?**	Subscription is not supported. Default Value.
true	**w?**	**r?**	Persistent Channel/Signal subscription is supported.
Query = false	**w←**	**r?**	Data Type: *boolean* Identifies a query message.
false	**w?**	**r?**	Message is not a query. Default Value.
true	**w?**	**r?**	Message is supported as a query.
Signal = false	**w←**	**r?**	Data Type: *boolean* Indicates whether Persistent Channel/Signal subscription is supported for this message.
false	**w?**	**r?**	Subscription is not supported. Default Value.
true	**w?**	**r?**	Persistent Channel/Signal subscription is supported.
Type	**w**	**r?**	Data Type: *NMTOKEN* Message Type. Value is "HoldQueue", "SubmitQueueEntry", "Status", etc.
<MESSAGE-TYPE>	**w**	**r**	Message type described by this response entry.

3.2.1.6 ResponseTypeObj – **QueueStatus**

Name or Value	Worker	Manager	Description

Name or Value	Worker	Manager	Description
Queue	**w**	**r?**	**Queue**/@***Status*** is set to the current queue state. All current **QueueEntry** elements are returned. See **Queue** below for remaining attributes.

3.2.1.7 ResponseTypeObj – **Resource**

Name or Value	Worker	Manager	Description
ResourceInfo	**w**	**r?**	See **ResourceInfo** below.

3.2.1.7.1 ResourceInfo

Name or Value	Worker	Manager	Description
AvailableAmount	**w**	**r?**	Data Type: *integer* Percentage of resource remaining
Level	**w**	**r?**	Data Type: *enumeration*
OK	**w?**	**r?**	Device reports resource is Ready.
Low	**w?**	**r?**	Device reports Warning for resource.
Empty	**w?**	**r?**	Device reports Error for resource.
Unknown	**w?**	**r?**	Resource status cannot be determined.
Location	**w**	**r?**	Data Type: *string*
Cyan	**w?**	**r?**	Resource is Cyan dry ink station.
Magenta	**w?**	**r?**	Resource is Magenta dry ink station.
Yellow	**w?**	**r?**	Resource is Yellow dry ink station.
Black	**w?**	**r?**	Resource is Black dry ink station.
Fifth Station	**w?**	**r?**	Resource is dry ink station in Fifth position.
Unknown Toner Station	**w?**	**r?**	Unidentified dry ink resource.
Feeder_A	**w?**	**r?**	Substrate Feeder A in NPP or Classic press.
Feeder_B	**w?**	**r?**	Substrate Feeder B in NPP or Classic press.
Feeder_C	**w?**	**r?**	Substrate Feeder C in NPP or Classic press.
Feeder_D	**w?**	**r?**	Substrate Feeder D in NPP or Classic press.
Feeder_E	**w?**	**r?**	Substrate Feeder E in NPP or Classic press.
Feeder_F	**w?**	**r?**	Substrate Feeder F in NPP or Classic press.
Feeder_G	**w?**	**r?**	Substrate Feeder G in NPP or Classic press.
Feeder_H	**w?**	**r?**	Substrate Feeder H in NPP or Classic press.
Main	**w?**	**r?**	Main substrate supply in Classic press.
Upper	**w?**	**r?**	Upper substrate supply in Classic press.
Lower	**w?**	**r?**	Lower substrate supply in Classic press.
Feeder 1	**w?**	**r?**	Substrate Feeder 1 in M700 press.
Feeder 2	**w?**	**r?**	Substrate Feeder 2 in M700 press.
Feeder 3	**w?**	**r?**	Substrate Feeder 3 in M700 press.

Name or Value	Worker	Manager	Description
Feeder 4	**w?**	**r?**	Substrate Feeder 4 in M700 press.
Feeder 5	**w?**	**r?**	Substrate Feeder 5 in M700 press.
Feeder 6	**w?**	**r?**	Substrate Feeder 6 in M700 press.
Feeder 7	**w?**	**r?**	Substrate Feeder 7 in M700 press.
Feeder 8	**w?**	**r?**	Substrate Feeder 8 in M700 press.
Unknown	**w?**	**r?**	Unidentified substrate location.
ResourceName	**w←**	**r?**	Data Type: *string* Name of media(substrate) or dry ink resource. Provided only in response to ***ResourceQuParams/@Exact***=false.
Ink	**w←**	**r?**	See **Ink** below. Provided for **Ink** resource AND only in response to ***ResourceQuParams/@Exact***=true.
Media	**w←**	**r?**	See **Media** below. Provided for **Media** resource AND only in response to ***ResourceQuParams/@Exact***=true.

3.2.1.7.1.1.1 Ink

Name or Value	Worker	Manager	Description
Class=Consumable	**w**	**r?**	Data Type: *enumeration*
Consumable	**w?**	**r?**	Consumable resource. Default value.
ColorName	**w**	**r?**	Data Type: *string*
Black	**w?**	**r?**	
Yellow	**w?**	**r?**	
Magenta	**w?**	**r?**	
Cyan	**w?**	**r?**	
Clear	**w?**	**r?**	
Red	**w?**	**r?**	
Green	**w?**	**r?**	
Blue	**w?**	**r?**	
Custom	**w?**	**r?**	
Micr	**w?**	**r?**	
Raised Clear	**w?**	**r?**	
XD Clear	**w?**	**r?**	
Traceless Clear	**w?**	**r?**	
Fluorescent Clear	**w?**	**r?**	
White	**w?**	**r?**	
Gold	**w?**	**r?**	
Pearlescent	**w?**	**r?**	
Neon Pink	**w?**	**r?**	
Unknown	**w?**	**r?**	
Family	**w**	**r?**	Data Type: *NMTOKEN*

Name or Value	Worker	Manager	Description
NexPress DryInk	**w**	**r?**	
Status	**w←**	**r?**	Data Type: *enumeration* Provided only when status has value of "Available".
Available	**w**	**r?**	Resource is not in error.

3.2.1.7.1.1.2 Media

Name or Value	Worker	Manager	Description
Brand	**w**	**r?**	Data Type: *string* Name of media (substrate).
Class=Consumable	**w**	**r?**	Data Type: *enumeration*
Consumable	**w?**	**r?**	Consumable resource. Default value.
Dimension	**w**	**r?**	Data Type: *XYPair* Specifies the dimensions of media in points.
Status	**w**	**r?**	Data Type: *enumeration*
Available	**w?**	**r?**	Supply is closed and locked.
Unavailable	**w?**	**r?**	Supply is not closed or not locked.

3.2.1.8 ResponseTypeObj – **Status**

Name or Value	Worker	Manager	Description
DeviceInfo	**w**	**r?**	See **DeviceInfo** below.
Queue	**w←**	**r?**	Provided only in response to ***StatusQuParams/@QueueInfo***=true. See **Queue** below.

3.2.1.8.1 DeviceInfo

Name or Value	Worker	Manager	Description
CounterUnit	**w←**	**r?**	Data Type: *string* Provided only in response to ***StatusQuParams/@DeviceDetails***=Brief or Full.
Sheets	**w?**	**r?**	
DeviceCondition	**w←**	**r?**	Data Type: *enumeration* Provided only when the press itself is reporting an error that requires operator attention.
NeedsAttention	**w?**	**r?**	
DeviceStatus	**w**	**r?**	Data Type: *enumeration*
Idle	**w?**	**r?**	No active JDF jobs have been submitted to the Internal Job Queue.
Down	**w?**	**r?**	JDF Portal shutdown started
Running	**w?**	**r?**	One or more active JDF jobs has been submitted to the Internal Job Queue. NOTE: DeviceStatus remains running until the JDF queue is empty.
Setup	**w?**	**r?**	JDF Portal initialization in progress

Name or Value	Worker	Manager	Description
Cleanup – TBD	**w?**	**r?**	
Stopped	**w?**	**r?**	
Unknown	**w?**	**r?**	
Device	**w←**	**r?**	Provided only when ***DeviceDetails*** = *Full* in status request. See **Device** below.
JobPhase	**w**	**r?**	See **JobPhase** below.
StatusDetails	**w←**	**r?**	Data Type: *enumeration* Provided only when ***DeviceCondition***. = *NeedsAttention*.
Unknown	**w?**	**r?**	Press condition not described by another defined enumeration.
Color Mismatch	**w?**	**r?**	Press is reporting a Color Mismatch condition.
ColorMode Mismatch	**w?**	**r?**	Press is reporting a ColorMode Mismatch condition.
Destination Tray Mismatch	**w?**	**r?**	Press is reporting a Destination Tray Mismatch condition.
Jammed	**w?**	**r?**	Press is reporting a Paper Jam condition.
Substrate Mismatch	**w?**	**r?**	Press is reporting a Substrate Mismatch condition.

3.2.1.8.1.1 Device

Name or Value	Worker	Manager	Description
Class	**w**	**r?**	Data Type: *enumeration*
Implementation	**w**	**r?**	
DeviceID	**w**	**r?**	Data Type: *string*
DeviceType	**w**	**r?**	Data Type: *string*
Manufacturer	**w**	**r?**	Data Type: *string*
Eastman Kodak Company	**w**	**r?**	
ModelName	**w**	**r?**	Data Type: *string* The value of ModelName will match PrinterPlatform in DeviceCapabilities response.
NexPress_Classic	**w?**	**r?**	NexPress Classic Color Press
NexPress_NPP	**w?**	**r?**	NexPress New Paper Platform Color Press
NexPress_M700	**w?**	**r?**	NexPress Model M700 Color Press
Unknown	**w?**	**r?**	Unknown Press

3.2.1.8.1.2 JobPhase

Name or Value	Worker	Manager	Description
Amount	**w←**	**r?**	Data Type: *double* Quantity printed. Provided only when non-zero while job is actively printing.

Name or Value	Worker	Manager	Description
Comment	**w←**	**r?**	Supplies details for Resource Check Warnings and Errors. Each warning or error is a separate comment. Provided only when ***StatusDetails*** = *ResourceCheck*. See **Comment** below.
JDF	**w←**	**r?**	JDF Job Ticket for job. Provided only when ***JobDetails*** = *Full* is requested.
JobID	**w**	**r?**	Data Type: *string* JobID of root JDF node in Job Ticket
JobPartID	**w?**	**r?**	Data Type: *string* JobPartID of root JDF node in Job Ticket
PercentCompleted	**w**	**r?**	Data Type: *double* Percentage of job completed *in the current processing phase*. NOTE: Value will be 100% after submission, but before RIPping begins; and again after RIPping completes, but before Printing begins. When job is actively printing, PercentComplete refers to percentage of sheets printed.
QueueEntryID	**w**	**r?**	Data Type: *string* Unique identifier within JDF Queue. Assigned to job upon submission.
StartTime	**w**	**r?**	Data Type: *dateTime* Date and Time job added to JDF Queue.
Status	**w**	**r?**	Data Type: *enumeration*
Aborted	**w?**	**r?**	Job has been aborted due to error or operator action
Completed	**w?**	**r?**	Job has finished printing successfully
FailedTestRun	**w?**	**r?**	JDF failed validation
InProgress	**w?**	**r?**	Job will progress as resources are available. Multiple jobs may be InProgress simultaneously
Stopped	**w?**	**r?**	Operator must resume job for it to continue processing.
Unknown	**w?**	**r?**	Job state is undetermined
StatusDetails	**w?**	**r?**	Data Type: *string* Free-Form text providing additional status information. The values in this field may change in future releases.
Submitting	**w?**	**r?**	
Normalizing	**w?**	**r?**	
Formatting	**w?**	**r?**	
Resource check	**w?**	**r?**	
RIP'ing	**w?**	**r?**	
Printing	**w?**	**r?**	
TotalAmount	**w←**	**r?**	Data Type: *double* Quantity to be printed. Provided only when non-zero while job is actively printing.

3.2.1.8.1.2.1 Comment

Name or Value	Worker	Manager	Description
Language	**w**	**r?**	Data Type: *enumeration*
en	**w?**	**r?**	English. (Comments are only written in English.)
Name	**w**	**r?**	Data Type: *enumeration*
NxpPreflightError	**w?**	**r?**	Resource Check Error
NxpPreflightWarning	**w?**	**r?**	Resource Check Warning
Comment	**r?**	**w?**	Data Type: *telem* Free-Form text containing details of the Resource Check warning or error. Contents match that provided in the Job Ticket Editor Resource Check Tab of the NexPress Client interface. Each field in the message is delimited by '|'.

3.2.1.9 ResponseTypeObj – **NXP:DeviceCapabilities**

Name or Value	Worker	Manager	Description
Full Device Capabilities File XML data	**w**	**r?**	See reference on Device Capabilities File and [KNDIG]

3.2.1.10 Queue

Name or Value	Worker	Manager	Description
DeviceID	**w**	**r?**	Data Type: *string*
Status	**w**	**r?**	Data Type: *enumeration*
Blocked	**w?**	**r?**	Queue is Closed and Held
Closed	**w?**	**r?**	Queue is Closed but not Held
Held	**w?**	**r?**	Queue is Held but not Closed
Running	**w?**	**r?**	Queue is Open, not Held, but all processing threads are busy
Waiting	**w?**	**r?**	Queue is Open, not Held, and processing threads are available
QueueEntry	**w←**	**r?**	One or more queue entries reflective of the current queue. Only provided when queue is non-empty. See **QueueElement** below.

3.2.1.10.1 QueueEntry

Name or Value	Worker	Manager	Description
EndTime	**w←**	**r?**	Data Type: *dateTime* Date and Time job completed printing. Only set for Completed, Aborted, or Stopped jobs.

Name or Value	Worker	Manager	Description
JobID	**w**	**r?**	Data Type: *string* ***JobID*** of root **JDF** node in job ticket
JobPartID	**w?**	**r?**	Data Type: *string* ***JobPartID*** of root **JDF** node in job ticket
Priority = 50	**w**	**r?**	Data Type: *integer* Queue submission priority 1-100. NOTE: Priority only effects job order prior to placement in the Internal Job Queue. Use of Priority to influence job processing order is not recommended.
1 - 100	**w?**	**r?**	Default: 50
QueueEntryID	**w**	**r?**	Data Type: *string* Unique identifier within JDF Queue. Assigned to job upon submission.
StartTime	**w**	**r?**	Data Type: *dateTime* Essentially the same as SubmissionTime
Status	**w**	**r?**	Data Type: *enumeration*
Aborted	**w?**	**r?**	Job has been aborted due to error or operator action
Completed	**w?**	**r?**	Job has finished printing successfully
Held	**w?**	**r?**	Job held following HoldQueueEntry command
Running	**w?**	**r?**	Job active in the Internal Job Queue. "Running" does not necessarily indicate the job is currently printing.
Suspended	**w?**	**r?**	Job suspended following SuspendQueueEntry, an explicit job stop using the NexPress Client interface, or a Portal restart with the job queued
Waiting	**w?**	**r?**	Job awaiting submission to the Internal Job Queue
SubmisssionTime	**w?**	**r?**	Data Type: *dateTime* Date and time job was added to the Portal queue.
HDM:FirstStart	**w?**	**r?**	Data Type: *string* ***NodeInfo/@FirstStart*** of job ticket. NOTE: JDF Portal does not enforce ***@FirstStart*** when scheduling job.
HDM:ID	**w**	**r?**	Data Type: *string* Same as HDM:NodeID
HDM:JobPriority	**w**	**r?**	Data Type: *integer* Same as Priority
HDM:JobURL	**w?**	**r?**	Data Type: *string* Filename of first element in AncestorPool in job ticket
HDM:LastEnd	**w?**	**r?**	Data Type: *string* ***NodeInfo/@LastEnd*** of job ticket. NOTE: JDF Portal does not enforce ***@LastEnd*** when scheduling job.
HDM:NodeID	**w?**	**r?**	Data Type: *string* ***ID*** of root **JDF** node in job ticket

Name or Value	Worker	Manager	Description
HDM:SpawnID	**w?**	**r?**	Data Type: *string* ***SpawnID*** of root **JDF** node in job ticket
HDM:Start	**w?**	**r?**	Data Type: *string* ***NodeInfo/@Start*** of job ticket. NOTE: JDF Portal does not enforce ***@Start*** when scheduling job.
HDM:StatusDetails	**w?**	**r?**	Data Type: *string* Free-form comment containing additional status details when applicable.
NXP:CDFEJobID	**w**	**r?**	Data Type: *string* Job identifier used by the Internal Job Queue.
xmlns:HDM	**w**	**r?**	Data Type: *string*
www.heidelberg.com/schema/HDM	**w**	**r?**	
xmlns:NXP	**w**	**r?**	Data Type: *string*
www.nexpress.com	**w**	**r?**	

3.2.2 Message – Signal

Name or Value	Worker	Manager	Description
ID	**w**	**r?**	Data Type: *ID* Generated by NexPress
refID	**w**	**r?**	Data Type: *NMTOKEN* Matches ***Query/@ID*** to which the subscription was attached.
Type	**w**	**r?**	Data Type: *enumeration* Same as ***Type*** from Query initiating persistent channel.
Status	**w**	**r?**	NexPress only supports opening a persistent channel on the Status query
xmlns:xsi	**w**	**r?**	Data Type: *URI*
http://www.w3.org/2001/XMLSchema-instance	**w**	**r?**	
xsi:type	**w**	**r?**	Data Type: *NMTOKEN* Value is generated through a concatenation of "Signal" and the ***Type*** value, e.g. *SignalStatus*
Notification	**w?**	**r?**	Only used to report an error condition on a subscribed queue entry. Normally signal responses do not include **Notification** element. See **Notification** above.
ResponseTypeObj	**w**	**r?**	Abstract element. NexPress only supports a persistent channel on Status Query; see **ResponseTypeObj –** Status.

4 JDF Product Intent Reference

This chapter documents NexPress support for JDF Product Intent as referenced in the [KNDIG]. Examples demonstrating the use of JDF Product Intent for the NexPress front end are provided in [KNDIG] and not here.

For this section, many tables have their columns as Manager and Worker. The Manager in this case is the Client Application sending a JMF message and the Worker is the NexPress front end or Kodak Imposition Viewer. The notation is similar to that in the Base ICS [BICS], except a value of "r" in the "Worker" column indicates the NexPress or Imposition Viewer requires the attribute, element, or value; "r?" indicates the field is supported and will be used if supplied.

The NexPress front end andNexPress workflow tools support only the attributes, values and sub-elements of the resources identified below. Any attributes, values or sub-elements of the resource other then those defined below will be ignored. If an illegal or unsupported attribute value is identified, the default value will be used instead.

4.1 General use of JDF Product Intent with PPML/VDX

JDF Product Intent is defined by [JDF]; it refers to the use of JDF for describing the print product(s) to be produced. A JDF Product Intent job ticket describes "what" to produce rather than "how" to produce it. In contrast, process-oriented JDF describes the details of "how" to manufacture the print product(s). JDF Product Intent data in conjunction with the PPML/VDX structured page content data provides a complete job definition, enabling a production process to be defined that can create variable print products such as customized postcards, booklets, and brochures.

There are two basic methods to link the PPML/VDX data with the JDF Product Intent data:

- **Integrated JDF Intent** uses a JDF job ticket stored within or referenced from the **ProductIntent** sub-element of the **PPMLVDX** as defined by [PPML/VDX].
- **Referential JDF Intent** uses a JDF job ticket that is not integral to the PPML/VDX structured data. In this case, the JDF Product node's **RunList** resource refers to the PPML/VDX data (i.e. **JOB** and **DOCUMENT** elements) through the use of the JDF ***RunTags*** partition key.

This following specifies the general restrictions and constraints for using JDF Product Intent and PPML/VDX with the NexPress front end and NexPress workflow tools.

- Only **Referential JDF Intent** MUST be used.
- All **Integrated JDF Intent** WILL BE ignored whether **Referential JDF Intent** is present or not.
- When embedding **Referential JDF Intent** within a PPML/VDX-Layout file, it MUST be embedded as a PDF COS stream object within the body of the file, and referenced by a **/JDF** key entry within the **/Catalog** dictionary at the root of the file.
- NexPress Workflow tools, including Imposition Viewer, only support JDF resources **ArtDeliveryIntent, BindingIntent**, **LayoutIntent**, **MediaIntent, RunList** and **Component**. Other JDF Product Intent resources SHOULD NOT be used with a PPML/VDX job and will be ignored.
- The NexPress front end only supports embedded **MediaIntent**. All other JDF Product Intent data is ignored.
- All parameters of supported JDF Product Intent resources and those of their sub-elements that have JDF Span data types MUST have a value specified for the ***Actual*** attribute. This implies that all negotiation for the final product description has been completed, and actual values have been determined.
- Only a single JDF Product Intent node MUST be used to describe the print products of a PPML/VDX job.

4.2 JDF Product Intent Node

Name or Value	Manager	Worker	Description
DescriptiveName	**w**	**r**	Data Type: *string* Should contain a human-readable string describing the print product
ID	**w**	**r**	Data Type: *ID*

Name or Value	Manager	Worker	Description
JobID	**w?**	**r?**	Data Type: *string*
Type	**w**	**r**	Data Type: *NMTOKEN*
Product	**w**	**r**	
Version	**w**	**r**	Data Type: *JDFJMFVersion*
1.3	**w**	**r**	
xmlns	**w**	**r**	Data Type: *URI*
http://www.CIP4.org/JDFSchema_1_1	**w**	**r**	
Status	**w**	**r**	Data Type: *enumeration*
Waiting	**w**	**r**	
JDF	**w?**	**r?**	NexPress Combined Process Node. A root JDF intent node may be submitted to the NexPress for processing if the JDF intent includes a NexPress Combined Process Node. The job ticket is constructed entirely from the JDF Process; the JDF intent is ignored.
AuditPool	**w?**	**r?**	See **AuditPool** below.
ResourceLinkPool	**w**	**r**	See **ResourceLinkPool** below.
ResourcePool	**w**	**r**	See **ResourcePool** below.

4.2.1 ResourceLinkPool

Name or Value	Manager	Worker	Description
ResourceLink *	**w**	**r**	List of **ResourceLink** entries. See below.
BindingIntentLink	**w?**	**r?**	
ComponentLink	**w?**	**r?**	
LayoutIntentLink	**w?**	**r?**	
MediaIntentLink	**w?**	**r?**	

4.2.1.1 BindingIntentLink

Name or Value	Manager	Worker	Description
rRef	**w**	**r**	Data Type: *IDREF*
Usage	**w**	**r**	Data Type: *enumeration*
Input	**w**	**r**	

4.2.1.2 ComponentLink

Name or Value	Manager	Worker	Description
rRef	**w**	**r**	Data Type: *IDREF*
Usage	**w**	**r**	Data Type: *enumeration*
Output	**w**	**r**	

4.2.1.3 LayoutIntentLink

Name or Value	Manager	Worker	Description
rRef	**w**	**r**	Data Type: *IDREF*

Name or Value	Manager	Worker	Description
Usage	**w**	**r**	Data Type: *enumeration*
Input	**w**	**r**	

4.2.1.4 MediaIntentLink

Name or Value	Manager	Worker	Description
rRef	**w**	**r**	Data Type: *IDREF*
Usage	**w**	**r**	Data Type: *enumeration*
Input	**w**	**r**	

4.2.2 ResourcePool

Name or Value	Manager	Worker	Description
Resource *	**w**	**r**	List of **Resource** entries. See below for detail.
BindingIntent	**w?**	**r?**	Specifies the style of binding for an instance of a finished print product. **PPML/VDX Usage**: The **BindingIntent** characterization will be applied to all Recipient Instances (i.e. **JOB** elements) of a PPML/VDX instance.
Component	**w?**	**r?**	Represents the complete or partially finished print product that is the output of the printing process.
LayoutElement	**w←**	**r?**	Either a sub-element of or referenced from a **RunList** resource. **PPML/VDX usage**: Identifies the PPML/VDX layout file of a PPML/VDX instance that is used as the source of page content data.
LayoutIntent	**w?**	**r?**	Specifies one or two-sided printing of finished pages. **PPML/VDX Usage**: The **LayoutIntent** characterization will be applied to all Recipient Instances (i.e. **JOB** elements) of a PPML/VDX instance.
MediaIntent	**w?**	**r?**	Specifies one or more descriptions or characterizations of media to use for printed pages. A maximum of three **MediaIntent** sub-element characterizations are allowed. **PPML/VDX Usage**: The **MediaIntent** characterization will be applied to all Recipient Instances (i.e. **JOB** elements) of a PPML/VDX instance.
RunList	**w?**	**r?**	**PPML/VDX usage**: Identifies the sequence of Recipient Instances, or records of the PPML/VDX file. These records are referenced by the **LayoutElement** sub-element as page content of the print products described by the JDF Product node.

4.3 BindingIntent resource

Name or Value	Manager	Worker	Description
Class	**w**	**r**	Data Type: *enumeration*
Intent	**w**	**r**	
ID	**w**	**r**	Data Type: *ID*
Status	**w**	**r**	Data Type: *enumeration*
Available	**w**	**r**	
BindingColor	**w?**	**r?**	Data Type: *EnumerationSpan* Defines the color of the spine material of the binding. See **BindingColor** below.
BindingSide	**w←**	**r?**	Data Type: *EnumerationSpan* MUST be present if **BindingType** is defined and ***BindingType/@Actual*** other than *LooseBinding.* See **BindingSide** below.
BindingType	**w**	**r**	Data Type: *EnumerationSpan* Desired binding for the job. See **BindingType** below.
CoilBinding	**w?**	**r?**	Details of coil binding. MAY be present if ***BindingType/@Actual*** =*CoilBinding.* See [JDF] for structure of **CoilBinding** element.
PlasticCombBinding	**w?**	**r?**	Details of plastic comb binding. MAY be present if ***BindingType/@Actual*** = *PlasticComb.* See [JDF] for structure of **PlasticCombBinding** element.
SaddleStitching	**w?**	**r?**	Details of saddle stitching. MAY be present if ***BindingType/@Actual*** = *SaddleStitch.* See [JDF] for structure of **SaddleStitching** element.
WireCombBinding	**w?**	**r?**	Details of wire comb binding. MAY be present if ***BindingType/@Actual*** = *WireComb.* See [JDF] for structure of **WireCombBinding** element.

4.3.1 BindingColor

Name or Value	Manager	Worker	Description
DataType	**w**	**r**	Data Type: *enumeration*
EnumerationSpan	**w**	**r**	
Actual	**w**	**r**	Data Type: *enumeration* See [JDF] for possible values. MUST be present, implying all negotiation for final product description has been completed.

4.3.2 BindingSide

Name or Value	Manager	Worker	Description
DataType	**w**	**r**	Data Type: *enumeration*
EnumerationSpan	**w**	**r**	

Name or Value	Manager	Worker	Description
Actual	**w**	**r**	Data Type: *enumeration* MUST be present, implying all negotiation for final product description has been completed.
Top	**w?**	**r?**	
Bottom	**w?**	**r?**	
Right	**w?**	**r?**	
Left	**w?**	**r?**	

4.3.3 BindingType

Name or Value	Manager	Worker	Description
DataType	**w**	**r**	Data Type: *enumeration*
EnumerationSpan	**w**	**r**	
Actual	**w**	**r**	Data Type: *enumeration* **PPML/VDX usage**: If other than *LooseBinding*, all pages of related PPML **DOCUMENT** sub-elements of a **JOB** element MUST be bound. This disallows the definition of a Recipient Instance having multiple bound documents; multiple bound documents are not supported by the NexPress front end and workflow tools.
CoilBinding	**w?**	**r?**	
LooseBinding	**w?**	**r?**	
PlasticComb	**w?**	**r?**	
SaddleStitch	**w?**	**r?**	
WireComb	**w?**	**r?**	

4.4 Component resource

Name or Value	Manager	Worker	Description
Class	**w**	**r**	Data Type: *enumeration*
Quantity	**w**	**r**	
ComponentType	**w**	**r**	Data Type: *enumerations*
FinalProduct	**w?**	**r?**	
PartialProduct	**w?**	**r?**	
ID	**w**	**r**	Data Type: *ID*
ProductType = Unknown	**w?**	**r?**	Data Type: *NMTOKEN*
BackCover	**w?**	**r?**	
Book	**w?**	**r?**	
BookBlock	**w?**	**r?**	
BookCase	**w?**	**r?**	
Brochure	**w?**	**r?**	
BusinessCard	**w?**	**r?**	

Name or Value	Manager	Worker	Description
Cover	**w?**	**r?**	
FrontCover	**w?**	**r?**	
Label	**w?**	**r?**	
Poster	**w?**	**r?**	
Unknown	**w?**	**r?**	Default Value.
Status	**w**	**r**	Data Type: *enumeration*
Unavailable	**w**	**r**	

4.5 LayoutElement element

Name or Value	Manager	Worker	Description
Class	**w?**	**r?**	Data Type: *enumeration* Not required if **LayoutElement** is defined as an immediate sub-element of **RunList**.
Parameter	**w**	**r**	
ElementType	**w**	**r**	Data Type: *NMTOKEN*
MultiSet	**w**	**r**	
ID	**w?**	**r?**	Data Type: *ID* Not required if **LayoutElement** is defined as an immediate sub-element of **RunList**.
HasBleeds = false	**w?**	**r?**	Data Type: *boolean*
false	**w?**	**r?**	Default Value.
true	**w?**	**r?**	**PPML/VDX usage**: SHOULD have the value "*true*" if the **PAGE_DESIGN**/@***BleedBox*** attribute of the PPML data of the PPML/VDX-Layout file is present.
Status	**w?**	**r?**	Data Type: *enumeration*
Available	**w**	**r**	
FileSpec	**w**	**r**	See **FileSpec** below.

4.5.1 FileSpec

Name or Value	Manager	Worker	Description
Application	**w?**	**r?**	Data Type: *string* **PPML/VDX usage**: The descriptive name of the software application used to create the PPML/VDX instance.
AppOS	**w?**	**r?**	Data Type: *string*
AppVersion	**w?**	**r?**	Data Type: *string*
OSVersion	**w?**	**r?**	Data Type: *string*
UID	**w?**	**r?**	Data Type: *string*

Name or Value	Manager	Worker	Description
URL	**w?**	**r?**	Data Type: *URL* **PPML/VDX usage**: MUST NOT be present if the JDF data is embedded within the PPML/VDX-Layout file. Note that the absence of this attribute is interpreted as an implicit reference to the containing PPML/VDX-Layout file. This is the normal case supported by the NexPress Workflow tools.

4.6 LayoutIntent resource

Name or Value	Manager	Worker	Description
Class	**w**	**r**	Data Type: *enumeration*
Intent	**w**	**r**	
ID	**w**	**r**	Data Type: *ID*
Sides	**w?**	**r?**	Data Type: *enumeration* **PPML/VDX usage**: If both single-sided printing and two-sided printing is desired for different pages of a print product defined in the same PPML/VDX instance, then two sided printing must be asserted for all pages, and blank pages (i.e. **<PAGE/>**) must be explicitly defined in the PPML data as appropriate for specifying a blank side of a finished page.
OneSided	**w?**	**r?**	Implies front-side
OneSidedBack	**w?**	**r?**	
TwoSidedHeadToHead	**w?**	**r?**	
TwoSidedHeadToFoot	**w?**	**r?**	
FinishedDimensions	**w?**	**r?**	**PPML/VDX usage**: If present, its value overrides the value of the ***TrimBox*** attribute of the PPML **PAGE_DESIGN** element. The value of **FinishedDimensions** SHOULD match the value of the ***TrimBox*** attribute of the PPML **PAGE_DESIGN** element. See **FinishedDimensions** below.
Status	**w**	**r**	Data Type: *enumeration*
Available	**w**	**r**	

4.6.1 FinishedDimensions

Name or Value	Manager	Worker	Description
DataType	**w**	**r**	Data Type: *enumeration*
ShapeSpan	**w**	**r**	
Actual	**w?**	**r?**	Data Type: *shape*
x y z	**w**	**r**	an array of three numbers: width x, height y, and depth z, where x,y, and z are each >= 0.

4.7 MediaIntent resource

Name or Value	Manager	Worker	Description
Class	**w**	**r**	Data Type: *enumeration*
Intent	**w**	**r**	
ID	**w**	**r**	Data Type: *ID*
PartIDKeys	**w?**	**r?**	Data Type: *enumeration* Partitions the **MediaIntent**. See **MediaIntent** sub-element below.
RunTags	**w**	**r**	
PrePrinted = false	**w?**	**r?**	Data Type: *boolean* If true, identifies media as pre-printed stock.
false	**w?**	**r?**	Default Value.
true	**w?**	**r?**	
Status	**w**	**r**	Data Type: *enumeration*
Available	**w**	**r**	
BackCoatings	**w?**	**r?**	Data Type: *EnumerationSpan* Pre-process coating that has been applied to the back surface of the media. If not present, the default ***Actual*** value is the same as the **FrontCoating**/@***Actual*** value. Do not specify here for partitioned **MediaIntent**. See **BackCoatings** below.
FrontCoatings	**w?**	**r?**	Data Type: *EnumerationSpan* Pre-process coating that has been applied to the front surface of the media. Do not specify here for partitioned **MediaIntent**. See **FrontCoatings** below.
MediaIntent *	**w?**	**r?**	Up to 3 partitions are allowed. See **MediaIntent** characterization below.
MediaType	**w?**	**r?**	Do not specify here for partitioned **MediaIntent**. See **MediaType** below.
StockBrand	**w?**	**r?**	Do not specify here for partitioned **MediaIntent**. See **StockBrand** below.

4.7.1 MediaIntent sub-element

Name or Value	Manager	Worker	Description
RunTags	**w?**	**r?**	Data Type: *NMTOKENS* **PPML/VDX usage**: If present, it MUST have values that are equal to the values of one or more PPML **DOCUMENT**/@***Label*** attributes.
BackCoatings	**w?**	**r?**	Data Type: *EnumerationSpan* Pre-process coating that has been applied to the back surface of the media If not present, the default ***Actual*** value is the same as the **FrontCoating**/@***Actual*** value. See below.

Name or Value	Manager	Worker	Description
FrontCoatings	**w?**	**r?**	Pre-process coating that has been applied to the front surface of the media. See **FrontCoatings** below.
MediaType	**w?**	**r?**	See **MediaType** below.
StockBrand	**w?**	**r?**	See **StockBrand** below.

4.7.1.1 BackCoatings

Name or Value	Manager	Worker	Description. Refer to [JDF]
DataType	**w**	**r**	Data Type: *enumeration*
EnumerationSpan	**w**	**r**	
Actual = None	**w?**	**r?**	Data Type: *enumeration*
None	**w?**	**r?**	Default Value.
Glossy	**w?**	**r?**	
HighGloss	**w?**	**r?**	
Matte	**w?**	**r?**	
Satin	**w?**	**r?**	
Semigloss	**w?**	**r?**	

4.7.1.2 FrontCoatings

Name or Value	Manager	Worker	Description. Refer to [JDF]
DataType	**w**	**r**	Data Type: *enumeration*
EnumerationSpan	**w**	**r**	
Actual =None	**w?**	**r?**	Data Type: *enumeration*
None	**w?**	**r?**	Default Value.
Glossy	**w?**	**r?**	
HighGloss	**w?**	**r?**	
Matte	**w?**	**r?**	
Satin	**w?**	**r?**	
Semigloss	**w?**	**r?**	

4.7.1.3 MediaType

Name or Value	Manager	Worker	Description. Refer to [JDF]
DataType	**w**	**r**	Data Type: *enumeration*
EnumerationSpan	**w**	**r**	
Actual	**w**	**r**	Data Type: *enumeration*
Paper	**w**	**r**	

4.7.1.4 StockBrand

Name or Value	Manager	Worker	Description. Refer to [JDF]
DataType	**w**	**r**	Data Type: *enumeration*
StringSpan	**w**	**r**	
Actual	**w**	**r**	Data Type: *string*

Name or Value	Manager	Worker	Description. Refer to [JDF]
Media0	**w?**	**r?**	Equivalent to *Body*
Media1	**w?**	**r?**	Equivalent to *Cover*
Media2	**w?**	**r?**	Equivalent to *Insert*
Body	**w?**	**r?**	Logical name for body media as known to the NexPress front end.
Cover	**w?**	**r?**	Logical name for cover media as known to the NexPress front end.
Insert	**w?**	**r?**	Logical name for insert media as known to the NexPress front end.

4.8 RunList

Name or Value	Manager	Worker	Description. Refer to [JDF]
Class	**w**	**r**	Data Type: *enumeration*
Parameter	**w**	**r**	
ComponentGranularity	**w**	**r**	Data Type: *enumeration*
Set	**w**	**r**	PPML/VDX usage: Indicates that each PPML JOB element is interpreted as containing the set of content pages for a JDF Product Intent instance.
ID	**w**	**r**	Data Type: *ID*
Sets = **0 ~ -1**	**w?**	**r?**	Data Type: *IntegerRangeList* **PPML/VDX usage**: Refers to an index range of PPML **JOB** elements of the PPML/VDX file. **PPML/VDX usage**: If not present, all **JOB** elements present in the PPML data are included as input to the **RunList** in the order they appear in the PPML data. Default Value: 0 ~ -1
0 ~ -1	**w**	**r**	This (the default) is the only supported value.
Status	**w**	**r**	Data Type: *enumeration*
Available	**w**	**r**	
LayoutElementRef	**w**	**r**	See **LayoutElementRef** below.

4.8.1 LayoutElementRef

Name or Value	Manager	Worker	Description
rRef	**w**	**r**	Reference to **LayoutElement** resource. Specifies ID of corresponding **LayoutElement.**

5 JDF Process Reference

This chapter contains the NexPress support for JDF Process as referenced in the [KNDIG].

The NexPress front end is described as an integrated workflow system comprising a sequence of process operations, or phases. The JDF job ticket submitted to the NexPress front end must contain a JDF combined process node that specifies the sequence of process steps and their parameters. The NexPress front end is a JDF Device; its JDF Portal Interface accepts JDF job tickets that contain a specific JDF combined process node (the NexPress Combined Process Node).

For all descriptions within this chapter, the Manager is the Client Application accessing the NexPress front end and the Worker is the NexPress front end responding to the Client Application. "Client Application" is used in the broadest terms to indicate any entity accessing the NexPress JDF interface.

The NexPress front end supports only the attributes, values and sub-elements of the resources identified in the sections below. Any attributes, values or sub-elements of the resource other then those defined below will be ignored. If an illegal or unsupported attribute value is identified, the default value will be used instead.

5.1 Overview of the NexPress Combined Process Node

JDF combined process nodes are used to model complex JDF Devices. A JDF combined process is a linear point-to-point processing chain comprised of the fixed sequence of JDF processes described by the node's **JDF**/@***Types*** attribute.

Resources are produced and consumed in succession as the sequence of processes from the combined node are executed. Intermediate resources, however, are not explicitly defined in the JDF data. These implicit resources do not have **ResourceLink** elements in the node's **ResourceLinkPool**. Such resources are referred to in [JDF] as exchange resources.

Specific input resources in the process chain must be explicitly defined and linked in accordance with this specification. Resources are defined as XML data structures with sub-elements and attributes that control how each sub-process in the chain transforms input resources to output resources.

The NexPress front end extends the base functionality of a *Combined Ripping* process that is built upon the JDF processes *ColorSpaceConversion*, *Interpreting*, *Rendering* and *Screening*. The NexPress front end also supports the JDF In-Rip imposition model. This imposition model specifies sheet imposition layout using a set of high-level parameters to control placement, or layout, of reader pages and sheet marks (i.e. trim and fold marks) on the sheet surface. This parameterized layout is an implementation of the *LayoutPreparation* process and is controlled by the **LayoutPreparationParams** resource. For JDF Process submitted to the the NexPress, *LayoutPreparation* and *Imposition* are added before the *Combined Ripping* process names and *DigitalPrinting* is added to the end of the process name list. This specific combination of JDF processes is referred to as the NexPress Combined Process Node.

The JDF Processes supported by the NexPress front end are listed here in processing order. Within the NexPress Combined Process Node, the output resources of one process are the implicit input resources of the next; additional explicit input resources can also be specified. Implicit resources are identified by brackets (e.g. [**ImplicitResource**]) in the sections that follow.

The NexPress requires that the list of combined types in JDF/@Types include "DigitalPrinting". Additional types should also be specified as appropriate; default values will be used for any required attributes that are omitted from the JDF job ticket.

5.1.1 NexPress Conforming Content Files

The NexPress front end workflow is based on *NexPress Conforming Content Files*. Such content files must first be assembled from composite pages and specified in reader order.

NexPress Conforming Content Files include PostScript and Adobe PDF documents for static jobs, and ANSI CGATS.20 (PPML/VDX) conforming data for variable data jobs.

NexPress Conforming Content Files may also represent a sequence of pre–imposed sheet surfaces for both static PDF and variable PPML/VDX jobs in which each PDF or PPML page definition is a pre-imposed layout printed 1-up onto a sheet surface.

5.1.2 LayoutPreparation process

LayoutPreparation defines the **Layout** resource for the *Imposition* process. The *LayoutPreparation* process is executed by the NexPress front end's built-in imposition engine.

The following identifies the explicit and [implict] resources for *LayoutPreparation*:

Name or Value	I/O	Description
LayoutPreparationParams	Input	Set of parameters required to control *LayoutPreparation*. From Manager.
RunList (Document)	Input	A structured list of incoming page contents conforming to the *NexPress Conforming Content Files* definition, implicitly partitioned to support page or sheet based print range parameterization From Manager.
[RunList (Marks)]	Output	Represents a structured list of marked page contents.
[Layout]	Output	Represents the layout instructions for *Imposition*

5.1.3 Imposition process

Imposition combines several pages of graphical content onto a single surface. The dimensions of the surface produced are dependent upon the physical output media, utilizing parameters set up by the *LayoutPreparation* process. Printer's marks can be added to the surface to facilitate various aspects of production.

Layout is an implicit input resource to Imposition. JDF defines the **Layout** resource structure broadly enough to encompass the needs of both fully specified and template–driven imposition. A fully described **Layout** includes an array of signatures. Each signature specifies an array of sheets, and each sheet can have up to two surfaces (front and back). The page images and any marks are placed on these surfaces using **PlacedObject** elements. A sheet that specifies no surface content will be blank. Source pages are placed onto surfaces using **ContentObject** subelements with ***Ord*** attributes; this attribute specifies an index into the document **RunList**. The **Layout** resource hierarchy explicitly specifies which pages will be imaged.

To describe automated imposition, **Layout** resources specify a single signature of sheet(s) on which page contents are imaged. The document **RunList** resource defines the sequence of pages to be imaged using automated layout. Pages are consumed in order from this **RunList** resource to satisfy the **ContentObject** subelements in the surfaces constructed by the layout signature. The signature is repeated until all pages of the **RunList** resource are consumed. Each time the signature is repeated, pages are consumed in chunks; the size of each chunk is determined by the value of ***MaxOrd*** + 1 (if present in the **Layout** resource), by the largest ***Ord*** value, or from the calculated ***OrdExpression*** value for any **ContentObject** sub-element in the signature (if ***MaxOrd*** is absent).

Media attributes are given for each sheet used in printing. Because the same signature is repeated until all pages are consumed, the **Layout** resource hierarchy can provide hints or preferences about special needs for sets of page content via **InsertSheet** elements. Inserting media is a way to separate sections of the document content. Alternate content is printed only as necessary to fill areas that need page content because new media has been added or to begin a document section as specified by the odd or even position of the signature.

The following identifies the explicit and [implict] resources for Imposition:

Name or Value	I/O	Description
RunList (Document)	Input	Same input used by *LayoutPreparation*. From Manager.
[RunList (Marks)**]**	Input	From *LayoutPreparation.*
[Layout]	Input	From *LayoutPreparation.*
[RunList (Sheet Surfaces)**]**	Output	Represents a structured list of the resulting imposed sheet surfaces

5.1.4 ColorSpaceConversion process

ColorSpaceConversion converts all colors used in the job to a known color space.

The NexPress front end supports the use and application of ICC color profiles. A separate input ICC profile can be associated with each PDF graphical content data type (including vector graphics and sampled image data) in combination with the input color spaces of RGB, CMYK, and CIE Based.

Also supported is the ability to specify a simulation color transform. This mechanism is used to emulate reference printing conditions or other color printing devices. For example, this feature is useful for proofing color content data targeted at a specific color-printing device, or at a standard printing condition such as CGATS TR001 (SWOP).

The NexPress front end stores several ICC output profiles for each supported print medium. Among these profiles are versions for available halftone screening methods in combination with each color process model. Separate profiles are also defined for protective and gloss clear coating with process CMYK imaged content.

Selection of the output ICC profile is automatic based on the process color model and the media selected by the process node. It is not explicitly controlled by the parameterization of the *ColorSpaceConversion* process.

The following identifies the explicit and [implict] resources for *ColorSpaceConversion*:

Name or Value	I/O	Description
ColorConversionParams	Input	Parameters that define how color spaces will be converted in the file. From Manager.
ColorantControl	Input	Identifies the assumed color model for the job. If not present, the default color model is CMYK. From Manager.
[RunList (Sheet Surfaces)**]**	Input	From *Imposition.*
[RunList (CM Sheet Surfaces)**]**	Output	Represents a structured list of the color managed, imposed sheet surfaces

5.1.5 Interpreting process

The *Interpreting* process parses graphical content in the page descriptions to produce a canonical display list of the elements to be drawn on each page. The *Interpreting* process is executed by Raster Image Processors (RIPs) within the NexPress front end. The RIPs are capable of interpreting *NexPress Conforming Content* data.

The following identifies the explicit and [implict] resources for *Interpreting*:

Name or Value	I/O	Description
ColorantControl	Input	Identifies the color model used by the job. From Manager.
InterpretingParams	Input	Provides the parameters needed to interpret PDL pages specified by *RunList*. From Manager.
FontPolicy	Input	Describes the behavior of the font machinery in the absence of requested fonts. From Manager.
[RunList (CM Sheet Surfaces)**]**	Input	From *ColorSpaceConversion* process.
[RunList (Interpreted Sheet Surfaces)**]**	Output	Represents a structured list of the interpreted, imposed sheet surfaces.

5.1.6 Rendering process

The *Rendering* process consumes the display list of graphical elements generated by an interpreter. It color manages and scans/converts the graphical elements according to the geometric and graphic state information contained within the display list.

The following identifies the explicit and [implict] resources for *Rendering*:

Name or Value	I/O	Description

Name or Value	I/O	Description
RenderingParams	Input	Parameters that refine the rendering of image data into color planes or subsequent processing by the engine. Many RenderingParams elements are engine-specific and not supported by all NexPress products. From Manager.
[RunList (Interpreted Sheet Surfaces)**]**	Input	From *Interpreting* process.
[RunList (ByteMaps)**]**	Output	Represents a structured list of the rasterized ByteMaps for imposed sheet surfaces.

5.1.7 Screening process

This *Screening* process consumes rasterized ByteMaps, producing rasterized and screened output data.

The following identifies the explicit and [implict] resources for *Screening*:

Name or Value	I/O	Description
ScreeningParams	Input	Parameters specifying which halftone screening method to use in the imaging process. From Manager.
[RunList (ByteMaps)**]**	Input	From *Rendering* process.
[RunList (Screened ByteMaps)**]**	Output	Represents a structured list of the screened, rasterized ByteMaps for imposed sheet surfaces.

5.1.8 DigitalPrinting process

DigitalPrinting is a direct printing process that occurs after prepress processes and before postpress processes. During *DigitalPrinting* the toner is directly transferred onto a substrate.

DigitalPrinting is the final sub-process of the NexPress Combined Process Node. Media are selected, and **RunList** (Document) content pages are mapped to physical media loaded in the NexPress digital production color press. Selection is performed using a RunTags partitioned **DigitalPrintingParams** resource in combination with annotations present in the *NexPress Conforming Content File*. **DigitalPrintingParams** also identifies the output tray and specifies sheet ordering properties of the output stack (Output Destination, Print Order, Collation, and Jog).

The properties of the output **Component** resources must be matched to the intended post-press processing workflow. An output **Component** resource characterizes the physical product that is input to a post-process finishing device in a JDF process network.

The following identifies the explicit and [implict] resources for *DigitalPrinting*:

Name or Value	I/O	Description
ColorantControl	Input	Defines the ordering and usage of inks in print modules.. From Manager.
DigitalPrintingParams	Input	Parameters specifying the printing device set up. From Manager.
Media	Input	Up to three different types of substrates are allowed for a single job in the NexPress digital production color press. One or more instances of this resource MUST be linked as an input.
[RunList (Screened ByteMaps)**]**	Input	From *Screening* process.
Component	Output	A physical resource that represents the stack of printed sheets produced.

5.2 JDF Process Node

The NexPress will only process JDF that contains a NexPress Combined Process Node. In addition, the following requirements must also be satisfied:

- There MUST only be a single NexPress Combined Process node in the JDF job ticket.
- The NexPress Process Node can be anywhere in the submitted JDF ticket, any JDF Product nodes are ignored by the NexPress during processing.
- For submissions using the NexPress Portal Interface, any NexPress Combined Process Node embedded within a PDL will be ignored. Embedded JDF Process nodes are supported for submissions that use the Virtual Printer Hot Folder interface.
- If Device resource is specified and Device/@DeviceID is defined, Device/@DeviceID must have either the value "2100" or match the JMF/@SenderID returned by the JDF Portal in its JMF response.

In the tables that follow, "Manager" refers to the Client Application creating the JDF Process; "Worker" is the NexPress front end. The notation is similar to that used by the Base ICS [BICS], except a value of "r" in the "Worker" column indicates the NexPress requires the attribute, element, or value; "r?" indicates the field is supported and will be used if supplied.

Name or Value	Manager	Worker	Description
DescriptiveName	**w?**	**r?**	Data Type: *string*
ID	**w**	**r**	Data Type: *ID*
JobID	**w?**	**r?**	Data Type: *string*
JobPartID	**w?**	**r?**	Data Type: *string*
SettingsPolicy	**w?**	**r?**	Data Type: *enumeration*
BestEffort	**w**	**r**	NexPress applies a policy of BestEffort
Status	**w**	**r**	Data Type: *enumeration*
Ready	**w?**	**r?**	
Waiting	**w?**	**r?**	
Type	**w**	**r**	Data Type: *NMTOKEN*
Combined	**w**	**r**	
Types	**w**	**r**	Data Type: *NMTOKENS*
LayoutPreparation Imposition ColorSpaceConversion Interpreting Rendering Screening DigitalPrinting Gathering HoleMaking Stitching	**w**	**r**	This set of types identifies the NexPress Combined Process Node. Any subset of listed types is permitted that contains "DigitalPrinting".
Version	**w**	**r?**	Data Type: *JDFJMFVersion*
1.3	**w**	**r**	
xmlns	**w**	**r?**	Data Type: *URI*
http://www.CIP4.org/JDFSchema_1_1	**w**	**r**	
xmlns:HDM	**w**	**r?**	Data Type: *URI*
http://www.heidelberg.com/schema/HDM	**w**	**r**	
xmlns:NXP	**w**	**r?**	Data Type: *string*
www.nexpress.com	**w**	**r**	
xmlns:xsi	**w**	**r?**	Data Type: *URI*

Name or Value	Manager	Worker	Description
http://www.w3.org/2001/XMLSchema-instance	**w**	**r**	
xsi:schemaLocation	**w**	**r?**	Data Type: *URI*
http://www.CIP4.org/JDFSchema_1_1	**w**	**r**	
xsi:type	**w?**	**!r**	Data Type: *NMTOKEN* The NexPress Portal does not do schema validation
Combined	**w**	**r**	
Activation = Active	**w?**	**r?**	Describes the activation status of the JDF node.
Active	**w?**	**r?**	The submitted JDF ticket is ready for processing.
Held	**w?**	**r?**	The submitted JDF ticket should enter the Portal Queue with an initial state of Held. ResumeQueueEntry is required before the job will begin processing.
AuditPool	**w?**	**r?**	See **AuditPool** below.
NodeInfo	**w?**	**r?**	Although deprecated in JDF 1.3, the NexPress supports use of this element to designate a TargetRoute. Alternatively, create it as a member of the ResourcePool. See **NodeInfo** resource within **ResourcePool** below.
ResourceLinkPool	**w**	**r**	See **ResourceLinkPool** below.
ResourcePool	**w**	**r**	See **ResourcePool** below.

5.2.1 AuditPool

Name or Value	Manager	Worker	Description
Created	**w?**	**r?**	See **Created** below.
Modified	**w?**	**r?**	See **Modified** below.
Notification	**r?**	**w**	Written as JDF ticket is processed. See **Notification** below.
ProcessRun	**r?**	**w**	Written upon completion of JDF ticket. See **ProcessRun** below.

5.2.1.1 Created

Name or Value	Manager	Worker	Description
AgentName	**w?**	**r?**	Data Type: *string*
AgentVersion	**w?**	**r?**	Data Type: *string*
Author	**w?**	**r?**	Data Type: *string*
ID	**w**	**r**	Data Type: *ID*
TimeStamp	**w?**	**r?**	Data Type: *dateTime*

5.2.1.2 Modified

Name or Value	Manager	Worker	Description
AgentName	**w?**	**r?**	Data Type: *string*
AgentVersion	**w?**	**r?**	Data Type: *string*
Author	**w?**	**r?**	Data Type: *string*
TimeStamp	**w?**	**r?**	Data Type: *dateTime*

5.2.1.3 Notification

Name or Value	Manager	Worker	Description
AgentName	**r?**	**w**	Data Type: *string* Kodak NexPress <SERVERNAME>
AgentVersion	**r?**	**w**	Data Type: *string*
Class	**r?**	**w**	Data Type: *enumeration*
Error	**r?**	**w?**	An error has occurred. Execution cannot continue.
Information	**r?**	**w?**	Information about a process.
Warning	**r?**	**w?**	A minor error occurred and an automatic fix was applied. Execution continues.
Comment	**r?**	**w?**	Data Type: *telem* Free-Form text
ID	**r?**	**w**	Data Type: *ID* Generated for this **Notification** node.
TimeStamp	**r?**	**w**	Data Type: *dateTime* Time of **Notification** node creation.

5.2.1.4 ProcessRun

Name or Value	Manager	Worker	Description
AgentName	**r?**	**w**	Data Type: *string* Kodak NexPress <SERVERNAME>
AgentVersion	**r?**	**w**	Data Type: *string*
Author	**r?**	**w**	Data Type: *string* NXP2100@<SERVERNAME>
Comment	**r?**	**w?**	Data Type: *telem* Free-Form text
End	**r?**	**w**	Data Type: *dateTime* Time job completed or aborted processing
EndStatus	**r?**	**w**	Data Type: *enumeration* Final job status
Completed	**r?**	**w?**	Job completed printing
Aborted	**r?**	**w?**	Job ticket aborted due to error, or job was removed by operator using the NexPress Client application.
ID	**r?**	**w**	Data Type: *ID* Generated for this **ProcessRun** node.

Name or Value	Manager	Worker	Description
NXP:CDFEJobID	**r?**	**w**	Data Type: *integer* ID of job as referenced by Internal Job Queue.
Start	**r?**	**w**	Data Type: *dateTime* Time job added to Internal Job Queue
TimeStamp	**r?**	**w**	Data Type: *dateTime* Time of **ProcessRun** node creation.

5.2.2 ResourceLinkPool

Name or Value	Manager	Worker	Description
ResourceLink *	**w**	**r**	One or more **ResourceLink** elements. See below.
ColorantControlLink	**w?**	**r?**	
ColorSpaceConversionParamsLink	**w?**	**r?**	
ComponentLink	**w**	**r**	
DeviceLink	**w?**	**r?**	Optional. If a Device Resource is defined, the value of DeviceID must be "2100".
DigitalPrintingParamsLink	**w**	**r**	
FontPolicyLink	**w?**	**r?**	
GatheringParamsLink	**w?**	**r?**	Required for post-fuser insertion of cover media
HoleMakingParamsLink	**w?**	**r?**	Required for punching
InterpretingParamsLink	**w?**	**r?**	
LayoutPreparationParamsLink	**w**	**r**	
MediaLink	**w**	**r**	
NodeInfoLink	**w?**	**r?**	
RenderingParamsLink	**w?**	**r?**	
RunListLink	**w**	**r**	
ScreeningParamsLink	**w**	**r**	
StitchingParamsLink	**w?**	**r?**	Required for stapling

5.2.2.1 ColorantControlLink

Name or Value	Manager	Worker	Description
CombinedProcessIndex	**w**	**r**	Data Type: *IntegerList* Maps to ColorSpaceConversion
rRef	**w**	**r**	Data Type: *IDREF*
Usage	**w**	**r**	Data Type: *enumeration*
Input	**w**	**r**	

5.2.2.2 ColorSpaceConversionParamsLink

Name or Value	Manager	Worker	Description

Name or Value	Manager	Worker	Description
CombinedProcessIndex	**w**	**r**	Data Type: *IntegerList* Maps to ColorSpaceConversion
rRef	**w**	**r**	Data Type: *IDREF*
Usage	**w**	**r**	Data Type: *enumeration*
Input	**w**	**r**	

5.2.2.3 ComponentLink

Name or Value	Manager	Worker	Description
CombinedProcessIndex	**w**	**r**	Data Type: *IntegerList* Maps to Interpreting
ProcessUsage	**w?**	**!r**	Data Type: *string* Not used.
Document	**w?**	**r?**	
Proof	**w?**	**r?**	
rRef	**w**	**r**	Data Type: *IDREF*
Usage	**w**	**r**	Data Type: *enumeration*
Output	**w**	**r**	

5.2.2.4 DeviceLink

Name or Value	Manager	Worker	Description
rRef	**w**	**r**	Data Type: *IDREF*
Usage	**w**	**r**	Data Type: *enumeration*
Input	**w**	**r**	

5.2.2.5 DigitalPrintingParamsLink

Name or Value	Manager	Worker	Description
CombinedProcessIndex	**w**	**r**	Data Type: *IntegerList* Maps to DigitalPrinting
rRef	**w**	**r**	Data Type: *IDREF*
Usage	**w**	**r**	Data Type: *enumeration*
Input	**w**	**r**	

5.2.2.6 FontPolicyLink

Name or Value	Manager	Worker	Description
CombinedProcessIndex	**w**	**r**	Data Type: *IntegerList* Maps to Interpreting
rRef	**w**	**r**	Data Type: *IDREF*
Usage	**w**	**r**	Data Type: *enumeration*
Input	**w**	**r**	

5.2.2.7 GatheringParamsLink

Name or Value	Manager	Worker	Description

Name or Value	Manager	Worker	Description
CombinedProcessIndex	**w**	**r**	Data Type: *IntegerList* Maps to Gathering
rRef	**w**	**r**	Data Type: *IDREF*
Usage	**w**	**r**	Data Type: *enumeration*
Input	**w**	**r**	

5.2.2.8 HoleMakingParamsLink

Name or Value	Manager	Worker	Description
CombinedProcessIndex	**w**	**r**	Data Type: *IntegerList* Maps to HoleMaking
rRef	**w**	**r**	Data Type: *IDREF*
Usage	**w**	**r**	Data Type: *enumeration*
Input	**w**	**r**	

5.2.2.9 InterpretingParamsLink

Name or Value	Manager	Worker	Description
CombinedProcessIndex	**w**	**r**	Data Type: *IntegerList* Maps to Interpreting
rRef	**w**	**r**	Data Type: *IDREF*
Usage	**w**	**r**	Data Type: *enumeration*
Input	**w**	**r**	

5.2.2.10 LayoutPreparationParamsLink

Name or Value	Manager	Worker	Description
CombinedProcessIndex	**w**	**r**	Data Type: *IntegerList* Maps to LayoutPreparation
rRef	**w**	**r**	Data Type: *IDREF*
Usage	**w**	**r**	Data Type: *enumeration*
Input	**w**	**r**	

5.2.2.11 MediaLink

Name or Value	Manager	Worker	Description
CombinedProcessIndex	**w**	**r**	Data Type: *IntegerList* Maps to DigitalPrinting
rRef	**w**	**r**	Data Type: *IDREF*
Usage	**w**	**r**	Data Type: *enumeration*
Input	**w**	**r**	

5.2.2.12 NodeInfoLink

Name or Value	Manager	Worker	Description
CombinedProcessIndex	**w**	**r**	Data Type: *IntegerList* Maps to DigitalPrinting
rRef	**w**	**r**	Data Type: *IDREF*

Name or Value	Manager	Worker	Description
Usage	**w**	**r**	Data Type: *enumeration*
Input	**w**	**r**	

5.2.2.13 RenderingParamsLink

Name or Value	Manager	Worker	Description
CombinedProcessIndex	**w**	**r**	Data Type: *IntegerList* Maps to Rendering
rRef	**w**	**r**	Data Type: *IDREF*
Usage	**w**	**r**	Data Type: *enumeration*
Input	**w**	**r**	

5.2.2.14 RunListLink

Name or Value	Manager	Worker	Description
CombinedProcessIndex	**w**	**r**	Data Type: *IntegerList* Maps to LayoutPreparation and Imposition
ProcessUsage	**w?**	**r?**	Data Type: *string*
Document	**w?**	**r?**	
rRef	**w**	**r**	Data Type: *IDREF*
Usage	**w**	**r**	Data Type: *enumeration*
Input	**w**	**r**	
Part *	**w?**	**r?**	References **RunList** (Documents) as input to the *LayoutPreparation* and *Imposition* processes. For each print range, a separate sub-element MUST be used.

5.2.2.14.1 Part

Both ***RunIndex*** and ***SheetIndex*** attributes select a set of pages from the **RunList** resource. Either contains an array of mixed ranges and individual indices separated by whitespace. Each range consists of two indices connected by a tilde (~). Negative numbers reference from the back of a file; each IntegerRange value of m ~ n MUST satisfy the condition: $m \geq 0$ and $m \leq n$. Additionally, for two successive IntegerRanges the value of the first n must be equal or less than the second value (monotonically increasing sequence ranges). If m ~ n is the last IntegerRange in the list, then n can have the value "–1" to represent the last element of the sequence. As logical indices, **RunIndex** and **SheetIndex** cannot be used as partition keys of RunList resources. See [JDF] for details and examples.

Name or Value	Manager	Worker	Description
RunIndex	**w?**	**r?**	Data Type: *IntegerRangeList* Only one of the attributes ***RunIndex*** or ***SheetIndex*** is allowed. Identifies by explicit index the pages from **RunList** (Document) to be printed.

Name or Value	Manager	Worker	Description
SheetIndex	**w?**	**r?**	Data Type: *IntegerRangeList* Only one of the attributes ***RunIndex*** or ***SheetIndex*** is allowed. Identifies the set of logical sheets to be printed. The NexPress front end determines which document pages are selected for printing using the specified ***SheetIndex***. In 1-up simplex printing, ***SheetIndex*** and ***RunIndex*** are identical.

5.2.2.15 ScreeningParamsLink

Name or Value	Manager	Worker	Description
CombinedProcessIndex	**w**	**r**	Data Type: *IntegerList* Maps to Screening
rRef	**w**	**r**	Data Type: *IDREF*
Usage	**w**	**r**	Data Type: *enumeration*
Input	**w**	**r**	

5.2.2.16 StitchingParamsLink

Name or Value	Manager	Worker	Description
CombinedProcessIndex	**w**	**r**	Data Type: *IntegerList* Maps to Stitching
rRef	**w**	**r**	Data Type: *IDREF*
Usage	**w**	**r**	Data Type: *enumeration*
Input	**w**	**r**	

5.2.3 ResourcePool

Name or Value	Manager	Worker	Description
Resource *	**w**	**r**	One or more **Resource** elements. See below.
ColorantControl	**w?**	**r?**	Specifies the process-color printing model to be used. See **ColorantControl** below.
ColorSpaceConversionParams	**w?**	**r?**	Specifies the color conversion to be performed for each type of color space found in the PDL. See **ColorSpaceConversionParams** below.
Component	**w**	**r**	Describes finished product including print quantity and dimensions. See **Component** below.
Device	**w?**	**r?**	Optional. If a Device Resource is defined, the value of DeviceID must be "2100". See **Component** below.

Name or Value	Manager	Worker	Description
DigitalPrintingParams	**w**	**r**	Specifies the printing parameters including output location, media/tag mapping, and page order. See **DigitalPrintingParams** below.
FitPolicy	**w?**	**r?**	Specifies the policy for the NexPress front end when fitting content to specific media. See **FitPolicy** below.
FontPolicy	**w?**	**r?**	Specifies the policy for the NexPress front end when font errors occur in PDL files. See **FontPolicy** below.
InterpretingParams	**w?**	**r?**	Specifies the manner in which the NexPress front end interprets the PDL pages. See **InterpretingParams** below.
LayoutPreparationParams	**w?**	**r?**	Provides the details of how page contents will be imaged onto media. See **LayoutPreparationParams** below.
Media	**w**	**r**	Describes the physical media and clearcoat selections. See **Media** below.
NodeInfo	**w?**	**r?**	The NexPress supports use of this element to designate a TargetRoute. See **NodeInfo** below.
RenderingParams	**w?**	**r?**	Identifies how the Rendering process should operate.
RunList	**w**	**r**	Identifies the source content file. See **RunList** below.
ScreeningParams	**w?**	**r?**	Identifies the screening method to use for processing. See **ScreeningParams** below.

5.3 ColorantControl resource

Name or Value	Manager	Worker	Description
Class	**w**	**r**	Data Type: *enumeration*
Parameter	**w**	**r**	
ID	**w**	**r**	Data Type: *ID*
ProcessColorModel = DeviceCMYK	**w←**	**r?**	Data Type: *NMTOKEN* If not present or value is invalid, the default value is used and **ColorantParams** is ignored. Only specific values of **ColorantParams** are allowed. See table below for valid combinations with **ProcessColorModel**.
DeviceCMYK	**w**	**r**	Default Value.
DeviceN	**w**	**r**	
Status	**w**	**r**	Data Type: *enumeration*
Available	**w**	**r**	

Name or Value	Manager	Worker	Description
NXP:HotOffsetCompensation	**w?**	**r?**	Specifies the level of HotOffset compensation required with Dimensional Easy Overcoat applications.
None			Default Value - No hot offset compensation
Low			low hot offset compensation
Medium			medium hot offset compensation
High			high hot offset compensation
ColorantParams	**w←**	**r?**	See **ColorantParams** and table below.

Specific combinations of ***ProcessColorModel*** and **ColorantParams** allowed.

ProcessColorModel	**ColorantParams**	**NexPress Color Mode**
Not present	N/A	Four Color
DeviceCMYK	Not present or empty	Four Color
DeviceCMYK	Present and containing exactly one **SeparationSpec** sub-element with a ***Name*** value of one of the clear dry ink names: NexPress DryInk clear, NexPress DryInk xd clear, NexPress DryInk uv clear, NexPress DryInk matte clear; or the name of one non-gamut ink: NexPress DryInk micr, NexPress DryInk light black spot, NexPress DryInk light black photo,NexPress DryInk custom color, NexPress DryInk white, NexPress DryInk gold, NexPress DryInk pearlescent.	Four Color + Clear DryInk
DeviceN	Present and containing exactly 5 **SeparationSpec** sub-elements (one **SeparationSpec** sub-element for each of the four CMYK process colorants and one designating the NexPress gamut expansion DryInk colorant). ***Name*** values of *Cyan*, *Magenta*, *Yellow* and *Black, NexPress DryInk blue, NexPress DryInk green, NexPress DryInk red, NexPress DryInk raised clear, NexPress DryInk neon pink*).	Five Color

Deleted: NexPress DryInk raised clear,

Deleted: or

Deleted: m

5.3.1 ColorantParams

Name or Value	Manager	Worker	Description
SeparationSpec *	**w?**	**r?**	Identifies the names of required colorants. Refer to the above table for usage; see **SeparationSpec** below for details.

5.3.1.1 SeparationSpec

Name or Value	Manager	Worker	Description
Name	**w?**	**r?**	Data Type: *string*
Cyan	**w?**	**r?**	Process Color Cyan
Magenta	**w?**	**r?**	Process Color Magenta
Yellow	**w?**	**r?**	Process Color Yellow
Black	**w?**	**r?**	Process Color Black

Name or Value	Manager	Worker	Description
NexPress DryInk clear	**w?**	**r?**	NexPress ClearCoat and GlossCoat ink. Must be specified whenever ClearCoat is selected.
NexPress DryInk red	**w?**	**r?**	NexPress Gamut Expansion DryInk Red. Specify exactly one Gamut Expansion DryInk when ***ProcessColorModel=DeviceN***
NexPress DryInk green	**w?**	**r?**	NexPress Gamut Expansion DryInk Green. Specify exactly one Gamut Expansion DryInk when ***ProcessColorModel=DeviceN***
NexPress DryInk blue	**w?**	**r?**	NexPress Gamut Expansion DryInk Blue. Specify exactly one Gamut Expansion DryInk when ***ProcessColorModel=DeviceN***
NexPress DryInk micr	**w?**	**r?**	NexPress MICR secure printing ink.
NexPress DryInk raised clear	**w?**	**r?**	NexPress Raised Clear ink.
NexPress DryInk xd clear	**w?**	**r?**	NexPress XD Clear ink.
NexPress DryInk uv clear	**w?**	**r?**	NexPress Red Fluorescing Clear ink
NexPress DryInk light black spot	**w?**	**r?**	NexPress Light Black ink
NexPress DryInk light black photo	**w?**	**r?**	NexPress Light Black ink
NexPress DryInk pearlescent	**w?**	**r?**	NexPress Pearlescent ink
NexPress DryInk matte clear	**w?**	**r?**	NexPress Matte Clear ink
NexPress DryInk custom color	**w?**	**r?**	NexPress Custom Color
NexPress DryInk gold	**w?**	**r?**	NexPress Gold ink
NexPress DryInk white	**w?**	**r?**	NexPress White ink
NexPress DryInk neon pink	**w?**	**r?**	NexPress Gamut Expansion DryInk Neon Pink. Specify exactly one Gamut Expansion DryInk when ***ProcessColorModel=DeviceN***

5.4 ColorSpaceConversionParams resource

Name or Value	Manager	Worker	Description
Class	**w**	**r**	Data Type: *enumeration*
Parameter	**w**	**r**	
ID	**w**	**r**	Data Type: *ID*
NXP:ColorMapping = true	**w?**	**r?**	Data Type: *boolean* Specifies the method of spot color to process color translation to be used.
false	**w?**	**r?**	Spot colors are mapped using the color equivalents specified in the content file.
true	**w?**	**r?**	Spot colors are mapped using spot color to process color conversion tables available in the NexPress front end. Default Value.

Name or Value	Manager	Worker	Description
Status	**w**	**r**	Data Type: *enumeration*
Available	**w**	**r**	
ColorSpaceConversionOp *	**w?**	**r?**	Up to five ColorSpaceConversionOp elements may be present to differentiate the following types of graphical objects: *CMYK Image* : Match *DeviceCMYK* Images *CMYK Graphics*: Match *DeviceCMYK* Graphics *RGB Image* : Match *DeviceRGB* Images *RGB Graphics* : Match *DeviceRGB* Graphics *CIE Based* : Device Independent *CIEBased* Color data See **ColorSpaceConversionOp** below for more detail..
FileSpec	**w?**	**r?**	Data Type: *refelement* Used to select an output ICC profile installed as a resource on the NexPress front end used for output color rendering simulation. If not present or **FileSpec** is not valid, the identified ICC profile is not applied. See below. A **FileSpec** element is sometimes required even when no ICC profile is required (as with "untag" operations for CIEBased ColorSpaceCovnersions).

5.4.1 ColorSpaceConversionOp

Name or Value	Manager	Worker	Description
IgnoreEmbeddedICC* = *false	**w?**	**r?**	Data Type: *boolean* Selective ignoring of embedded ICC profiles with respect to the source object types and source color space combinations is not supported by the NexPress front end. However IgnoreEmbeddedICC can be used to globally ignore embedded ICC profiles for all supported objects and color spaces available.
false	**w?**	**r?**	Default Value.
true	**w?**	**r?**	Ignore all embedded ICC profiles. Also requires ***ColorSpaceConversionOp/@SourceObjects*** MUST have the value "*All*" , and ***ColorSpaceConversionOp/@SourceCS*** MUST have the value "*DevIndep*".
Operation	**w**	**r**	Data Type: *enumeration* Controls behavior of color space conversion utility
Convert	**w?**	**r?**	Transforms graphical elements to final target color space.

Name or Value	Manager	Worker	Description
Untag	**w?**	**r?**	Removes all profiles and color characterizations from graphical elements. An operation of "Untag" is required to disable ColorConversion operations that are defined by the default job ticket.
PreserveBlack = false	**w?**	**r?**	Data Type: *boolean* Controls how the tints of black should be handled. Only applicable to CMYK Image and CMYK Graphics. There are interactions between ***SourceCS***, ***SourceObjects***, ***PreserveBlack*** and **RGBGray2Black**. Refer to the table below for allowable values of these attributes.
false	**w?**	**r?**	Tints of black are handled through the standard ICC workflow. Default Value.
true	**w?**	**r?**	Tints of black are converted into other shades of black. Algorithm is implementation-specific.
RenderingIntent	**w←**	**r?**	Data Type: *enumeration* Specifies the output rendering intent for print process simulation in a CIE-Based **ColorSpaceConversionOp** element. Requires the explicit definition of **ColorSpaceConversionParams/FileSpec** and its attributes ***@ResourceUsage***="ReferenceOutputProfile" and ***@UserFileName***. ***RenderingIntent*** is ignored if ***FileSpec/ColorSpaceConversionParams/FileSpec/@ResourceUsage*** = "ActualOutputProfile". ***RenderingIntent*** MUST be defined if FileSpec is fully specified. When **RenderingIntent** is defined, ***SourceCS*** = "*DevIndep*".
RelativeColorimetric	**w?**	**r?**	
AbsoluteColorimetric	**w?**	**r?**	
RGBGray2Black = false	**w?**	**r?**	Data Type: *boolean* Controls mapping of gray values (R = G = B) from CMY to Black ink. Only applicable for RGB Graphics elements (SourceCS = RGB; SourceObjects of type Text, LineArt, or SmoothShades). Cannot be enabled when **RenderingParams/@NXP:RichBlack** = true; there are also interactions between ***SourceCS***, ***SourceObjects***, ***PreserveBlack*** and **RGBGray2Black**. Refer to the table below for allowable values of these attributes.
false	**w?**	**r?**	Feature is disabled. Default Value.
true	**w?**	**r?**	Values below the RGBGray2BlackThreshold are replaced by Black equivalents.
RGBGray2BlackThreshold = 1	**w?**	**r?**	Data Type: *double* A value between 0.0 and 1.0 specifying the threshold value above which gray (R = G = B) IS NOT converted to Black. The threshold is only applicable when RGBGray2Black is true. A "0" value will convert only R = G = B = 0 to Black; A "1" value will convert all R = G = B values to Black.

Name or Value	Manager	Worker	Description
SourceRenderingIntent = Perceptual	**w?**	**r?**	Data Type: *enumeration* Specifies rendering intent for the source profile of this **ColorSpaceConversionOp**.
Saturation	**w?**	**r?**	
Perceptual	**w?**	**r?**	Default Value.
RelativeColorimetric	**w?**	**r?**	
AbsoluteColorimetric	**w?**	**r?**	
SourceCS	**w**	**r**	Data Type: *enumeration* Identifies which of the incoming color spaces are operated on by this **ColorSpaceConversionOp**. There are interactions between ***SourceCS***, ***SourceObjects***, ***PreserveBlack,*** and ***RGBGray2Black***. Refer to the table below for allowable values of these attributes.
CMYK	**w?**	**r?**	
DevIndep	**w?**	**r?**	
RGB	**w?**	**r?**	*In instances when **SourceCS=**RGB, **FileSpec@ResourceUsage** shall be force to be SourceProfile. This prevents color management from being disabled for RGB colorspaces.*
SourceObjects = All	**w←**	**r?**	Data Type: *enumerations* List of object classes that identifies which incoming graphical objects are operated on. List all applicable values. There are interactions between ***SourceCS***, ***SourceObjects***, ***PreserveBlack,*** and ***RGBGray2Black***. Refer to the table below for allowable values of these attributes.
All	**w?**	**r?**	Default Value.
ImagePhotographic	**w?**	**r?**	Contone images
ImageScreenShot	**w?**	**r?**	Images largely comprised of rasterized vector art
Text	**w?**	**r?**	Text
LineArt	**w?**	**r?**	Line Art
SmoothShades	**w?**	**r?**	Gradients and blends
FileSpec	**w?**	**r?**	Data Type: *refelement* Reference to a FileSpec resource for an ICC Profile that describes the assumed color space. If not present, embedded profiles are used. ***FileSpec@ResourceUsage*** may be *SourceProfile.*

Allowable combinations of ***SourceCS***, ***SourceObjects***, ***PreserveBlack***, and ***RGBGray2Black*** for **ColorSpaceConversionOp**.

Object Type	***SourceCS***	***SourceObjects***	***PreserveBlack***	***RGBGray2Black***
CMYK Images	*CMYK*	*ImagePhotographic* *ImageScreenShot*	*True* or *False*	*N/A*

Object Type	*SourceCS*	*SourceObjects*	*PreserveBlack*	*RGBGray2Black*
CMYK Graphics	*CMYK*	*Text LineArt SmoothShades*	*True* or *False*	*N/A*
RGB Images	*RGB*	*ImagePhotographic ImageScreenShot*	*N/A*	*N/A*
RGB Graphics	*RGB*	*Text LineArt SmoothShades*	*N/A*	*True* or *False*
CIE Based	*DevIndep*	*All*	*N/A*	*N/A*

5.4.2 FileSpec

Name or Value	Manager	Worker	Description
ResourceUsage	**w**	**r**	Data Type: *NMTOKEN* In instances where ***ColorSpaceConversionOp/@SourceCS*** = *RGB*, ***ResourceUsage*** will be forced to *SourceProfile*. This prevents disabling color management for RGB color spaces.
ActualOutputProfile	**w?**	**r?**	
ReferenceOutputProfile	**w?**	**r?**	Indicates that the referenced output profile is used for color rendering simulation (i.e. proof emulation).
SourceProfile	**w?**	**r?**	Indicates that the referenced file is a source ICC profile.
UserFileName	**w**	**r**	Data Type: *string* Identifies the name of the ICC profile resource as it is known to the NexPress front end. The names of the ICC profiles installed on the NexPress front end can be obtained through the NexPress front end's device capabilities file (refer to [KNDIR]).

5.4.3 Allowable combinations of *ColorSpaceConversion/FileSpec* and *ColorSpaceConversionOp/@RenderingIntent*

ColorSpaceConversionParams/FileSpec		ColorSpaceConversionOp	
/@ResourceUsage	/@UserFileName	/@RenderingIntent (SourceCS_DevIndep)	Expected Result*
ReferenceOutputProfile	Provided	RelativeColorimetric	Print
ReferenceOutputProfile	Provided	AbsoluteColorimetric	Print
ReferenceOutputProfile	Provided	Not Provided	Error
ReferenceOutputProfile	Not Provided	RelativeColorimetric	Error
ReferenceOutputProfile	Not Provided	Not Provided	Error
ActualOutputProfile	Provided	Not Provided	Print
ActualOutputProfile	Not Provided	Not Provided	Print

ActualOutputProfile	Not Provided	RelativeColorimetric	Print
ActualOutputProfile	Provided	RelativeColorimetric	Print
Not Provided	Provided	RelativeColorimetric	Error
Not Provided	Provided	Not Provided	Print
FileSpec Not Provided		RelativeColorimetric	Error
FileSpec Not Provided		Not Provided	Print

5.5 Device resource

Name or Value	Manager	Worker	Description
Class	**w**	**r**	Data Type: *enumeration*
Implementation	**w**	**r**	
DeviceID	**w?**	**r?**	Data Type: *string* Device identifier. Restricts processing of the JDF node to only the specified device. Optional, but if supplied, ***DeviceID*** must match **JMF/@SenderID** reported by the device in JMF responses. ***DeviceID*** is NOT case sensitive.
<SERVERNAME>	**w?**	**r?**	Use the same string value returned in ***JMF/@SenderID***.
2100	**w?**	**r?**	*DEPRECATED* Not JDF1.3 compliant. Supported for compatibility with past NexPress releases.
DeviceType	**w?**	**!r**	Data Type: *string* Not used.
FriendlyName	**w?**	**!r**	Data Type: *string* Not used.
ID	**w**	**r**	Data Type: *ID*
ModelName	**w?**	**r?**	Data Type: *string* Identifies intended printer model for JDF ticket. See ***Device/@ModelName*** returned in JMF Status query. **This tag is not currently used by NexPress for job acceptance or processing.**
NexPress_Classic	**w?**	**r?**	NexPress Classic Color Press
NexPress_NPP	**w?**	**r?**	NexPress New Paper Platform Color Press
NexPress_M700	**w?**	**r?**	NexPress Model M700 Color Press
Unknown	**w?**	**r?**	Unknown Press
Status	**w?**	**r?**	Data Type: *enumeration* ***Component/@Status*** is not used.
Available	**w?**	**r?**	
Unavailable	**w?**	**r?**	

5.6 Component resource

Name or Value	Manager	Worker	Description
Amount	**w**	**r?**	Data Type: *integer* Number of copies to produce. NOTE: The use of this attribute by the NexPress front end is different from its semantics as defined in [JDF].
Class	**w**	**r**	Data Type: *enumeration*
Quantity	**w**	**r**	
ComponentType	**w**	**r**	Data Type: *enumerations* Category of the component.
PartialProduct	**w**	**r**	The NexPress only supports a component type of PartialProduct
Dimensions = 0 0 0	**w?**	**r?**	Data Type: *shape* Output component resource is linked as an input resource to the succeeding postpress process. Default of *"0 0 0"* specifies unknown. In this case a portrait orientation (Y > X) is assumed.
ID	**w**	**r**	Data Type: *ID*
Status	**w?**	**r?**	Data Type: *enumeration* ***Component/@Status*** is not used.
Available	**w?**	**r?**	
Unavailable	**w?**	**r?**	

5.7 DigitalPrintingParams resource

A **DigitalPrintingParams** resource MUST be specified in one of the following two syntax variants:

- As a single non-partitioned **DigitalPrintingParams** resource
- As a partitioned **DigitalPrintingParams** resource where it shall be partitioned by ***SheetIndex*** and/or ***RunTags***. This syntax variant should be used when specifying the use of multiple mediums.

Name or Value	Manager	Worker	Description
Class	**w**	**r**	Data Type: *enumeration*
Parameter	**w**	**r**	
ID	**w**	**r**	Data Type: *ID*
Collate	**w?**	**r?**	Data Type: *enumeration* Determines sequencing of sheets in the document. If not specified or has a non-supported value, the NexPress uses the collation setting in the system default hot folder.
None	**w?**	**r?**	Do not collate sheets in the document or documents comprising the job.
SheetSetAndJob	**w?**	**r?**	Collate sheets in the document, documents in the set, and sets in the job.

Name or Value	Manager	Worker	Description
Disjointing	**w?**	**r?**	Specifies offset and stacking settings. See **Disjointing** below.
OutputBin = SystemSpecified	**w?**	**r?**	Data Type: *NMTOKEN* Specifies the bin to which the finished document should be output.
Booklet	**w?**	**r?**	Inline Finisher
LargeCapacity	**w?**	**r?**	Main Delivery
LargeCapacity-2	**w?**	**r?**	Main Delivery 2
SystemSpecified	**w?**	**r?**	The destination uses the system default. Default Value.
Top	**w?**	**r?**	Proof Delivery
Tray-1	**w?**	**r?**	M700e FinisherUpper tray
Tray-2	**w?**	**r?**	M700e FinisherLower tray
PageDelivery = SystemSpecified	**w?**	**r?**	Data Type: *enumeration* Specifies how pages are to be delivered to the output bin or finisher.
SameOrderFaceUp	**w?**	**r?**	Order as defined by the RunList, with "front" sides of the media up.
SameOrderFaceDown	**w?**	**r?**	Order as defined by the RunList, with "front" sides of the media down.
ReverseOrderFaceUp	**w?**	**r?**	Order reversed, as defined by the RunList, with "front" sides of the media up.
ReverseOrderFaceDown	**w?**	**r?**	Order reversed, as defined by the RunList, with "front" sides of the media down.
SystemSpecified	**w?**	**r?**	Delivery order uses the system default. Default Value.
PartIDKeys	**w?**	**r?**	Data Type: *enumerations* If not present, **DigitalPrintingParams** is a single unpartioned resource. Use when specifying the use of multiple mediums.
SheetIndex	**w←**	**r?**	Partition using the NexPress cover-mode method of page-to-media mapping.
RunTags	**w←**	**r?**	Partition using the NexPress label method of page-to-media mapping.
SheetIndex RunTags	**w←**	**r?**	Partition using both NexPress cover-mode and NexPress label methods of page-to-media mapping. (NOTE: You must specify the value of PartIDKeys in this order -- SheetIndex, then RunTags -- when they are used together.)
Status	**w**	**r**	Data Type: *enumeration*
Available	**w**	**r**	

Name or Value	Manager	Worker	Description
NXP:ColorFlow	**w?**	**r?**	Data Type: *boolean* When enabled, job must adhere to ColorFlow policy of the press. This may prevent the job from printing if the press requires color calibration.

5.7.1 DigitalPrintingParams sub-element

Name or Value	Manager	Worker	Description
RunTags	**w?**	**r?**	Data Type: *NMTOKENS* Supports the NexPress label methods (stamp annotation and PPML/VDX labels) for PDL page to media mapping. Creates a partition consisting of sheets tagged with RunTags values. To support stamp annotation, RunTags values are limited to "SubstrateTypeCover" and "SubstrateTypeInsert". To support PPML/VDX labels, values must correspond to the arbitrary values of the PPML DOCUMENT/@Label attributes. A RunTags value is ignored if it does not correspond to a PDF stamp annotation or PPML DOCUMENT/@Label.
SubstrateTypeCover	**w?**	**r?**	Defines a partition for substrates tagged with the stamp annotation: SubstrateTypeCover.
SubstrateTypeInsert	**w?**	**r?**	Defines a partition for substrates tagged with the stamp annotation: SubstrateTypeInsert.
<arbitrary DOCUMENT/@Label >	**w?**	**r?**	Defines a partition for substrates in a PPML Document that are assigned the arbitrary label.
SheetIndex	**w?**	**r?**	Data Type: *IntegerRangeList* Creates a partition that maps sheets to **MediaRef**
0 (used without ***RunTags*** key)	**w?**	**r?**	Defines a partition consisting of the first sheet.
-1 (used without ***RunTags*** key)	**w?**	**r?**	Defines a partition consisting of the last sheet.
0 -1 (used without ***RunTags*** key)	**w?**	**r?**	Defines a partition consisting of the first and last sheets.
1 ~ -1 (used with ***RunTags*** key)	**w?**	**r?**	Defines a partition consisting of all but the first sheet.
0 ~ -2 (used with ***RunTags*** key)	**w?**	**r?**	Defines a partition consisting of all but the last sheet.
1 ~ -2 (used with ***RunTags*** key)	**w?**	**r?**	Defines a partition consisting of all but the first and last sheets.

Name or Value	Manager	Worker	Description
MediaRef	**w←**	**r?**	Data Type: *refelement* The **Media** resource can be partitioned to designate mixed media usage within the job. Alternatively, ***MediaRef*** can be added to a partitioned **DigitalPrintingParams**. ***MediaRef*** is not used if the **Media** resource is itself partitioned by ***SheetIndex*** and/or ***RunTags***.

5.7.1.1 MediaRef

Name or Value	Manager	Worker	Description
rRef	**w**	**r**	

5.7.2 Disjointing

Name or Value	Manager	Worker	Description
OffsetAmount = 1	**w?**	**r?**	Data Type: *integer* Offset Set Count. Default = 1.
OffsetDirection = None	**w?**	**r?**	Data Type: *enumeration* Control offset stacking of printed output.
None	**w?**	**r?**	Offset Stacking Disabled. Default Value.
Alternate	**w?**	**r?**	Offset Stacking Enabled. Stack is alternated after producing each OffsetAmount number of sets.
NXP:SeparatorAmount	**w?**	**r?**	Data Type: *integer* When specified, a separator sheet will be inserted between sets consisting of SeparatorAmount components.

5.8 FitPolicy resource

Name or Value	Manager	Worker	Description
Class	**w**	**r**	Data Type: *enumeration*
Parameter	**w**	**r**	
ID	**w**	**r**	Data Type: *ID*
SizePolicy	**w**	**r?**	Data Type: *enumeration* Allows printing even if the container size does not match the requirements of the data. If not present, the behavior uses system defaults.
Abort	**w?**	**r?**	Emit an error and abort printing
ClipToMaxPage	**w?**	**r?**	Clip page contents to size of the container. Printed area is centered in the source image.
ReduceToFit	**w?**	**r?**	Scale down page contents to fit the container; maintain the aspect ratio.
Status	**w**	**r**	Data Type: *enumeration*
Available	**w**	**r**	

5.9 FontPolicy resource

Name or Value	Manager	Worker	Description
Class	**w**	**r**	Data Type: *enumeration*
Parameter	**w**	**r**	
ID	**w**	**r**	Data Type: *ID*
PreferredFont	**w?**	**!r**	Data Type: *NMTOKEN* Ignored
Status	**w**	**r**	Data Type: *enumeration*
Available	**w**	**r**	
UseDefaultFont	**w?**	**r?**	Data Type: *boolean* If neither ***UseDefaultFont*** nor ***UseFontEmulation*** is specified, the NexPress uses the ***UseDefaultFont*** setting in the system default hot folder. If ***UseFontEmulation*** is specified, the default value is false.
false	**w?**	**r?**	
true	**w?**	**r?**	Resort to a default font if font cannot be found.
UseFontEmulation	**w?**	**r?**	Data Type: *boolean* If neither ***UseDefaultFont*** nor ***UseFontEmulation*** is specified, the NexPress uses the ***UseFontEmulation*** setting in the system default hot folder. If ***UseDefaultFont*** is specified, the default value is false.
false	**w?**	**r?**	
true	**w?**	**r?**	Emulate a required font if font cannot be found.

5.10 GatheringParams resource

Name or Value	Manager	Worker	Description
Class	**w**	**r**	Data Type: *enumeration*
Parameter	**w**	**r**	
ID	**w**	**r**	Data Type: *ID*
NoOp = false	**w**	**r**	Data Type: *boolean*
false	**w?**	**r?**	
true	**w?**	**r?**	Ignore this process
Status	**w**	**r**	Data Type: *enumeration*
Available	**w**	**r**	

5.11 HoleMakingParams resource

Name or Value	Manager	Worker	Description
Class	**w**	**r**	Data Type: *enumeration*

Name or Value	Manager	Worker	Description
Parameter	**w**	**r**	
ID	**w**	**r**	Data Type: *ID*
NoOp = false	**w**	**r**	Data Type: *boolean*
false	**w?**	**r?**	
true	**w?**	**r?**	Ignore this process
Status	**w**	**r**	Data Type: *enumeration*
Available	**w**	**r**	

5.12 InterpretingParams resource

Name or Value	Manager	Worker	Description
Class	**w**	**r**	Data Type: *enumeration*
Parameter	**w**	**r**	
ID	**w**	**r**	Data Type: *ID*
NXP: KnockoutColors = false	**w?**	**r?**	Data Type: *boolean* Determines behavior of the overprint conversion feature " Colors Knockout".
false	**w?**	**r?**	No knockout is performed. Default Value.
true	**w?**	**r?**	The operation to set colors to knockout on PDF data will be performed by the NexPress preflight processor.
NXP: OverprintBlack = false	**w?**	**r?**	Data Type: *boolean* Determines behavior of the overprint conversion feature "Black Overprint" for the NexPress front end.
false	**w?**	**r?**	Black will not be set to overprint. Default Value.
true	**w?**	**r?**	Black to overprint is enabled.
HonorPDFOverprint = false	**w?**	**r?**	Data Type: *boolean* If true, instructs the RIPs to perform OverprintPreview.
false	**w?**	**r?**	OverprintPreview disabled. Default Value.
true	**w?**	**r?**	OverprintPreview enabled.
Status	**w**	**r**	Data Type: *enumeration*
Available	**w**	**r**	

5.13 LayoutPreparationParams resource

Name or Value	Manager	Worker	Description
Class	**w**	**r**	Data Type: *enumeration*
Parameter	**w**	**r**	

Name or Value	Manager	Worker	Description
CreepValue	**w?**	**r?**	Data Type: *XYPair* Specifies the horizontal and vertical creep compensation values in points. The first value is the creep compensation for horizontal gutters; the second value is for vertical gutters. In the NexPress, gutters can only be decremented for creep compensation. X and Y <= 0. Positive values are set to 0. If not present, the creep values are taken from the NexPress *substrate type* in the Media Catalog named in **Media**/***@Brand*** for the body media defined by **Media**/**Location**/***@LocationName***=*Media0*.
FrontMarkList	**w?**	**r?**	Data Type: *NMTOKENS* List of marks to be produced on the sheet surface. During two-sided printing, also affects back side.
ColorControlStrip	**w?**	**r?**	Enable Control Density Strip
CutMark	**w?**	**r?**	Enable Cut Marks
JobField	**w?**	**r?**	Enable Slug Line
IdentificationField	**w?**	**r?**	Enable Bar Code. NOTE: ***NXP:MarkParams/@NXP:BCFinishingDeviceID*** must have a value of 2 or greater to enable bar code printing.
NXP:NGIdentificationField	**w?**	**r?**	Enable NG Bar Code. Specifies where and whether a barcode identifier mark for the NexGlosser is placed and printed onto each glossed sheet.
NXP:DuploSCC645Mark	**w?**	**r?**	Enable 645 Mark. Specifies where and whether an optical registration mark for the Duplo DC-645 SCC unit is placed and printed onto each glossed sheet.
NXP:DuploSCCDBMMark	**w?**	**r?**	Enable DBM Mark. Specifies where and whether an optical registration mark for the Duplo DSF-2000 DBM Booklet maker with SCC unit shall be placed and printed onto each glossed sheet.

Name or Value	Manager	Worker	Description
Gutter = "0.0 0.0"	**w?**	**r?**	Data Type: *XYPair* Defines the width in points of the horizontal and vertical gutters formed between rows and columns of pages in a multi-up sheet layout. The first value is the width of all horizontal gutters; the second value is the width of all vertical gutters. For creeping gutters (identified by ***@VerticalCreep*** or ***@HorizontalCreep***), specifies the initial gutter width. If not present, the implied value is *"0.0 0.0"* which means that the page cells of a multi–up grid of page cells touch.
HorizontalCreep	**w?**	**r?**	Data Type: *IntegerList* Specifies which horizontal gutters creep. Values are zero-based indexes. NexPress supports a maximum of two columns, so a value of "0" indicates that the first and only horizontal gutter creeps. MUST only be present when ***@PageDistributionScheme*** = *Saddle*.
0	**w?**	**r?**	First and only horizontal gutter creeps.
ID	**w**	**r**	Data Type: *ID*
ImageShift	**w?**	**r?**	Specifies positioning of the page cell grid, inclusive of any gutter widths, onto the substrate. The value defines the margin for Param imposition mode. When defining **LayoutParams/ImageShift** with Param imposition mode, **PageCell/ImageShift** must not be defined. Because ***ShiftBack*** is not supported, the front and back origins are locked for two-sided printing. A shift of the page cell grid on the front side forces a shift of the page cell grid on the back side in the direction that maintains alignment with the page cell grid on the front. The origin of the page cell grid of the front is the lower left corner of the substrate, and the origin of the coordinate system of the back side is the lower right corner. For example, a front side horizontal shift of x=144 results in a corresponding back side horizontal shift of x= -144. See **ImageShift** below.

Name or Value	Manager	Worker	Description
NumberUp	**w?**	**r?**	Data Type: *XYPair* Defines a regular, multi-up grid of page cells into which content pages are mapped. The first value is the number of columns, the second value is the number or rows in the multi-up grid. Restrictions on the allowed values of this attribute are detailed in a separate table below.
PageCell	**w?**	**r?**	Describes how page contents will be imaged onto page cells. All cells on a sheet must use the same page cell size. See **PageCell** below.
PageDistributionScheme = Sequential	**w?**	**r?**	Data Type: *NMTOKEN* Specifies how pages are distributed onto a multi-up grid of finished page cells defined by ***@NumberUp***. Restrictions on the allowed values of this attribute are detailed in a separate table below.
Perfect	**w?**	**r?**	Distribute pages onto a sequence of one or more signatures in proper order for perfect binding.
Saddle	**w?**	**r?**	Distribute pages onto a sequence of one or more imposition layouts in proper order for saddle stitch binding.
Sequential	**w?**	**r?**	Distribute pages onto the multi-up layout according to the value of ***@PresentationDirection***. Default Value.
PresentationDirection	**w?**	**r?**	Data Type: *enumeration* Indicates the order in which content pages will be distributed into page cells of the ***@NumberUp*** layout. Permutations specify page flow and are dependent upon the finishing intent specified.
zYx	**w?**	**r?**	For use with Saddle Stitch and Perfect Bound. Designates Page Distribution of "Saddle Stitch" or "Perfect Bound (2nd Fold Ver)". See 5.11.5.
zyX	**w?**	**r?**	For use with Saddle Stitch. Designates Page Distribution of "Saddle Stitch Japan". See 5.11.5.
zxY	**w?**	**r?**	For use with Perfect Bound. Designates Page Distribution of "Perfect Bound (2nd Fold Hor)". See 5.11.5.
zYX	**w?**	**r?**	For use with Perfect Bound. Designates Page Distribution of "Perfect Bound Japan (2nd Fold Ver)". See 5.11.5.

Name or Value	Manager	Worker	Description
zXY	**w?**	**r?**	For use with Perfect Bound. Designates Page Distibution of "Perfect Bound Japan (2nd Fold Hor)". See 5.11.5.
Xyz	**w?**	**r?**	For use with Sequential. Designates Page Distribution of "Consecutive". See 5.11.5.
zXy	**w?**	**r?**	For use with Sequential. Designates Page Distribution of either "Cut and Stack" or "Odd-Even Perfecting". See 5.11.5.
Rotate	**w?**	**r?**	Data Type: *enumeration* For sheet rotation with the M700e finisher
Rotate180	**w?**	**r?**	Sheet rotation by 180 degrees
Sides = TwoSidedFlipY	**w?**	**r?**	Data Type: *enumeration* Indicates whether the content layout is imaged on one or both sides of the media. ***NOTE: The JDF 1.3 specification uses a default value of OneSidedFront.***
OneSidedFront	**w?**	**r?**	Page content is imaged onto the front side of the media.
TwoSidedFlipX	**w?**	**r?**	Page content is imaged onto both the front and back sides of media sheets so that the corresponding page cells back up to eachother when flipping around the X-axis. Equivalent to "Work and Tumble".
TwoSidedFlipY	**w?**	**r?**	Page content is imaged onto both the front and back sides of media sheets so that the corresponding page cells back up to eachother when flipping around the Y-axis. Equivalent to "Work and Turn". Default Value.
StackDepth = 0	**w?**	**r?**	Data Type: *integer* The number of sheets in a stack when imposing along the Z axis. This parameter is used together with ***PageDistributionScheme*** and ***PresentationDirection*** to determine the distribution of pages in the output stack. Restrictions on the allowed values of this attribute are detailed in a separate table below.
0	**w?**	**r?**	Entire job defines one stack. Default Value.
1	**w?**	**r?**	Any non-zero value is processed as "1".
Status	**w**	**r**	Data Type: *enumeration*
Available	**w**	**r**	
StepDocs	**w?**	**!r**	Data Type: *XYPair* Specifies the number of instance documents to impose on one sheet. Ignored.

Name or Value	Manager	Worker	Description
StepRepeat = "1 1 1"	**w?**	**r?**	Data Type: *IntegerList* A list of three integers specifying the number of identical pages to impose. The first value is the number of repeats along the X axis,the second value is the number of repeats along the Y axis, and the third value is the number of repeats down the stack (along the Z axis). Restrictions on the allowed values of this attribute are detailed in a separate table below.
1 1 1	**w?**	**r?**	No Step and Repeat. Default Value.
X 1 1 where (X>1)	**w?**	**r?**	Repeat columns.
1 Y 1 where (Y>1)	**w?**	**r?**	Repeat rows.
X Y 1 where (X>1, Y>1)	**w?**	**r?**	Repeat columns and rows.
VerticalCreep	**w?**	**r?**	Data Type: *IntegerList* Specifies which vertical gutters creep. Values are zero-based indexes. NexPress supports a maximum of two rows, so a value of "0" indicates that the first and only vertical gutter creeps. MUST only be present when ***@PageDistributionScheme*** = *Saddle*
0	**w?**	**r?**	First and only vertical gutter creeps.
NXP:ImpoTemplate	**w?**	**r?**	NexPress proprietary element. Identifies an imposition template, and enables NexPress LayoutPreparation process in Template mode. See **NXP:ImpoTemplate** below.
NXP:MarkParams *	**w?**	**r?**	NexPress proprietary element. Specifies where special NexPress imposition marks will be placed onto surfaces. See **NXP:MarkParams** below.

5.13.1 ImageShift

Name or Value	Manager	Worker	Description
PositionX = None	**w?**	**r?**	Data Type: *enumeration*
Center	**w?**	**r?**	For the NexPress front end, this centers the grid both horizontally and vertically. NOTE: This is a different behavior than defined by the JDF Specification.
None	**w?**	**r?**	Default Value.
ShiftFront = "0 0"	**w?**	**r?**	Data Type: *XYPair* Specifies the Front side offset for Plain imposition mode. Default Value: 0 0.

5.13.2 PageCell

Name or Value	Manager	Worker	Description
TrimSize	**w?**	**r**	If not present, the default value is derived from the content file's pages. In the case of PDF, the value is taken from the /***TrimBox*** key of the Page dictionary. If the ***TrimSize*** attribute is present, it specifies the dimensions of an imposition cell in 1/72 inch units. NOTE: It is recommended that ***TrimSize*** always be supplied.
FitPolicy	**w?**	**r**	Refer to section 5.8 for details.
ImageShift	**w?**	**r**	Specifies the positioning of a single page onto the substrate. In most cases this is used for positioning page content on the sheet surface where the page content itself represents an already imposed sheet layout. This is also known as Plain imposition in the context of the NexPress front end. When attributes of ***PageCell/@ImageShift*** are supplied, any values of ***LayoutPreparationParams/@ImageShift*** are ignored. See **ImageShift** below.
NXP:Bleed	**w?**	**r**	Data type: *Number* This private attribute specifies a bleed value in points to be used for the calculation of an implicit TrimBox of a PDF page. If not specified, the implied value is *0*. The TrimBox is calculated in case of /TrimBox = /BleedBox and is assumed to reside inside the /BleedBox a distance as specified by the value of the NXP:Bleed attribute.

5.13.2.1 ImageShift

Name or Value	Manager	Worker	Description
ShiftFront = "0 0"	**w?**	**r?**	Data Type: *XYPair* Specifies the Front side offset for Plain imposition mode. Default Value: 0 0.
ShiftBack	**w?**	**r?**	Data Type: *XYPair* Specifies the Back side offset for Plain imposition mode. If not specified, ***ShiftBack*** is calculated from ***ShiftFront***..

5.13.3 NXP:ImpoTemplate

Name or Value	Manager	Worker	Description

Name or Value	Manager	Worker	Description
NXP:Name	**w**	**r**	Data Type: *string* Identifies an element of the CDFE Imposition Template resource category.
NXP:Type	**w?**	**r?**	Data Type: *enumeration* Identifies the type of page distribution for reiterated ImpositionTemplate resources.
PagePairs	**w?**	**r?**	Pages are distributed by pairs in a proper order for saddle stitch binding.
PageSets	**w?**	**r?**	Pages are divided into sets that are distributed continuously set by set.

5.13.4 NXP:MarkParams

Name or Value	Manager	Worker	Description
NXP:BCFinishingDeviceID = 1	**w?**	**r?**	Data Type: *integer* Identifier for device-specific bar code parameter sets.
1	**w?**	**r?**	No device (no bar code is printed)
2	**w?**	**r?**	Custom device
>2	**w?**	**r?**	Reserved for use by NexPress
NXP:BCOnFront = false	**w?**	**r?**	Data Type: *boolean* Specifies the substrate side on which to print the bar code.
false	**w?**	**r?**	Back side. Default Value.
true	**w?**	**r?**	Front side.
NXP:BCReverseNumbering = true	**w?**	**r?**	Data Type: *boolean* Specifies the numbering direction for the serial sheet count field within the supplied **NXP:Pattern**.
false	**w?**	**r?**	Ascending
true	**w?**	**r?**	Descending. Default Value.
NXP:Mode	**w?**	**r?**	Data Type: *enumeration* Specifies the location mode of the imposition mark relative to the surface. If ***NXP:Type*** = *CutMark*, this attribute is ignored. NOTE: Cutmarks are always placed relative to the trim box of imposed pages.
Bottom	**w?**	**r?**	Placement near the lower left corner.
Right	**w?**	**r?**	Placement near the upper right corner.
Left	**w?**	**r?**	Placement near the upper left corner.
Top	**w?**	**r?**	Placement near the upper right corner.
Explicit	**w?**	**r?**	Placement is explicitly specified by means of ***NXP:Position*** and ***NXP:Rotate*** values

Name or Value	Manager	Worker	Description
NXP:Pattern	**w?**	**r?**	Data Type: *String* Pattern for the IdentificationField (= bar code identifier). Possible values using a regular expression syntax are specified in CDFE JT Spec. e.g. "R{6}S{3}T{3}J{10}".
NXP:Position	**w?**	**r?**	Data Type: *XYPair* Specifies the position of the lower left mark element corner relative to the lower left surface corner.
NXP:Rotate	**w?**	**r?**	Data Type: *enumeration* Specifies the orthogonal rotation angle of the imposition mark.
Rotate0	**w?**	**r?**	
Rotate90	**w?**	**r?**	
Rotate180	**w?**	**r?**	
Rotate270	**w?**	**r?**	
NXP:Type	**w**	**r**	Data Type: *NMTOKEN* Specifies the type of NexPress-specific imposition mark.
ColorControlStrip	**w?**	**r?**	
CutMark	**w?**	**r?**	
IdentificationField	**w?**	**r?**	
JobField	**w?**	**r?**	
NXP:NGIdentificationField	**w?**	**r?**	
NXP:DuploSCC645Mark	**w?**	**r?**	
NXP:DuploSCCDBMMark	**w?**	**r?**	
NXP:UserText	**w?**	**r**	Data Type: *string* User-defined text for the *JobField* (= slug line text). Only used if NXP:Type = "JobField"

5.13.5 Details on LayoutPreparationParams usage

The ***NumberUp, PageDistributionScheme, PresentationDirection, Sides, StackDepth,*** and ***StepRepeat*** attributes of **LayoutPreparationParams** parameterize how **RunList** (Document) pages are distributed onto press sheets. This section describes usage restrictions for these attributes.

The following table relates a finishing method to combinations of ***@PageDistributionScheme, @PresentationDirection***, and ***@StackDepth*** required to achieve appropriately imposed sheets.

Finishing Method	*PageDistributionScheme*	*PresentationDirection*	*StackDepth*
Saddle Stitch	Saddle	zYx	--
Saddle Stitch Japan	Saddle	zyX	--
Perfect Bound (2^{nd} Fold Ver)	Perfect	zYx	1
Perfect Bound	Perfect	zxY	1

Finishing Method	***PageDistributionScheme***	***PresentationDirection***	***StackDepth***
(2nd Fold Hor)			
Perfect Bound Japan (2nd Fold Ver)	Perfect	zYX	1
Perfect Bound Japan (2nd Fold Hor)	Perfect	zXY	1
Consecutive	Sequential	Xyz	--
Odd-Even Perfecting	Sequential	zXy	1
Cut and Stack	Sequential	zXy	--

To guarantee that each ***NumberUp*** cell grid can be subdivided evenly into a grid of product sheets applied repeatedly *(**StepRepeat**)* for each ***PageDistributionScheme*** attribute value (*Saddle*, *Perfect*, or *Sequential*), the NexPress front end's imposition engine only supports a restricted range of ***NumberUp*** cell grid values.:

NumberUp	**StepRepeat**	**Product Sheet**	**Subdivision Rule**
N_X x N_Y	P_X x P_Y	R_X x R_Y	$N_X = P_X \times R_X$; $N_Y = P_Y \times R_Y$

For ***PageDistributionScheme*** *=Saddle*, additional constraints are required for use with **LayoutPreparationParams**/@***Sides*** because the spine's gutter must be parallel with the flip axis. These constraints are referred to as **(C1)** and **(C2)** as defined and referenced in the following tables:

Constraint (in term of JDF)	**Name**	**Meaning (in GUI terminology)**
Sides = "TwoSidedFlipX" not allowed	(C1)	2–sided, Head–to–Foot not allowed
Sides = "TwoSidedFlipY" not allowed	(C2)	2–sided, Head–to–Head not allowed

Ranges and Constraints for the values of ***PageDistributionScheme***, ***NumberUp***, and ***StepRepeat:***

PageDistributionScheme	***NumberUp***	***StepRepeat***	Product Sheet	Additional Constraints
Perfect	1 x 1	1 x 1	1 x 1	
	2 x 1	2 x 1	1 x 1	
	1 x 2	1 x 2	1 x 1	
	2 x 2	2 x 2	1 x 1	
	2 x 1	1 x 1	2 x 1	
	2 x 2	1 x 2	2 x 1	
	1 x 2	1 x 1	1 x 2	
	2 x 2	2 x 1	1 x 2	
	2 x 2	1 x 1	2 x 2	
Saddle	2 x 1	1 x 1	2 x 1	(C1)
	2 x 2	1 x 2	2 x 1	(C1)
	1 x 2	1 x 1	1 x 2	(C2)
	2 x 2	2 x 1	1 x 2	(C2)
Sequential *)	N_X x N_Y	1 x 1	N_X x N_Y	$1 \le N_X \le 20$, $1 \le N_Y \le 20$

PageDistributionScheme	*NumberUp*	*StepRepeat*	Product Sheet	Additional Constraints
	N_X x N_Y	1 x N_Y	N_X x 1	$1 \le N_X \le 20$, $1 \le N_Y \le 20$
	N_X x N_Y	N_X x 1	1 x N_Y	$1 \le N_X \le 20$, $1 \le N_Y \le 20$
	N_X x N_Y	N_X x N_Y	1 x 1	$1 \le N_X \le 20$, $1 \le N_Y \le 20$

*) For ***PageDistributionScheme*** = *Sequential*, the ranges and constraints are as enumerated in the table above. The Product Sheet column values interpreted as Cut Blocks for finishing are only applicable for *Consecutive* page distribution (***PresentationDirection*** = "*XYz*"). For *Odd–Even Perfecting* and *Cut and Stack* (***PresentationDirection*** = "*zXY*"), each grid cell represents a Cut Block of its own.

5.14 Media resource

This resource may be partitioned for the purposes of specifying mixed media. This should not be done if **DigitalPrintingParams** is partitioned for the same purpose.

Name or Value	Manager	Worker	Description
Brand	**w?**	**r?**	Data Type: *string* Used by the NexPress front end to identify a supported substrate by its name in the *Media Catalog*. The list of supported media, including the names and characterization information can be obtained directly from the capabilities file of a target NexPress front end. Refer to [KNDIR] for more information about the NexPress front end's capabilities file. If ***Brand*** is not supplied, the NexPress uses the media specified for the system default hot folder.
Class	**w**	**r**	Data Type: *enumeration*
Consumable	**w**	**r**	
Dimension	**w?**	**r?**	Data Type: *XYPair* The width and height dimensions for the Media, measured in points. The order of the XY pair also specifies the orientation of the substrate and how page content is to be imaged onto its surface(s). In the NexPress front end, the medium resource identified by the ***Brand*** attribute has rectangular dimensions specified as width W and height H. These dimensions form a rectangular area in either Landscape or Portait orientation as origined from (0,0). Where W ≤ H: **Portrait** – (0, 0), (W, H) **Landscape** – (0, 0), (H, W) If ***Dimension*** is not supplied, the NexPress uses the media dimensions specified for the system default hot folder.

Name or Value	Manager	Worker	Description
Location	**w**	**r**	Data Type: *refelement* Identifies the "logical tray" corresponding to this Media resource. These logical trays have special names that control media selection when ***PartIDKeys*** is not used for **Media** or **DigitalPrintingParams**. See **Location** below.
Media	**w?**	**r?**	Data Type: *refelement* For partitioned **Media**.
PartIDKeys	**w?**	**r?**	Data Type: *enumerations* The **Media** resource can be partitioned to designate mixed media usage within the job. Alternatively, ***MediaRef*** can be added to a partitioned **DigitalPrintingParams**. The **Media** resource should not be partitioned if **DigitalPrintingParams** will also be partitioned by ***SheetIndex*** and/or ***RunTags*** and contain ***MediaRef***.
SheetIndex	**w←**	**r?**	Partition using the NexPress cover-mode method of page-to-media mapping.
RunTags	**w←**	**r?**	Partition using the NexPress label method of page-to-media mapping.
SheetIndex RunTags	**w←**	**r?**	Partition using both NexPress cover-mode and NexPress label methods of page-to-media mapping. (NOTE: You must specify the value of PartIDKeys in this order -- SheetIndex, then RunTags -- when they are used together.)
RunTags	**w?**	**r?**	Data Type: *NMTOKENS* Supports the NexPress label methods (stamp annotation and PPML/VDX labels) for PDL page to media mapping. Creates a partition consisting of sheets tagged with RunTags values. To support stamp annotation, RunTags values are limited to "SubstrateTypeCover" and "SubstrateTypeInsert". To support PPML/VDX labels, values must correspond to the arbitrary values of the PPML DOCUMENT/@Label attributes. A RunTags value is ignored if it does not correspond to a PDF stamp annotation or PPML DOCUMENT/@Label.
SubstrateTypeCover	**w?**	**r?**	Defines a partition for substrates tagged with the stamp annotation: SubstrateTypeCover.
SubstrateTypeInsert	**w?**	**r?**	Defines a partition for substrates tagged with the stamp annotation: SubstrateTypeInsert.

Name or Value	Manager	Worker	Description
<arbitrary PPLM DOCUMENT/@Label >	**w?**	**r?**	Defines a partition for substrates in a PPM Document that are assigned the arbitrary label.
SheetIndex	**w?**	**r?**	Data Type: *IntegerRangeList* Creates a partition that maps sheets to this **Media.**
0 (used without ***RunTags*** key)	**w?**	**r?**	Defines a partition consisting of the first sheet.
-1 (used without ***RunTags*** key)	**w?**	**r?**	Defines a partition consisting of the last sheet.
0 -1 (used without ***RunTags*** key)	**w?**	**r?**	Defines a partition consisting of the first and last sheets.
1 ~ -1 (used with ***RunTags*** key)	**w?**	**r?**	Defines a partition consisting of all but the first sheet.
0 ~ -2 (used with ***RunTags*** key)	**w?**	**r?**	Defines a partition consisting of all but the last sheet.
1 ~ -2 (used with ***RunTags*** key)	**w?**	**r?**	Defines a partition consisting of all but the first and last sheets.
Status	**w**	**r**	Data Type: *enumeration*
Available	**w**	**r**	
NXP:ClearCoat = None	**w?**	**r?**	Data Type: *enumeration* **DEPRECATED AFTER SYSTEM RELEASE 8.2. Use NXP:ClearCoatFront and NXP:ClearCoatBack for proper control of gloss type on media..** NexPress proprietary attribute controls application of a "Clear Coat" Protection finish (clear inverse mask) to a **Media**. Identifies the side(s) of the medium to which the "Clear Coat" is applied. NOTE: Controls clear coat "protection", not "gloss".
None	**w?**	**r?**	No application to either side of the media. Default Value.
Front	**w?**	**r?**	Application to the front side.
Back	**w?**	**r?**	Application to the back side.
Both	**w?**	**r?**	Application to the front and back sides.
NXP:ClearCoatBack = none	**w?**	**r?**	Data Type: *enumeration* NexPress proprietary attribute controls application of a "Clear Coat" finish (clear inverse mask) to the back side of a **Media**. Value identifies the type of "Clear Coat" to be applied.
none	**w?**	**r?**	No clear coat application to back side of media. Default Value.
protection	**w?**	**r?**	Application of protection coating to back side of media.

Name or Value	Manager	Worker	Description
gloss	**w?**	**r?**	Application of gloss coating to back side of media.
iqFlood	**w?**	**r?**	Application of clear toner using "Flood" selection to back side of media.
iqImage	**w?**	**r?**	Application of clear toner using "Image" selection to back side of media.
xdPhoto	**w?**	**r?**	Application of photographic clear toner using "Photo" selection to back side of media.
xdGraphic	**w?**	**r?**	Application of photographic clear toner using "Graphic" selection to back side of media.
raiseAllSimple	**w?**	**r?**	Application of Dimensional clear toner to the back side of media. Also called Dimensional Easy Overcoat.
matteClearEasy	**w?**	**r?**	Application of Matte Clear toner to the back side of the media.
NXP:ClearCoatFront = none	**w?**	**r?**	Data Type: *enumeration* NexPress proprietary attribute controls application of a "Clear Coat" finish (clear inverse mask) to the front side of a **Media**. Value identifies the type of "Clear Coat" to be applied.
none	**w?**	**r?**	No clear coat application to front side of media. Default Value.
protection	**w?**	**r?**	Application of protection coating to front side of media.
gloss	**w?**	**r?**	Application of gloss coating to front side of media.
iqFlood	**w?**	**r?**	Application of clear toner using "Flood" selection to front side of media.
iqImage	**w?**	**r?**	Application of clear toner using "Image" selection to front side of media.
xdPhoto	**w?**	**r?**	Application of photographic clear toner using "Photo" selection to front side of media.
xdGraphic	**w?**	**r?**	Application of photographic clear toner using "Graphic" selection to front side of media.
raiseAllSimple	**w?**	**r?**	Application of Dimensional clear toner to the front side of the media. Also called Dimensional Easy Overcoat.
matteClearEasy	**w?**	**r?**	Application of Matte Clear toner to the front side of the media.

5.14.1 Location

Name or Value	Manager	Worker	Description

Name or Value	Manager	Worker	Description
LocationName	**w?**	**r?**	Data Type: *string* Name of a logical substrate supply. ***LocationName*** is generally required by NexPress; however, Body and Cover media will be inferred using partitioning if **Location/@LocationName** omitted.
Media0	**w?**	**r?**	Corresponds to the *Body* substrate.
Media1	**w?**	**r?**	Corresponds to *Cover* substrate.
Media2	**w?**	**r?**	Corresponds to *Insert* substrate.
Media3	**w?**	**r?**	Corresponds to *Separators* substrate.
Media4	**w?**	**r?**	Corresponds to the *Insert2* substrate.
Media5	**w?**	**r?**	Corresponds to the *Insert3* substrate.
Media6	**w?**	**r?**	Corresponds to the *Insert4* substrate.
Media7	**w?**	**r?**	Corresponds to the *Insert5* substrate.
Media8	**w?**	**r?**	Corresponds to the *Insert6* substrate.
Media9	**w?**	**r?**	Corresponds to the *Insert7* substrate.
Media10	**w?**	**r?**	Corresponds to the *Insert8* substrate.
LargeCapacity	**w?**	**r?**	Same as Media0. Corresponds to the *Body* substrate.
Top	**w?**	**r?**	Same as Media1. Corresponds to the *Cover* substrate.

5.15 NodeInfo resource

Name or Value	Manager	Worker	Description
TargetRoute	**w?**	**r?**	Data Type: *URL* URL where JDF is to be sent after completion. The completed JDF file can be sent to a web server by specifying "HTTP://servername:PORT#"
NXP:DeleteWhenDone	**w?**	**r?**	Data Type: *boolean* Overrides the Portal Configuration bDeleteWhenDone during processing of this JDF ticket. Only applies to JDF jobs submitted through the JDF Hotfolder. Portal Configuration is set by c:\CDFE_CONFIG\InitialConfig\JDFPortal_cfg.xml.
true	**w?**	**r?**	If job submitted by JDF Hotfolder, delete when the job completes or aborts.
false	**w?**	**r?**	If job submitted by JDF Hotfolder, DO NOT delete when the job completes or aborts.
NXP:DeviceWorkflow	**w?**	**r?**	Used to define stop points for JDF job during processing. See **NXP:DeviceWorkflow** below.

5.15.1 NXP:DeviceWorkflow

Name or Value	Manager	Worker	Description

Name or Value	Manager	Worker	Description
NXP:StopPoints	**w**	**r**	Data Type: *enumerations* Stop job after each specified processing phase. A JDF job ticket can only add stop points; stop points defined in the default Job Ticket cannot be disabled.
NXP:Submitted	**w?**	**r?**	Enable job stop after submission.
NXP:PDFCreated	**w?**	**r?**	Enable job stop after PDF has been created. (Added for completeness; JDF only accepts PDF content files).
NXP:ResourcesChecked	**w?**	**r?**	Enable job stop after Resource Check.
NXP:RIPped	**w?**	**r?**	Enable job stop has been RIPped.

5.16 RenderingParams resource

Name or Value	Manager	Worker	Description
ADBE:RemoveOverPrintParams	**w?**	**r?**	Data Type: *refelement* See **ADBE:RemoveOverPrintParams** below.
NXP:CLCDDIQParams	**w?**	**r?**	Data Type: *refelement* See **NXP:CLCDDIQParams** below.
NXP:PureBlack = true	**w?**	**r?**	Data Type: *boolean* Render RGB Black (0,0,0) as CMYK (0,0,0,100). Cannot be enabled when ***@NXP:RichBlack*** = true.
true	**w?**	**r?**	Enable. Default Value.
false	**w?**	**r?**	Disable
NXP:RichBlack = false	**w?**	**r?**	Data Type: *boolean* Render RGB Black (0,0,0) as CMYK (80, 60, 40, 100); and CMYK Black (0,0,0,100) as CMYK (80, 60, 40, 100). Cannot be enabled when ***@NXP:PureBlack*** = true OR when **ColorSpaceConversionParams/ ColorSpaceConversionOp/ @RGBGray2Black** = true.
true	**w?**	**r?**	Enable
false	**w?**	**r?**	Disable. Default Value.
NXP:DMCLProcessColorantThreshold	**w <--**	**r**	*Required if Dimensional Easy Overcoat is requested* **Data Type:** *integer* Defines the CMYK threshold to be used when generating the Dimensional Clear overcoat. **Range:** 0 - 255

Name or Value	Manager	Worker	Description
NXP:DMCLThresholdApplication	**w <--**	**r**	*Required if Dimensional Easy Overcoat is requested* **Data Type**: *string* This parameter controls the usage of the **DMCLProcessColorantThreshold** parameter.
Above			Dimensional overcoat will be generated if any of the CMYK channels have values greater than the overcoat threshold
Below			Dimensional overcoat will be benerated if any of the CMYK channels have values greater than the overcoat threshold
NXP:MTCLProcessColorantThreshold	**w <--**	**r**	*Required if Matte Clear Easy Overcoat is requested* **Data Type:** *integer* Defines the CMYK threshold to be used when generating the Matte Clear overcoat. **Range:** 0 - 255
NXP:MTCLThresholdApplication	**w <--**	**r**	*Required if Matte Clear Easy Overcoat is requested* **Data Type**: *string* This parameter controls the usage of the **MTCLProcessColorantThreshold** parameter.
Above			Matte Clear overcoat will be generated if any of the CMYK channels have values greater than the overcoat threshold
Below			Matte Clear overcoat will be benerated if any of the CMYK channels have values greater than the overcoat threshold

5.16.1 ADBE:RemoveOverPrintParams

Name or Value	Manager	Worker	Description
EmulateNAInkOverprint	**w?**	**r?**	Data Type: *boolean* Enable overprint for spot colors that are not available. No default value.
true	**w?**	**r?**	Enable
false	**w?**	**r?**	Disable

5.16.2 NXP:CLCDDIQParams

Name or Value	Manager	Worker	Description
PD = 0	**w?**	**r?**	Data Type: *integer* Sets "Print Density" CLCDD engine parameter. Values 0 - 10.
SF = 0	**w?**	**r?**	Data Type: *integer* Sets "Sharpness Filter" CLCDD engine parameter. Values 0 - 7.

Name or Value	Manager	Worker	Description
TS = DeviceDefault	**w?**	**r?**	Data Type: *NMTOKEN* Sets "Toner Savings" CLCDD engine parameter.
DeviceDefault	**w?**	**r?**	Engine uses its default setting. Default Value.
On	**w?**	**r?**	Enabled
Off	**w?**	**r?**	Disabled
TextGS = DeviceDefault	**w?**	**r?**	Data Type: *NMTOKEN* Sets "Text Gradation Smoothing" CLCDD engine parameter. ***This attribute is provided to enable future development; a value other than DeviceDefault or None may produce undesireable results. Enabling TextGS can distort the edges of text.***
DeviceDefault	**w?**	**r?**	Engine uses its default setting. Default Value.
None	**w?**	**r?**	Disabled
Fine	**w?**	**r?**	"Fine" selection.
Coarse	**w?**	**r?**	"Coarse" selection.
GraphicsGS = DeviceDefault	**w?**	**r?**	Data Type: *NMTOKEN* Sets "Graphics Gradation Smoothing" CLCDD engine parameter.
DeviceDefault	**w?**	**r?**	Engine uses its default setting. Default Value.
None	**w?**	**r?**	Disabled
Fine	**w?**	**r?**	"Fine" selection.
Coarse	**w?**	**r?**	"Coarse" selection.
ImageGS = DeviceDefault	**w?**	**r?**	Data Type: *NMTOKEN* Sets "Image Gradation Smoothing" CLCDD engine parameter.
DeviceDefault	**w?**	**r?**	Engine uses its default setting. Default Value.
None	**w?**	**r?**	Disabled
Fine	**w?**	**r?**	"Fine" selection.
Coarse	**w?**	**r?**	"Coarse" selection.
HTTextBW = DeviceDefault	**w?**	**r?**	Data Type: *NMTOKEN* Sets "Halftone Mode for Black-Only Text" CLCDD engine parameter.
DeviceDefault	**w?**	**r?**	Same as "Gradation" selection. Default Value.
Resolution	**w?**	**r?**	"Resolution" selection.
Gradation	**w?**	**r?**	"Gradation" selection.
ErrorDiffusion	**w?**	**r?**	"ErrorDiffusion" selection.

Name or Value	Manager	Worker	Description
HTGraphicsBW = DeviceDefault	**w?**	**r?**	Data Type: *NMTOKEN* Sets "Halftone Mode for Black-Only Graphics" CLCDD engine parameter.
DeviceDefault	**w?**	**r?**	Same as "Gradation" selection. Default Value.
Resolution	**w?**	**r?**	"Resolution" selection.
Gradation	**w?**	**r?**	"Gradation" selection.
ErrorDiffusion	**w?**	**r?**	"ErrorDiffusion" selection.
HTImageBW = DeviceDefault	**w?**	**r?**	Data Type: *NMTOKEN* Sets "Halftone Mode for Black-Only Images" CLCDD engine parameter.
DeviceDefault	**w?**	**r?**	Same as "Gradation" selection. Default Value.
Resolution	**w?**	**r?**	"Resolution" selection.
Gradation	**w?**	**r?**	"Gradation" selection.
ErrorDiffusion	**w?**	**r?**	"ErrorDiffusion" selection.
HTTextCMYK = DeviceDefault	**w?**	**r?**	Data Type: *NMTOKEN* Sets "Halftone Mode for Color Text" CLCDD engine parameter.
DeviceDefault	**w?**	**r?**	Same as "Gradation" selection. Default Value.
Resolution	**w?**	**r?**	"Resolution" selection.
Gradation	**w?**	**r?**	"Gradation" selection.
ErrorDiffusion	**w?**	**r?**	"ErrorDiffusion" selection.
HTGraphicsCMYK = DeviceDefault	**w?**	**r?**	Data Type: *NMTOKEN* Sets "Halftone Mode for Color Graphics" CLCDD engine parameter.
DeviceDefault	**w?**	**r?**	Same as "Gradation" selection. Default Value.
Resolution	**w?**	**r?**	"Resolution" selection.
Gradation	**w?**	**r?**	"Gradation" selection.
ErrorDiffusion	**w?**	**r?**	"ErrorDiffusion" selection.
HTImageCMYK = DeviceDefault	**w?**	**r?**	Data Type: *NMTOKEN* Sets "Halftone Mode for Color Images" CLCDD engine parameter.
DeviceDefault	**w?**	**r?**	Same as "Gradation" selection. Default Value.
Resolution	**w?**	**r?**	"Resolution" selection.
Gradation	**w?**	**r?**	"Gradation" selection.
ErrorDiffusion	**w?**	**r?**	"ErrorDiffusion" selection.

Name or Value	Manager	Worker	Description
HTLineCMYK = DeviceDefault	**w?**	**r?**	Data Type: *NMTOKEN* Sets "Halftone Mode for Color Lines" CLCDD engine parameter. ***This attribute is provided to enable future development; a value other than DeviceDefault may produce undesireable results.***
DeviceDefault	**w?**	**r?**	Same as "Gradation" selection. Default Value.
Resolution	**w?**	**r?**	"Resolution" selection.
Gradation	**w?**	**r?**	"Gradation" selection.
ErrorDiffusion	**w?**	**r?**	"ErrorDiffusion" selection.
ASTText = DeviceDefault	**w?**	**r?**	Data Type: *NMTOKEN* Sets "Adaptive Smoothing Technology for Text" CLCDD engine parameter.
DeviceDefault	**w?**	**r?**	Engine uses its default setting. Default Value.
None	**w?**	**r?**	Disabled
Fine	**w?**	**r?**	"Fine" selection.
Coarse	**w?**	**r?**	"Coarse" selection.
ASTGraphics = DeviceDefault	**w?**	**r?**	Data Type: *NMTOKEN* Sets "Adaptive Smoothing Technology for Graphics" CLCDD engine parameter.
DeviceDefault	**w?**	**r?**	Engine uses its default setting. Default Value.
None	**w?**	**r?**	Disabled
Fine	**w?**	**r?**	"Fine" selection.
Coarse	**w?**	**r?**	"Coarse" selection.
ASTImage = DeviceDefault	**w?**	**r?**	Data Type: *NMTOKEN* Sets "Adaptive Smoothing Technology for Images" CLCDD engine parameter.
DeviceDefault	**w?**	**r?**	Engine uses its default setting. Default Value.
None	**w?**	**r?**	Disabled
Fine	**w?**	**r?**	"Fine" selection.
Coarse	**w?**	**r?**	"Coarse" selection.
ASTLine = DeviceDefault	**w?**	**r?**	Data Type: *NMTOKEN* Sets "Adaptive Smoothing Technology for Lines" CLCDD engine parameter. ***This attribute is provided to enable future development; a value other than DeviceDefault may produce undesireable results.***
DeviceDefault	**w?**	**r?**	Engine uses its default setting. Default Value.

Name or Value	Manager	Worker	Description
None	**w?**	**r?**	Disabled
Fine	**w?**	**r?**	"Fine" selection.
Coarse	**w?**	**r?**	"Coarse" selection.

5.17 RunList resource

Name or Value	Manager	Worker	Description
Class	**w**	**r**	Data Type: *enumeration*
Parameter	**w**	**r**	
ComponentGranularity = Document	**w?**	**r?**	Data Type: *enumeration* Specifies the grouping of PDL pages that produce an output component instance.
Document	**w?**	**r?**	Required for PDF data. Default Value.
Set	**w?**	**r?**	Required for PPML/VDX data.
Directory	**w?**	**r?**	Data Type: *string* Defines a base path applicable to relative URIs in **LayoutElement/FileSpec/@URL**. If ***Directory*** is not supplied, the JDF Hotfolder location is used as a base path to any relative URI.
ID	**w**	**r**	Data Type: *ID*
PartUsage	**w?**	**r?**	Data Type: *enumeration*
Implicit	**w?**	**r?**	Required with RunList partitions using SheetIndex or RunIndex.
Sets	**w←**	**r?**	Data Type: *IntegerRangeList* Zero-based list of document set indices in a multi-document sets file. Only valid if ***LayoutElement/@ElementType*** = "MultiSet". If not present, all document sets are selected.
Status	**w**	**r**	Data Type: *enumeration*
Available	**w**	**r**	
LayoutElement	**w**	**r**	Describes the document content file. See **LayoutElement** below.
Disposition	**w?**	**r?**	Controls deletion of the content file after the JDF ticket has been processed. See **Disposition** element below.

5.17.1 LayoutElement

Name or Value	Manager	Worker	Description
ElementType = Unknown	**w?**	**r?**	Data Type: *enumeration*
Document	**w?**	**r?**	An ordered set of one or more pages, e.g. Adobe PDF.

Name or Value	Manager	Worker	Description
MultiDocument	**w?**	**?**	An ordered set of one or more documents including document breaks, e.g. PPML/VDX.
MultiSet	**w?**	**r?**	An ordered set of one or more document sets including document set breaks, e.g. PPML/VDX.
Unknown	**w?**	**r?**	The type will be ascertained from the content. Default Value.
FileSpec	**w**	**r**	URL and metadata for content file.

5.17.1.1 FileSpec

Name or Value	Manager	Worker	Description
URL	**w**	**r**	Location of the page content file as either an Absolute or a Relative URI. Supported schemes include *http:*, *ftp:*, *file:*, and *cid:*. A relative URI references content files within subdirectories of the JDF HotFolder. Relative URIs are located from the JDF HotFolder directory, or the location specified by **Runlist/@Directory**. NOTE: DFE Server processes must have sufficient read permissions to access resources specified with "file:".
Disposition	**w?**	**r?**	Controls deletion of the content file after the JDF ticket has been processed. See **Disposition** element below.

5.17.2 Disposition

Name or Value	Manager	Worker	Description
DispositionAction = "Delete"	**w?**	**r?**	Data Type: *enumeration* Only Delete is supported.
Delete	**w?**	**r?**	Remove content file upon abort or completion AND the date/time specified by ***Until*** has elapsed. Default Value.
Until = CurrentTime	**w?**	**r?**	Data Type: *dateTime* Date and Time when content file can be removed.

5.18 ScreeningParams resource

Name or Value	Manager	Worker	Description
Class	**w**	**r**	Data Type: *enumeration*
Parameter	**w**	**r**	
ID	**w**	**r**	Data Type: *ID*
Status	**w**	**r**	Data Type: *enumeration*
Available	**w**	**r**	

Name or Value	Manager	Worker	Description
NXP: ScreeningID = 1	**w?**	**r?**	Data Type: *string* Proprietary attribute to select the screening style. If not present, the default value is used. NOTE: Beginning with System Release 8.5, new screens are available only by name. Legacy screen enumerations continue to be supported in addition to the screen name for older screens. Supported ***ScreeningID*** names are listed in the NexPress Device Capabilities File.
1	**w?**	**r?**	Classic screening system. Default Value.
2	**w?**	**r?**	Optimum screening system.
3	**w?**	**r?**	Line screening system.
4	**w?**	**r?**	Supra screening system.
Classic	**w?**	**r?**	Classic screening system.
Optimum	**w?**	**r?**	Optimum screening system.
Line	**w?**	**r?**	Line screening system.
Supra	**w?**	**r?**	Supra screening system.
Stochastic	**w?**	**r?**	Stochastic screening system.
ClassicHD	**w?**	**r?**	ClassicHD (High Definition) screening system. Recommended for NexPress HD Inks only.

5.19 StitchingParams resource

Name or Value	Manager	Worker	Description
Class	**w**	**r**	Data Type: *enumeration*
Parameter	**w**	**r**	
ID	**w**	**r**	Data Type: *ID*
NoOp = false	**w**	**r**	Data Type: *boolean*
false	**w?**	**r?**	
true	**w?**	**r?**	Ignore this process
Status	**w**	**r**	Data Type: *enumeration*
Available	**w**	**r**	
StitchType	**w?**	**r?**	Data Type: *enumeration*
Corner	**w?**	**r?**	Angled corner stitching
Saddle	**w?**	**r?**	Booklet stitching
Side	**w?**	**r?**	Edge stitching

Note: the usage of StitchingParams relies on the Component (exchange resource) for rotation assignment

6 JDF Portal Configuration File

Some aspects of JDF Portal operation are specified by entries within the XML configuration file, "C:\CDFE_CONFIG\InitialConfig\JDFPortal_cfg.xml". It is not appropriate for most of these configuration entries to be changed from their installed default values. However, depending upon other tools in the JDF workflow, it may be desireable to change the following entries; others in the file should not be modified. These configurable entries are shown below with their default values:

```
<properties>
    …
    <componentConfig type="BASE::idlJDFPortalCfgObjectData" elementtype="struct">
        <description>The JDFPortal's private configuration data</description>
        <driverCount      value="50"       type="unsigned short"   elementtype="simple"/>
        <bDeleteWhenDone  value="TRUE"   type="boolean"         elementtype="simple"/>
        <LocatorTTL value="2" type="unsigned long"   elementtype="simple"/>
        …
    </componentConfig>
</properties>
```

6.1 driverCount

The JDF Portal assigns a driver thread to submit and subsequently monitor each active JDF job. A job is active after it is released from the Portal Job Queue to the Internal Job Queue for processing, until the job completes or has been aborted. "properties/componentConfig/driverCount/@value" determines the number of JDF submitted jobs that can simultaneously be active in the Device Queue. Poor performance may result from an excessively high driverCount value.

6.2 bDeleteWhenDone

This attribute value controls whether the JDF Portal will remove a job submitted using the JDF Hotfolder upon completion or when the job is aborted.

Jobs submitted using the JDF Hotfolder are deleted automatically only if bDeleteWhenDone is TRUE.

Jobs submitted using JMF/SubmitQueueEntry use QueueSubmissionParams/Disposition to control whether a job is automatically deleted; bDeleteWhenDone is not used for these jobs.

A job can override bDeleteWhenDone by defining NodeInfo/NXP:DeleteWhenDone within the JDF ticket.

6.3 LocatorTTL

The JDF Portal includes a multicast locator that supports integration with Creo Prinergy. The locator is enabled by default, but for installations that do not use Prinergy, this locator can be disabled by changing the LocatorTTL value to "0". When enabled, the portal broadcasts its JMF HTTP port location on multicast channel 234.5.6.7 using port 50006 with a Time-To-Live value determined by LocatorTTL.

7 PDF Reference

This chapter describes NexPress support for extensions to PDF as referenced in the [KNDIG]. Use of the PDF extensions (e.g. job submission) for the NexPress front end is provided in [KNDIG] and not here. The NexPress front end aligns with [PDF]. The following identifies these extensions to PDF, specifically in the area of media mapping.

For all tables in this chapter, the Manager is the Client Application accessing the NexPress front end, and the Worker is the NexPress front end responding to the Client Application. "Client Application" is used in the broadest terms to indicate any entity accessing the NexPress front end.

The NexPress front end and NexPress workflow tools support only the keys, values and sub-elements identified below. Any keys, values or sub-elements other then those defined below will be ignored. If an illegal or unsupported attribute value is identified, the rules identified above will be used.

7.1 PDF Stamp Annotations

Up to three different substrates may be mapped to the various pages of a single PDF print job in the case of a PDF-based print job submitted to a NexPress digital production color press. In a workflow, the creator of the job ticket (i.e. the client to the NexPress front end) specifies this mapping.

The NexPress front end uses **PDF Stamp Annotation** to map specific PDF pages to specific physical media loaded in the NexPress digital production color press. **PDF Stamp Annotation** places specific meta-data, in the form of specialized PDF stamp annotations, within the PDF. The **PDF Stamp Annotation** method uses this meta-data to assign pages of a PDF file to logical media *Body*, *Cover*, or *Insert*. In a job ticket, the three supply trays of the NexPress digital production color press can be assigned a name that matches that of the logical media. The job ticket maps the stamp annotation meta-data to the physical substrate for the job. The names of the physical substrates are listed in the NexPress front end's media catalog; the media catalog is contained in the NexPress front end's Device Capabilities File.

The **PDF Stamp Annotation** method

- uses a combination of embedded PJTF, a Virtual Printer Job Ticket Template, and the PDF stamp annotations to define the media mapping.
- is associated with a specific page within the PDF file through a /**Page** object. The /**Page** object includes the /**Annots** object that represents the PDF stamp annotation as defined in [PDF].
- places a PDF stamp annotation in an annotation object (i.e. /**Annots**) inside a PDF file, per [PDF]. An /**Annots** object is a PDF dictionary that describes the type, appearance, position, and other attributes of an annotation on a page.

PDF stamp annotations used for the **PDF Stamp Annotation** method are visible and printable if their graphical representation overlaps any part of the page content area when displayed by the Adobe® Acrobat® viewer application, and they are also visible in the Acrobat® viewer when located in the gray area off the page content area. If the PDF stamp annotation overlaps the page area it may print as part of the page content. Because this is likely to be undesirable, it is recommended that PDF stamp annotations identifying logical substrates are placed in the gray area beside the page image. In this way, the annotations will never be printed regardless of the setting of the Annotations check box within the Adobe® Acrobat® print window. The rendered appearance of the PDF stamp annotation in Adobe® Acrobat® is not relevant to the NexPress front end; the NexPress front end only uses the PDF stamp annotation as it is placed in the PDF file.

The following table describes usage of the PDF annotation object, /**Annots,** inside a PDF file:

Name or Value	Manager	Worker	Description
/Type	**w**	**r**	Type: *name*
/*Annot*	**w**	**r**	
/Subtype	**w**	**r**	Type: *name* Annotation subtype.
/*Stamp*	**w**	**r**	Only the value of "*Stamp*" is supported.

Name or Value	Manager	Worker	Description
/Name	**w?**	**r?**	Type: *name* Identifies whether the page is to be rendered on Cover or Insert substrate. Only valid when the ***SubType*** key has a value of *Stamp*. If not present, and ***SubType*** has the value *Stamp*, the NexPress front end takes no action. An explicit value for the body substrate is not defined; body is the default in the absence of a PDF stamp annotation on a page.
/*SubstrateTypeCover*	**w?**	**r?**	The page is to be rendered on Cover stock.
/*SubstrateTypeInsert*	**w?**	**r?**	The page is to be rendered on Insert stock.
/Rect	**w**	**r**	Type: *rectangle* An array of numbers defining the lower left and upper right corner coordinates of the rectangular region of the page into which the annotation is rendered.

8 PPML/VDX Reference

The NexPress front end and NexPress workflow tools, such as the Imposition Viewer, support interpreting and processing of PPML/VDX-based data as defined by this document and supporting documents. Refer to [PPML/VDX] and [PPML/VDX-AN] for detailed information on the PPML syntax. Refer to [KNDIG] for detailed information on job submission methods for jobs containing PPML/VDX.

8.1 PPML/VDX Background

PPML/VDX, formally known as ANSI CGATS.20:2002 or ANSI PPML/VDX, is based on the PODi Personalized Print Markup Language (PPML) standard. PPML specifies the use of PDF-based data in conjunction with PPML data as a complete data definition for specifying page content in variable data printing (VDP) applications. The PPML/VDX standard was developed by the Committee for Graphic Arts Technologies Standards (CGATS) and is accredited by the American National Standards Institute (ANSI).

A NexPress legacy format, NexPress Portable Digital Master (PDM), is an earlier draft standard that became the ANSI CGATS.20-2002 standard in July of 2002. Although PDM is suitable for VDP applications, the use of PPML/VDX is recommended due to the workflow advantages its features enable. PDM has been deprecated; support may be removed in future releases of the NexPress front end. An ANSI PPML/VDX document should use PPML/VDX:2002 as the value for the GTS PPMLVDXVersion key in its Info dictionary.

8.1.1 PPML/VDX conformance levels

PPML/VDX specifies two conformance levels known as PPML/VDX-Strict and PPML/VDX-Relaxed.

- **PPML/VDX-Strict** requires all PDF files to conform to either the PDF/X-1a or PDF/X-3 standards, and requires the PPML data to be embedded within the PPML/VDX layout file. It is also requires that all **Binding** references in the **ContentBindingTable** specify MD5 checksums and unique IDs for all referenced PPML/VDX-Content files. The purpose of this conformance level is to allow the specifier of the PPML/VDX job to maximize control over the integrity of the PPML/VDX instance. This relieves the receiver of the exchanged data from most liability issues related to the correctness of the exchanged data. The receiver is guaranteed that all color data is identified and that the completeness of the files set can be verified.
- **PPML/VDX-Relaxed** relaxes many of the restrictions of PPML/VDX-Strict. For example, the PPML data may be specified in a separate XML data file, and the MD5 Checksum and unique ID in **Binding** references in the **ContentBindingTable** are optional. The use of PDF is not limited to PDF/X-1a and PDF/X-3; any PDF data can be used. The purpose of this conformance level is to enable the use of PPML/VDX data in trusted exchange scenarios, as well as within environments with more controlled and integrated authoring and print production workflows. The degree of integrity verification support depends upon the liability requirements of the VDP application, and how tightly integrated the workflow.

8.1.2 PPML/VDX exchange modes

The two conformance levels of PPML/VDX accommodate multiple exchange modes, including Single file PPML/VDX and Multiple-file PPML/VDX transfer.

- A **Single-file PPML/VDX** instance consists of the PPML/VDX-Layout file with the PPML data embedded. This is the most compact and portable form of a PPML/VDX instance; no external PDF file references need to be resolved to obtain dependent PPML/VDX-Content files or PPML data. The PDF pages used as page content referenced by **EXTERNAL_DATA_ARRAY** elements in the PPML data are completely contained within the PPML/VDX-Layout file itself. In accordance with the PPML/VDX Standard [PPML/VDX], the **ContentBindingTable** element must only contain a **Self** sub-element whose ***Src*** attribute value must exactly match that of the ***Src*** attribute of all PPML **EXTERNAL_DATA_ARRAY** elements.
- **A Multiple-file PPML/VDX** instance consists of a single PPML/VDX-Layout file and one or more additional PPML/VDX-Content files. The **ContentBindingTable** element of the PPML/VDX-Layout file contains a **Binding** sub-element for each unique PPML/VDX-Content file that is referenced by **EXTERNAL_DATA_ARRAY** elements in the PPML data. Within each **Binding** element, a ***Src*** attribute

contains a URL referencing a PPML/VDX-Content file. **The URL reference of the *Src* attribute must be resolveable in the receiving environment.** Specifically, the controller (such as the NexPress front-end) receiving the PPML/VDX-Layout file must have access to any referenced PPML/VDX content files. The ***Src*** value should not be black or have value of "SELF". Alternatively, ***LocalSrc*** can be used in **PPML/VDX-Relaxed** to define an alternate URL reference for ***Src***.

8.1.3 PPML/VDX Data Structure

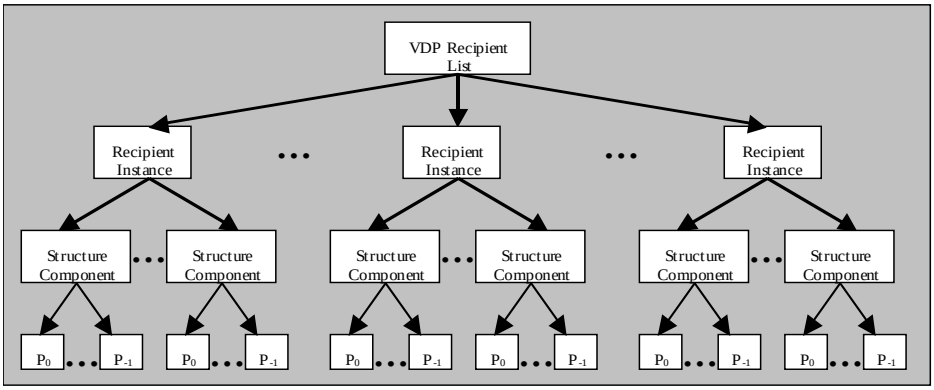
The ability to late-stage target a PPML/VDX job in a production workflow places additional requirements on the structure of the PPML/VDX data. These requirements include the need to guarantee page independence and efficiency of access to each PPML page definition and its associated PDL content data in order for it to be efficiently manipulated and reordered in preparation for print production. This is important because the order in which pages are utilized on a sequence of imposed sheets (printer spreads) is usually different from the reader order in which the PPML data specifies them.

VDP languages that combine layout information with actual content data cannot guarantee efficient parsing and manipulation as required by an imposition process. This is why embedding graphical content data within the PPML data is prohibited by the PPML/VDX standard. For similar reasons, PDF is used to specify graphical content data.

The PPML data of a PPML/VDX Instance defines a reader order set of Recipient Instances. A Recipient Instance can be thought of as the definition of one or more print products or components of print products to be manufactured. Each print product is customized for a single recipient, with no implied job ticket information

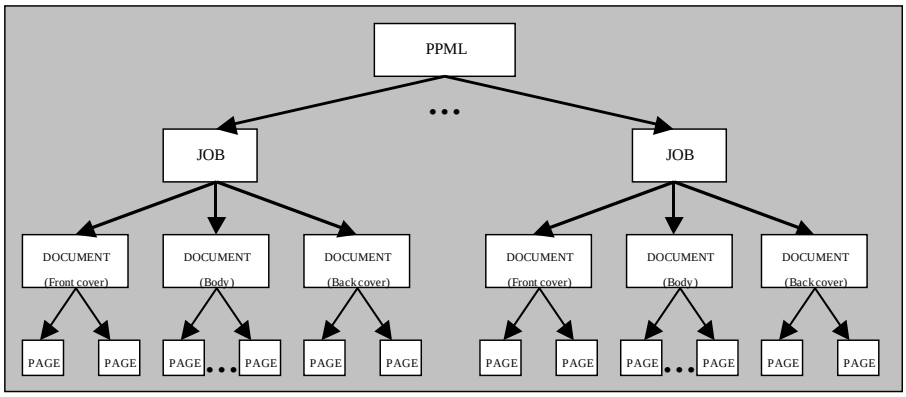
The term Recipient Instance, as it is used in this document, corresponds to the terms *record*, or *live* that are commonly used in VDP jargon today. A Recipient Instance therefore corresponds to the original recipient database record(s) used to drive the VDP composition engine in the creation of variable content pages described by the PPML/VDX data.

A PPML/VDX instance generated by a database-driven composition process can itself be thought of as a database of final form content pages. This PPML/VDX "database" of content pages is readily searchable and Recipient Instances and their structure components and pages can be efficiently located and selected. The PPML/VDX-Layout file must be structured in a way that allows different mediums, or print substrates, to be used in print product definitions. The semantic model is hierarchical for an individual Recipient Instance represented by the PPML data. The root of the hierarchy contains an ordered list of Recipient Instances; each Recipient Instance contains one or more sets of ordered pages in which each set of pages comprises the content of a structure component as shown in the diagram below:



A structure component must have the same page layout and medium (substrate) type. For example, if the front cover and body pages have the same layout but different media, at least two structure components are required: one for the front cover, and another for the body.

To convey the mapping of content pages for a PPML/VDX instance to finished page surfaces, the structure of the PPML data is organized according to this same structure. For example, if a book had front cover, body and back cover components, separate structure components must be specified within each recipient instance. Replacing "Recipient Instance" with "Job" and "Structure Component" with "Document", the resulting PPML hierarchy is represented in the following diagram:



8.2 Use of PDF cropping and trimming boxes in PPML/VDX

This section describes the rendering of PDF content pages in PPML/VDX by the NexPress. It explains how the PDF page boxes are interpreted and applied when rendered in the context of additional clipping boxes that may be present. The PDF content is referenced as compound elements in the PPML data.

- The value of the **/Page** dictionary's ***/MediaBox*** key specifies the maximum rectangular extent of content that may contain visible marks if a ***/CropBox*** key is not present. Content marks outside this region are clipped.
- In PPML/VDX, the content within the common rectangular region defined by the ***/CropBox*** and ***/MediaBox*** rectangles may be further manipulated and clipped by PPML **TRANSFORM** and **CLIP_RECT** elements.
- Regardless of the size and location of the ***/CropBox***, the lower left corner of the ***/MediaBox*** is referenced by the PPML **OBJECT** element's ***Position*** attribute.
- The NexPress front end will ignore ***/CropBox***, ***/BleedBox***, ***/TrimBox***, and ***/ArtBox*** if present in a PDF ***/Page*** dictionary.
- The rectangle defined by the PPML/VDX PAGE_DESIGN ***/TrimBox*** specifies the rectangular extent of the finished page and its content.
- To prevent the rendering of content marks defined outside of the ***/TrimBox*** rectangle without having to edit the PDF data, the application generating PPML/VDX should specify an appropriate PPML **CLIP_RECT** that prevents such marks from being imaged. This **CLIP_RECT** should be derived from the PDF page's ***/TrimBox***.
- The PPML CLIP_RECT should be extended to include the bleed area when bleed content marks are defined outside the PDF **/Page** dictionary's **/TrimBox** rectangle, and the object occurs on the PPML page defined by the ***PageDesign/@BleedBox*** attribute with bleed on one or more edges. The CLIP_RECT should be derived from the /BleedBox rather than the /TrimBox on those edges.
- If not specified, the ***/TrimBox*** defaults to the ***/CropBox***. If ***/CropBox*** is not specified, it defaults to ***/MediaBox***.

The PPML/VDX standard requires that the ***/TrimBox*** attribute within PPML **PAGE_DESIGN** specify the rectangular dimensions of PPML pages. Note that this PPML ***TrimBox*** attribute is different from the PDF ***/Page*** dictionary's ***/TrimBox***.

8.3 Specifying PPML/VDX

The following describes the adherence of PPML/VDX for the NexPress front end and NexPress workflow tools:

- Only PPML elements **PPML**, **PAGE_DESIGN**, **JOB**, **DOCUMENT** and **PAGE** are supported. Other PPML elements SHOULD NOT be used with a PPML/VDX job, and are ignored by NexPress products.
- Both PPML/VDX-Strict and PPML/VDX-Relaxed conformance levels are supported with the exception that only PDF content is consumed for PPML/VDX and not PDF/X as defined by [PPML/VDX]. Consequently, the NexPress front end does not interpret output rendering intents.
- Single-file PPML/VDX data is supported.
- Multiple-file PPML/VDX data is supported with the following exceptions:
  - Externalization of the PPML data from the PPML/VDX-Layout file through the use of the **PPMLRef** element is not supported.
  - The ***MD5_Checksum*** and ***UniqueID*** attributes of each **Binding** element are ignored.
  - Only the URL schemes *file:*, *ftp:*, and *http:* for the value of the ***Src*** attribute of **Binding** elements are supported for access to PPML/VDX-Content files.

    Example: For a PPML/VDX-Content file accessable in the receiver's environment: file:///Acme/VDPJobs/assets/promo/ppmlvdxContent/version1.3/logo.pdf

    Example: For a PPML/VDX-Content file in a different, or remote environment: "http://www.acme.com/VDPJobs/assets/promo/ppmlvdxContent/version1.3/logo.pdf".
  - All allowable URL schemes MUST be resolvable in the receiver's environment when needed.
- ***Binding/@LocalSrc*** is supported for entries in the **ContentBindingTable**. When supplied, ***LocalSrc*** will be resolved instead of ***Binding/@Src***.
- In the case of a multiple-file PPML/VDX instance, the PPML/VDX-Layout file's **ContentBindingTable** element is examined, and all referenced PPML/VDX-Content files identified in the **Binding** element entries are resolved and fetched. Processing of the PPML/VDX job will be aborted if any PPML/VDX-Content file reference cannot be resolved.

8.3.1 PPML element

The **PPML** element is the top level of the PPML data and encompasses all other PPML data.

- The **PPML** element MUST contain one **JOB** element for each Recipient Instance.

8.3.2 PAGE_DESIGN element

A **PAGE_DESIGN** element MUST be specified within the **PPML** element, ahead of any **JOB** sub-elements. **PAGE_DESIGN** values are inherited by lower level elements of the PPML hierarchy and do not need to be re-specified at lower levels unless the inherited values are overridden. To reduce redundant information and minimize the size of the PPML data, **PAGE_DESIGN** elements SHOULD NOT be restated within the scope of a **PAGE_DESIGN** with the same values.

8.3.3 JOB element

The **JOB** element is the container of all of the print product components and their pages particular to the finished print product specified for a single recipient.

- **JOB** elements MUST contain at least one **DOCUMENT** element.
- If multiple **DOCUMENT** elements occur in a **JOB** element and all specify pages that belong to the same bound print product, those **DOCUMENT** elements MUST be specified in reader order.

NOTE: This requirement allows simple PPML/VDX reader viewers to display the ordered sequence of related pages without processing job ticket information to obtain the reader order of the **DOCUMENT** element components.

EXAMPLE: If a **JOB** element contains three **DOCUMENT** sub-elements in the definition of a single finished print product, one **DOCUMENT** sub-element each for Front Cover, Body, and Back Cover components respectively, the order of those **DOCUMENT** elements within the **JOB** element must match that reader order. Refer to the complete example below.

- The ***Label*** attribute for the **JOB** element MUST be present with a string value that is unique relative to the value of the ***Label*** attributes for all other **JOB** elements of the containing **PPML** element.

  NOTE: The ***Label*** attribute of the **JOB** element is used by various processes in the production workflow to uniquely identify it for selection. Among other uses, named identification of Recipient Instances facilitates selective reprint of spoiled sheets. Its value may also match the primary key of the recipient database for accessing recipient record information. Such information as the recipient's gender, name and address can then be associated with the content pages.

- The value of the **JOB**/@***Label*** attribute MUST be an *NMTOKEN* and thus not contain white space characters.

8.3.4 DOCUMENT element

The **DOCUMENT** element is used to specify a reader ordered list of content pages specified as **PAGE** elements belonging to a single structure component.

- The **DOCUMENT** element MUST contain a reader order list of one or more **PAGE** sub-elements as required by [PPML].
- The orientation of each page MUST be consistent with the aspect ratio of its dimensions as specified by the **DOCUMENT** element. The page orientation must match that of the finished print product.
- The trim and bleed boxes specified by **PAGE_DESIGN**/@***TrimBox*** and **PAGE_DESIGN**/@***BleedBox*** MUST be homogeneous for all pages defined by the **DOCUMENT** element.
- The pages specified by a **DOCUMENT** element MUST have common product intent semantics.

  NOTE: Even though the value of the **DOCUMENT**/@***Label*** attribute may be descriptive and convey the meaning of the component, its use as the component's identifier requires that it be unique within the containing **DOCUMENT** element. Additional component characteristics including media selection are specified in the JDF Product Intent data or in the accompanying JDF Job Ticket. The value of **DOCUMENT**/@***Label*** attribute must match that of a ***RunTags*** partition key in JDF media resources.

- The **DOCUMENT** element's ***Label*** attribute is an NMTOKEN and MUST be present. As such it cannot contain white space characters
- The value of the ***Label*** attribute of the **DOCUMENT** element, also referred to as the component name, MUST be unique among component names of all other **DOCUMENT** elements in the same containing **JOB** element.
- The ***Label*** attributes occurring in different **JOB** elements SHOULD have the same value for any **DOCUMENT** elements with equivalent product intent and print product component semantics.

  EXAMPLE: If all recipients receive a wire comb bound book where the body components of all books require the same substrate (**MediaIntent**), page layout (**LayoutIntent**), and binding style (**BindingIntent**), all body components are homogeneous. In this case, the value of the **DOCUMENT** element ***Label*** attributes for each Recipient Instance should be the same (e.g. **DOCUMENT**/@***Label***="*Body*").

8.3.5 PAGE element

The following list specifies the technical requirements for the use of the **PAGE** element:

- Binding style and media usage for all pages MUST be explicitly specified.

NOTE: The PPML/VDX standard allows the use of JDF product intent in the definition of finished pages in which blank pages are implied based on media usage in the context of a binding style. The NexPress implementation of PPML/VDX requires that blank pages be included as **<PAGE/>** elements so that that the exact number of pages is specified for binding style and media usage. When specifying blank pages, care should be taken to not lay down a blank page the size of the TrimBox, as this could result in significantly greater memory usage than required.

9 Device Capabilities File Reference

This chapter identifies NexPress support for its Device Capabilities File as referenced in the [KNDIG]. The NexPress maintains a single file to describe system capabilities. This file, DevCaps.xml, is written to \\\CDFExchg\DevCaps.xml on the DFE Server. It is also available as http://<SERVERNAME>/cdfeUisAccess/DevCaps.xml. Usage of this Device Capabilities File is provided in [KNDIG] and not here.

9.1 Understanding the Device Capabilities File

The Device Capabilities File is written in XML; the legal elements and their structure are defined by a Document Type Definition (DTD). When reading the DTD, it is helpful to understand a few basic rules of syntax. Refer to Section 9.3 for the complete NexPress DevCaps.xml DTD.

- The DTD defines the structure of an XML file. Each element may contain other elements, and be characterized by one or more attributes. The Device Capabilities File often uses attributes to specify the units of an element, and to define an enumeration list of possible values. Attributes are identified as #IMPLIED or #REQUIRED indicating whether the attribute may be omitted from the element definition in the XML file.
- The relationship between elements is defined using a regular expression grammar. Each element is specified as a sequence of elements in which '*' denotes "zero or more", '+' denotes "one or more", '?' denotes "zero or one", and the absence of a modifier denotes "precisely one". Root elements must be defined in the DTD before any sub-elements that reference them. Each element eventually resolves to either #PCDATA, #CDATA, or EMPTY representing parsed or non-parsed character data (text), or an empty element, respectively. When an element is defined as EMPTY, its value is described by the containing attribute.
- In the DTD, an element is defined as <!ELEMENT ElementName (ElementSequence)>. "ElementName" specifies a new element name. "ElementSequence" is a list of other elements comprising this new ElementName; it can also be #PCDATA, #CDATA, or EMPTY as appropriate.
- In the DTD, an attribute corresponds to a defined ElementName; each element can have any number of possible attributes. The attribute is defined as <!ATTLIST ElementName AttributeName (AttributeDefinition) RequiredTag>. "ElementName" specifies the element corresponding to this attribute. "AttributeName" is the identifying text string. In the Device Capabilities File, "AttributeDefinition" is usually an enumeration of one or more text strings, but can also be defined as CDATA if the attribute could contain a generic string. "RequiredTag" indicates whether the attribute is optional.
- The DTD indicates the possible content of the XML file it describes. The presence of #IMPLIED attribute tags, and "*" or "?" on element sequence definitions, permits variability in the actual content and form of the Device Capabilities File. In practice much less variance is observed in the content of the Device Capabilities File than the DTD allows.

9.2 Device Capabilities Elements

The NexPress front end generates a DevCaps.xml that contains only the keys, values and sub-elements identified below.

9.2.1 BayID

BayID defines the physical supply bay on the NexPress digital production color press identified by the containing **SubstrateSupply**. The association of a physical supply location to **BayID** is controlled by the print engine. The assigned value is provided as an attribute.

- **BayID** occurs as a sub-element of **SubstrateSupply**.

Attribute Name or value	Writer	Description

Attribute Name or value	Writer	Description
Enum	***w***	Type: *Enumeration* Defines the physical supply bay on the NexPress digital production color press identified by the containing **SubstrateSupply**.
bayA	***w?***	
bayB	***w?***	
bayC	***w?***	
bayD	***w?***	
bayE	***w?***	
bayF	***w?***	
bayG	***w?***	
bayH	***w?***	
ClassicMain	***w?***	Main Supply of a Classic NexPress 2100
ClassicUpper	***w?***	Upper Supply of a Classic NexPress 2100
ClassicLower	***w?***	Lower Supply of a Classic NexPress 2100
InserterUpper	***w?***	
InserterLower	***w?***	
Unknown	***w?***	

DTD Representation:

```
<!ELEMENT BayID EMPTY>
<!ATTLIST BayID Enum (bayA|bayB|bayC|bayD|bayE|bayF|bayG|bayH|ClassicMain|ClassicUpper|ClassicLower|
                      InserterUpper|InserterLower|Unknown) #REQUIRED>
```

9.2.2 BoundedMediumData

BoundedMediumData defines the characteristics of a substrate medium and their permitted value ranges.

- **BoundedMediumData** may occur as a sub-element of **SystemMedium**, **MinMediumData**, and **MaxMediumData**.
- **BoundedMediumData** may contain sub-elements **Length**, **Width**, **MediumWeight**, **Transparency**, **Thickness**, and **Creep**.

DTD Representation:

```
<!ELEMENT BoundedMediumData (Length?, Width?, SubstrateSizeName?, MediumWeight?, Transparency?,
    Thickness?, Creep?)>
```

9.2.3 Capacity

Capacity defines the sheet capacity of the substrate tray identified by the containing **SubstrateSupply**. Capacity is specified as an integer quantity of sheets. The basis weight of the substrate used in defining capacity is print engine dependent, and not specified by this interface.

- **Capacity** occurs as a sub-element of **SubstrateSupply**.

DTD Representation:

```
<!ELEMENT Capacity (#PCDATA)>
```

9.2.4 CertifiedMedium

CertifiedMedium defines the certification status of the containing **PrinterMedium**. The values defining characteristics of a certified medium have been validated by NexPress; the medium has been certified for use in the NexPress digital production color press.

- **CertifiedMedium** may occur as a sub-element of the **PrinterMedium** element.

NOTE: When **CertifiedMedium** is defined for a **PrinterMedium**, the medium is certified. In practice the Device Capabilities File does not specify an attribute value for ***Present***, so the default value of *True* applies.

Attribute Name or value	Writer	Description
Present = True	***w?***	Type: *Boolean* Defines the certification status of the containing **PrinterMedium**.
True	***w?***	Medium is certified for use on the NexPress digital production color press. Default Value.
False	***w?***	Medium is not certified for use on the NexPress digital production color press.

DTD Representation:

```
<!ELEMENT CertifiedMedium EMPTY>
    <!ATTLIST CertifiedMedium
        Present (True | False) #IMPLIED>
```

9.2.5 CMYKGraphicICCProfile

CMYKGraphicICCProfile defines the name of an input ICC profile suitable for use with four component PDF graphics and text data that are specified using the **DeviceCMYK** color space. This name should correspond to an entry in **ICCProfileList**. Its value is a character string.

- **CMYKGraphicICCProfile** occurs as a sub-element of **SystemDefaults**.

DTD Representation:

```
<!ELEMENT CMYKGraphicICCProfile (#PCDATA)>
```

9.2.6 CMYKImageICCProfile

CMYKImageICCProfile element defines the name of an input ICC profile suitable for use with four component PDF image data that are specified using the **DeviceCMYK** color space. This name should correspond to an entry in **ICCProfileList**. Its value is a character string.

- **CMYKImageICCProfile** occurs as a sub-element of **SystemDefaults**.

DTD Representation:

```
<!ELEMENT CMYKImageICCProfile (#PCDATA)>
```

9.2.7 Collate

Collate defines the supported job collate options. Available collate options and their values are defined as attributes.

- **Collate** may occur as a sub-element of **Collation**.

Attribute Name or value	Writer	Description
Enable = True	***w?***	Type: *Boolean* Defines the supported job collate options.

Attribute Name or value	Writer	Description
True	***w?***	Printing system supports enabling collation. Default Value
False	***w?***	Printing system supports disabling collation.

DTD Representation:

```
<!ELEMENT Collate EMPTY>
    <!ATTLIST Collate
        Enable (True | False) #IMPLIED>
```

9.2.8 Collation

Collation defines a list of collate options that are available to a print job. When collation is enabled, a document remains a single set during printing; the number of sets produced is the requested count. When collation is disabled, the requested count is applied to each page and printed as a set; the number of sets produced is the number of pages in the document.

- **Collation** may occur as a sub-element of **FinishingOptions**.
- **Collation** may contain the sub-element **Collate**.

DTD Representation:

```
<!ELEMENT Collation (Collate*)>
```

9.2.9 ColorSpace

ColorSpace defines a color space. A color space is characterized by an ICC profile. Its value is supplied as an attribute.

- **ColorSpace** may occur as a sub-element of **ICCProfile** and **ColorSpaceList**.

Attribute Name or value	Writer	Description
Enum = Gray	***w?***	Type: *Enumeration* The name of an input or output color-space as characterized by an ICC profile.
Gray	***w?***	Default Value
CMYK	***w?***	
RGB	***w?***	
LAB	***w?***	
XYZ	***w?***	
LUV	***w?***	
YXY	***w?***	
3CLR	***w?***	
4CLR	***w?***	
5CLR	***w?***	
Present	***!w***	Type: *Boolean* Not supported

DTD Representation:

```
<!ELEMENT ColorSpace EMPTY>
    <!ATTLIST ColorSpace
```

Enum (Gray | CMYK | RGB | LAB | XYZ | LUV | YXY | 3CLR | 4CLR | 5CLR) #IMPLIED
Present (True | False) #IMPLIED>

9.2.10 ColorList

ColorList enumerates the set of substrate color types supported by the substrate catalog.

- **ColorList** may occur as a sub-element of **MediumConstraints**.
- **ColorList** may contain sub-elements of **ColorType**.

DTD Representation:

<!ELEMENT ColorSpaceList (ColorSpace*)>

9.2.11 ColorSpaceList (Obsolete)

ColorSpaceList defines the set of color spaces used by the printer.

NOTE: Use of **ColorSpaceList** is obsolete. It is included in the Device Capabilities File, but the contents of **ColorSpaceList** do not convey meaningful device capability information.

- **ColorSpaceList** may occur as a sub-element of **Printer**.
- **ColorSpaceList** may contain the sub-element **ColorSpace**.

DTD Representation:

<!ELEMENT ColorSpaceList (ColorSpace*)>

9.2.12 ColorType

ColorType defines the substrate color.

- **ColorType** may occur as a sub-element of **ColorList** and **SystemMedium**.

Attribute Name or value	Writer	Description
Enum = White	***w?***	Type: *Enumeration* Color of the substrate.
Unknown	***w?***	
WhiteNeutral	***w?***	
GrayNeutral	***w?***	
Yellowish	***w?***	
Blueish	***w?***	
Reddish	***w?***	
Greenish	***w?***	
White	***w?***	Default Value
Blue	***w?***	
Cream	***w?***	
BrightYellow	***w?***	
Gray	***w?***	
Green	***w?***	
Ivory	***w?***	

Orange	***w?***	
Pink	***w?***	
Red	***w?***	
Yellow	***w?***	
Transparency	***w?***	
Other	***w?***	
Present	***!w***	Type: *Boolean* Not supported

DTD Representation:

```
<!ELEMENT ColorType EMPTY>
    <!ATTLIST ColorType
        Enum (Unknown | WhiteNeutral | GrayNeutral | Yellowish | Blueish | Reddish | Greenish | White |
            Blue | Cream | BrightYellow | Gray | Green | Ivory | Orange | Pink | Red | Yellow | Transparency |
            Other) #IMPLIED
        Present (True | False) #IMPLIED>
```

9.2.13 Comment

Comment contains a human-readable comment. **Comment** is usually free-form descriptive text that provides a more detailed explanation of the containing element. Its value is an ASCII string.

- **Comment** may occur as a sub-element of **Printer**, **SystemMedium**, **Font**, and **ImpositionTemplate**.

DTD Representation:

```
<!ELEMENT Comment (#PCDATA)>
```

9.2.14 CoreFont

CoreFont defines whether the font is a member of the standard set of Adobe® core fonts.

- **CoreFont** may occur as a sub-element of **Font**.

NOTE: Defining the **CoreFont** element is sufficient to designate a core font. In practice the Device Capabilities File does not specify an attribute value for ***Present***, so the default value of *True* applies.

Attribute Name or value	Writer	Description
Present = True	***w?***	Type: *CDATA* Defines whether the font is a member of the standard set of Adobe core fonts.
True	***w?***	Font is a core font. Default Value.
False	***w?***	Font is not a core font.

DTD Representation:

```
<!ELEMENT CoreFont EMPTY>
 <!ATTLIST CoreFont
    Present (True | False) #IMPLIED>
```

9.2.15 CoverList

CoverList defines a list of cover mode media mapping options that are available to a print job.

- **CoverList** may occur as a sub-element of **FinishingOptions**.

- **CoverList** may contain the sub-element **CoverType**.

DTD Representation:

```
<!ELEMENT CoverList (CoverType*)>
```

9.2.16 CoverType

CoverType defines a type of cover that can be produced in the printed document. Available types of covers are defined as attributes.

- **CoverType** may occur as a sub-element of **CoverList**.

Attribute Name or value	Writer	Description
Enum = None	***w?***	Type: *Enumeration* Defines type of cover that can be produced in the printed document.
None	***w?***	The print product has no cover. Default Value.
Front	***w?***	The print product only has a front cover.
Back	***w?***	The print product only has a back cover.
Both	***w?***	The print product has both a front and a back cover.
Present	***!w***	Type: *Boolean* Not supported.

DTD Representation:

```
<!ELEMENT CoverType EMPTY>
    <!ATTLIST CoverType
        Enum (None | Front | Back | Both) #IMPLIED
        Present (True | False) #IMPLIED>
```

9.2.17 Creep

Creep defines the creep value of a substrate as it pertains to perceived image movement during gather/fold finishing. **Creep** can be in units of millimeters or inches; units are provided as an attribute.

- **Creep** may occur as a sub-element of **BoundedMediumData**.

Attribute Name or value	Writer	Description
Unit = UM	***w?***	Type: *Enumeration* Unit of measure for **Creep**.
UM	***w?***	Micrometers. Default Value
IN	***w?***	Inches.

DTD Representation:

```
<!ELEMENT Creep (#PCDATA)>
    <!ATTLIST Creep
        Unit (UM|IN) #IMPLIED>
```

9.2.18 CurrentColorMode (Obsolete)

CurrentColorMode defines the current color policy of the printing system. Its value is provided as an attribute.

- **CurrentColorMode** may occur as a sub-element of **FifthColor**.

NOTE: **CurrentColorMode** is obsolete. Historically it was used to report the color policy in use by the printer because changing from four to five color printing required a significant changeover interval. The press now operates in an *Auto* color mode using four or five dry-ink printing stations as appropriate.

Attribute Name or value	Writer	Description
Enum = Auto	***w?***	Type: *Enumeration* Defines the current color policy of the printing system.
Auto	***w?***	The current color mode is set to automatic. Default Value.
Black	***w?***	The current color mode of operation is black printing only.
CMYK	***w?***	The current color mode of operation is 4-color CMYK.
5CLR	***w?***	The current color mode 5 color printing – CMYK and a fifth process colorant.
Unknown	***w?***	The current color mode is undefined.

DTD Representation:

```
<!ELEMENT CurrentColorMode EMPTY>
    <!ATTLIST CurrentColorMode
        Enum (Auto | Black | CMYK | 5CLR | Unknown) #IMPLIED>
```

9.2.19 CurrentLoadedFifthColor

CurrentLoadedFifthColor identifies the fifth dry-ink printing station installed and loaded in the printing system. The color must be a member of the **InstalledFifthColorList** for it to be loaded.

- **CurrentLoadedFifthColor** may occur as a sub-element of **FifthColor**.
- **CurrentLoadedFifthColor** may contain the sub-element **FifthColorName**.

DTD Representation:

```
<!ELEMENT CurrentLoadedFifthColor (FifthColorName, SupportedEngineSpeedList)>
```

9.2.20 DefaultDestination

When **DefaultDestination** is defined for a **Delivery**, the **DeliveryLocation** is the device default.

- **DefaultDestination** may occur as a sub-element of **Delivery**.

9.2.21 Delivery

Delivery defines a set of print output destinations available to the printing system and their capabilities. Each **Delivery** element has a unique identifier, ID, provided as an attribute.

- **Delivery** may occur as a sub-element of **SubstrateDeliveryList**.
- **Delivery** contains the sub-elements **DeliveryType**, **OffsetCapable**, **StapleCapable**, **OperatorSelectable**, **OutputDelivery**, **MinSize**, **MaxSize, SupportedEngineSpeedList, DefaultDestination**.

Attribute Name or value	Writer	Description
ID	***w***	Type: *CDATA* Unique identifier for delivery element. The format of ID is not defined by this interface. Hex string values (such as 0x11 and 0x41) are typical, but may change in the future.

DTD Representation:

```
<!ELEMENT Delivery (DeliveryType, OffsetCapable, StapleCapable, OperatorSelectable, OutputDelivery, MinSize,
    MaxSize, SupportedEngineSpeedList, DefaultDestination)>
    <!ATTLIST Delivery
        ID CDATA #REQUIRED>
```

9.2.22 DeliveryLocation

DeliveryLocation defines an enumeration of all output destinations across the NexPress product line.

- **Delivery** may occur as a sub-element of **SubstrateDeliveryList**.

DTD Representation:

```
<!ELEMENT DeliveryLocation EMPTY>
    <!ATTLIST DeliveryLocation Enum (Main | Main2 | Proof | Proof2 | InlineFinisher | FinisherUpper |
    FinisherLower | FinisherSaddle | Hidden | Unknown) #REQUIRED>
```

9.2.23 DeliveryType

DeliveryType defines a substrate output location. The location can be a tray or bin on the printer or output accessory.

- **DeliveryType** may occur as a sub-element of **Delivery**.
- **DeliveryType** contains the sub-element **DeliveryTypeDetails**.

Attribute Name or value	Writer	Description
Enum	***w***	Type: *Enumeration* Defines a substrate output location.
stacker	***w?***	High capacity output stacker.
proof	***w?***	Proof output tray.
waste	***w?***	Output tray for discarding spoiled sheets.
finisher	***w?***	Finisher device.
unknown	***w?***	Unknown device.

DTD Representation:

```
<!ELEMENT DeliveryType (DeliveryTypeDetails)>
    <!ATTLIST DeliveryType
        Enum (stacker | proof | waste | finisher | unknown) #REQUIRED>
```

9.2.24 DeliveryTypeDetails

DeliveryTypeDetails provides more specific information about the **DeliveryType** location; such information is useful for differentiating between multiple locations on the output device. The value of **DeliveryTypeDetails** is provided as an attribute.

- **DeliveryTypeDetails** may occur as a sub-element of **DeliveryType**.
- **DeliveryTypeDetails** contains the sub-element **SupportedEngineSpeedList.**

Attribute Name or value	Writer	Description
Enum	***w***	Type: *Enumeration* Provides detailed **DeliveryType** capabilities.
Deposit	***w?***	Has no special capability.

Attribute Name or value	Writer	Description
Rotary	***w?***	Has rotary capability.
HighVolumeTray	***w?***	Proof output tray has high capacity (in terms of quantity of sheets it can hold).
LowCapacityProofWaste	***w?***	Proof and waste output trays have low capacity (in terms of quantity of sheets it can hold).
LowCapacityProof	***w?***	Proof output tray has low capacity (in terms of quantity of sheets it can hold).
Vertical	***w?***	Has vertical stacking capability.
Dfa	***w?***	Inline finisher that uses the DFA interface.
HighCapacity	***w?***	High Capacity Stacker (NexPress M700 only)
ProofWaste	***w?***	Proof Tray (NexPress M700 only)
Saddle	***w?***	
Invalid	***w?***	
Unknown	***w?***	
Position	***w***	Type: *CDATA* A numerical assignment designated by the print engine. Position assignment is not defined by this interface.

DTD Representation:

```
<!ELEMENT DeliveryTypeDetails (SupportedEngineSpeedList?)>
    <!ATTLIST DeliveryTypeDetails
        Enum (Deposit | Rotary | HighVolumeTray | LowCapacityProofWaste | LowCapacityProof | Vertical | Dfa |
            HighCapacity | ProofWaste | Saddle | Invalid | Unknown) #REQUIRED
        Position CDATA #REQUIRED>
```

9.2.25 DeviceID

DeviceID identifies the NexPress front end from which the Device Capabilities File was obtained. It is the name of the NexPress front end on a computer network domain. Its value is an ASCII string.

- **DeviceID** may occur as a sub-element of **ManufacturingCapabilities**.

DTD Representation:

```
<!ELEMENT DeviceID (#PCDATA)>
```

9.2.26 DiscreteSize

DiscreteSize describes the size constraint (Width, Height) of a medium element.

- **DiscreteSize** occurs as a sub-element of **SupportedSizeList**.

DTD Representation:

```
<!ELEMENT DiscreteSize (Width, Height)>
<!ATTLIST DiscreteSize Unit (MM) #IMPLIED>
```

9.2.27 DoorNumber

DoorNumber defines the door number assigned to the containing **SubstrateSupply**. Assignment of a value to **DoorNumber** is print engine-dependent and not specified by this interface; the value is an integer.

- **DoorNumber** occurs as a sub-element of **SubstrateSupply**.

DTD Representation:

```
<!ELEMENT DoorNumber (#PCDATA)>
```

9.2.28 EngineSpeed

EngineSpeed identifies the speed in A4 pages/minute which the NexPress can run.

- **EngineSpeed** may occur as a sub-element of **SupportedEngineSpeedList, MaxLicensedEngineSpeed**.

Attribute Name or value	Writer	Description
Enum	***w***	Type: *Enumeration* Engine printing speed in A4 pages/minute
70	***w?***	
83	***w?***	
100	***w?***	
120	***w?***	
Unknown	***w?***	

DTD Representation:

```
<!ELEMENT EngineSpeed EMPTY>
    <!ATTLIST EngineSpeed
        Enum (70 | 83 | 100 | 120 | Unknown) #REQUIRED>
```

9.2.29 ExternalControllerType

ExternalControllerType defines the type of DFE attached to the printing station.

- **ExternalControllerType** may occur as a sub-element of **Printer**.

Attribute Name or value	Writer	Description
Enum = None	***w***	Type: *Enumeration* Defines type of the external controller.
NONE	***w?***	Default Value.
FIERY	***w?***	
BRISQUE	***w?***	
CREO_PODS	***w?***	

DTD Representation:

```
<!ELEMENT ExternalControllerType EMPTY>
    <!ATTLIST ExternalControllerType
        Enum (NONE | FIERY | BRISQUE | CREO_PODS) #REQUIRED>
```

9.2.30 FeedEdge

FeedEdge identifies the leading edge of the substrate as it travels through the paper path.

- **FeedEdge** may occur as a sub-element of **PrinterMedium**.

DTD Representation:

<!ELEMENT FeedEdge EMPTY>

<!ATTLIST FeedEdge Enum (ShortEdge|LongEdge) #IMPLIED>

9.2.31 FeederType

FeederType identifies the type of substrate supply.

- **FeederType** may occur as a sub-element of **SubstrateSupply**.

Attribute Name or value	Writer	Description
Enum	***w?***	Type: *Enumeration* No Default
1K_Drawer	***w?***	1000 sheet supply drawer
4.5K_Drawer	***w?***	4500 sheet supply drawer
RollFeed	***w?***	Roll Feeder
Unknown	***w?***	

DTD Representation:

```
<!ELEMENT FeederType EMPTY>
    <!ATTLIST FeederType
        Enum (1K_Drawer | 4.5K_Drawer | RollFeed | Unknown) #IMPLIED
```

9.2.32 FifthColor

FifthColor defines the printing system's support for fifth color dry-ink printing stations. Support for a fifth dry-ink printing station is reported as an attribute.

- **FifthColor** may occur as a sub-element of Printer.
- **FifthColor** may contain sub-elements **InstalledFifthColorList**, **CurrentColorMode**, and **CurrentLoadedFifthColor**.

Attribute Name or value	Writer	Description
Supported = True	***w?***	Type: *Boolean* Defines support for a fifth dry-ink printing station.
True	***w?***	Printing system supports a fifth, dry-ink printing station. Default Value.
False	***w?***	Printing system does not support a fifth, dry-ink printing station.

DTD Representation:

```
<!ELEMENT FifthColor (InstalledFifthColorList*, CurrentColorMode?, CurrentLoadedFifthColor?)>
    <!ATTLIST FifthColor
        Supported (True | False) #IMPLIED>
```

9.2.33 FifthColorName

FifthColorName defines the name of a fifth dry-ink printing station. The name is an attribute.

- **FifthColorName** may occur as a sub-element of **InstalledFifthColorList** and **CurrentLoadedFifthColor**.

- **FifthColorName** may contain sub-element **SecuredColorant**.

Attribute Name or value	Writer	Description
Enum	***w?***	Type: *Enumeration* Defines name of a fifth dry-ink printing station. Enumeration values are system-dependent.

DTD Representation:

```
<!ELEMENT FifthColorName (SecuredColorant?)>
    <!ATTLIST FifthColorName
        Enum (#PCDATA) #IMPLIED>
```

9.2.34 FinishingOptions

FinishingOptions defines supported finishing available on the printing system. Finishing options pertain to the handling of printed output.

- **FinishingOptions** may occur as a sub-element of **ManufacturingCapabilities**.
- **FinishingOptions** may contain sub-elements **CoverList**, **InsertList**, **Collation**, **Jogging**, **SubstrateDeliveryList**, and **PageOrderList**.

DTD Representation:

```
<!ELEMENT FinishingOptions (CoverList?, InsertList?, Collation?, Jogging?, SubstrateDeliveryList?,
    PageOrderList?)>
```

9.2.35 Font

Font defines the characteristics of a font installed on the printing system.

- **Font** may occur as a sub-element of **FontList**.
- **Font** contains sub-element **Name**. It may also contain sub-elements **Comment**, **FontType**, and **CoreFont**.

DTD Representation:

```
<!ELEMENT Font (Name, Comment?, FontType?, CoreFont?)>
```

9.2.36 FontDefaults

FontDefaults defines default values for all font characteristics in the containing **FontList**. The default is only used if the defaulted element is not explicitly defined in **Font**.

- **FontDefaults** may occur as a sub-element of **FontList**.
- **FontDefaults** contains sub-element **Font**.

NOTE: **FontDefaults** does not define a default value for **Name**. Each **Font** element within **FontList** must have a unique name.

DTD Representation:

```
<!ELEMENT FontDefaults (Font)>
```

9.2.37 FontList

FontList defines the list of installed fonts. These fonts are available to any submitted JDF ticket or Virtual Printer Hot Folder. Each listed font defines only those elements for which the value differs from that in **FontDefaults**.

- **FontList** may only occur as a sub-element of **ManufacturingCapabilities**.
- **FontList** may contain sub-elements **FontDefaults** and **Font**.

DTD Representation:

<!ELEMENT FontList (FontDefaults?, Font*)>

9.2.38 FontType

FontType defines the data format of a font. Its value is supplied as an attribute.

- **FontType** may occur as a sub-element of **Font**.

Attribute Name or value	Writer	Description
Enum = PostScriptType1	***w?***	Type: *Enumeration* Defines the data format of a font.
PostScriptType1	***w?***	Default Value.
PostScriptType1MM	***w?***	
PostScriptType3	***w?***	
PostScriptType42	***w?***	
PostScriptTTF	***w?***	
PostScriptType0CID	***w?***	
Present	***!w***	Type: *Boolean* Not supported.

DTD Representation:

```
<!ELEMENT FontType EMPTY>
    <!ATTLIST FontType
        Enum (PostScriptType1 | PostScriptType1MM | PostScriptType3 | PostScriptType42 |
            PostScriptTTF | PostScriptType0CID) #IMPLIED
        Present (True | False) #IMPLIED>
```

9.2.39 GlossUnitCompatible

GlossUnitCompatible indicates containing **PrinterMedium** is compatible with the external glosser.

- **GlossUnitCompatible** may occur as a sub-element of **PrinterMedium**.

DTD Representation:

```
<!ELEMENT GlossUnitCompatible EMPTY>
```

9.2.40 HasGlosserSettings

HasGlosserSettings indicates glosser settings are defined for containing **PrinterMedium**.

- **HasGlosserSettings** may occur as a sub-element of **PrinterMedium**.

DTD Representation:

```
<!ELEMENT HasGlosserSettings EMPTY>
```

9.2.41 Height

Height defines the height/length component of a rectangular dimension. **Height** can be in units of millimeters or inches.

- **Height** may occur as a sub-element of **MinSize** and **MaxSize**.

Attribute Name or value	Writer	Description

Attribute Name or value	Writer	Description
Unit = MM	***w?***	Type: *Enumeration* Unit of measure for **Width**.
MM	***w?***	Millimeters. Default Value
IN	***w?***	Inches.

DTD Representation:

```
<!ELEMENT Height (#PCDATA)>
```

9.2.42 I2PPlatform

I2PPlatform reports the I2P platform type used within the printing system.

- **I2PPlatform** may occur as a sub-element of **Printer**.

Attribute Name or value	Writer	Description
Enum	***w***	Type: *Enumeration* Printing system platform type.
I2P	***w?***	First generation I2P.
I2P2	***w?***	Second generation I2P.
Unknown	***w?***	Unknown.

DTD Representation:

```
<!ELEMENT I2PPlatform EMPTY>
    <!ATTLIST I2PPlatform
        Enum (I2P | I2P2 | Unknown) #REQUIRED>
```

9.2.43 ICCProfile

ICCProfile defines the name and color space definition of an ICC profile. An attribute identifies those profiles mapped to a substrate.

- **ICCProfile** may occur as a sub-element of **ICCProfileList**.
- **ICCProfile** contains sub-element **Name**. It may also contain sub-element **ColorSpace**.

Attribute Name or value	Writer	Description
SubstrateICCProfile = True	***w?***	Type: *Boolean* Indicates whether or not the ICC profile is an output ICC profile characterizing a supported medium from a substrate entry of the Media Catalog.
True	***w?***	The ICC profile is and output ICC profile and is referenced from a Media Catalog entry. Default Value.
False	***w?***	The ICC profile is not an output pro-file and is not referenced from a Media Catalog entry.

DTD Representation:

```
<!ELEMENT ICCProfile (Name, ColorSpace?)>
    <!ATTLIST ICCProfile
        SubstrateICCProfile (True | False) #IMPLIED>
```

9.2.44 ICCProfileDefaults

ICCProfileDefaults defines default values for all ICC profile characteristics in the containing **ICCProfileList**. The default is only used if the defaulted element is not explicitly defined in **ICCProfile**.

- **ICCProfileDefaults** may occur as a sub-element of **ICCProfileList**.
- **ICCProfileDefaults** contains the sub-element **ICCProfile**.

NOTE: **ICCProfileDefaults** does not define a default value for **Name**. Each **ICCProfile** element within **ICCProfileList** must have a unique name.

DTD Representation:

```
<!ELEMENT ICCProfileDefaults (ICCProfile)>
```

9.2.45 ICCProfileList

ICCProfileList defines ICC profile resources available on the printing system. All ICC profiles in this list are available to any Virtual Printer Hot Folder. Each listed **ICCProfile** defines only those elements for which the value differs from that in **ICCProfileDefaults**.

- **ICCProfileList** may occur as a sub-element of **ManufacturingCapabilities**.
- **ICCProfileList** contains the sub-element **ICCProfileDefaults**. It may also contain the sub-element **ICCProfile**.

DTD Representation:

```
<!ELEMENT ICCProfileList (ICCProfileDefaults, ICCProfile*)>
```

9.2.46 ImpositionTemplate

ImpositionTemplate defines properties of an imposition template.

- **ImpositionTemplate** occurs as a sub-element of **ImpositionTemplateDefaults**. It may also occur as a sub-element of **ImpositionTemplateList**.
- **ImpositionTemplate** contains the sub-element **Name**. It may also contain sub-elements **Comment**, **PagesPerSheet**, **SheetsPerSignature**, and **LayoutMode**.

NOTE: **Name** identifies the imposition template resource stored on the NexPress front end. This name was provided during template installation. The NexPress front end's installation includes a set of predefined imposition templates that are stored as system resources.

DTD Representation:

```
<!ELEMENT ImpositionTemplate (Name, Comment?, PagesPerSheet?, SheetsPerSignature?, LayoutMode?)>
```

9.2.47 ImpositionTemplateDefaults

ImpositionTemplateDefaults defines default values for all imposition template characteristics in the containing ImpositionTemplateList. The default is only used if the defaulted element is not explicitly defined in ImpositionTemplate.

- **ImpositionTemplateDefaults** may occur as a sub-element of **ImpositionTemplateList**.
- **ImpositionTemplateDefaults** contains the sub-element **ImpositionTemplate**.

NOTE: **ImpositionTemplateDefaults** does not define a default value for **Name**. Each **ImpositionTemplate** element within **ImpositionTemplateList** must have a unique name.

DTD Representation:

```
<!ELEMENT ImpositionTemplateDefaults (ImpositionTemplate)>
```

9.2.48 ImpositionTemplateList

ImpositionTemplateList defines the imposition template resources available on the printing system. All imposition templates in this list are available to any submitted JDF ticket or Virtual Printer Hot Folder. Imposition templates are used to place multiple images from the document onto one printed substrate. Each listed **ImpositionTemplate** defines only those elements for which the value differs from that in **ImpositionTemplateDefaults**.

- **ImpositionTemplateList** may occur as a sub-element of **ManufacturingCapabilities**.
- **ImpositionTemplateList** may contain sub-elements **ImpositionTemplateDefaults** and **ImpositionTemplate**.

DTD Representation:

```
<!ELEMENT ImpositionTemplateList (ImpositionTemplateDefaults?,
    ImpositionTemplate*)>
```

9.2.49 InsertCapable

InsertCapable signifies whether or not an Inserter device is supported by the press.

DTD Representation:

```
<!ELEMENT InsertCapable EMPTY>
    <!ATTLIST InsertCapable Enum (True|False) #REQUIRED>
```

9.2.50 InsertList

InsertList defines a list of insert options that are available to a print job.

- **InsertList** may occur as a sub-element of **FinishingOptions**.
- **InsertList** may contain the sub-element **Inserts**.

DTD Representation:

```
<!ELEMENT InsertList (Inserts*)>
```

9.2.51 Inserts

Inserts defines the supported insert. Available insert options and their values are defined as attributes.

- **Inserts** may occur as a sub-element of **InsertList**.

Attribute Name or value	Writer	Description
Present = True	***w?***	Type: *Boolean* Defines the supported insert options.
True	***w?***	Printing system supports enabling inserts. Default Value.
False	***w?***	Printing system supports disabling inserts.

DTD Representation:

```
<!ELEMENT Inserts EMPTY>
    <ATTLIST Inserts
        Present (True | False) #IMPLIED>
```

9.2.52 InstalledFifthColorList

InstalledFifthColorList defines the list of supported colors for the fifth dry-ink printing station. This list contains those colors that have been set-up for use on the printer. A color must be installed before it can be loaded and used.

- **InstalledFifthColorList** may occur as a sub-element of **FifthColor**.

DTD Representation:

```
<!ELEMENT InstalledFifthColorList (FifthColorName*)>
```

9.2.53 JobTicketTemplate

JobTicketTemplate defines the name of the default job ticket. On the NexPress front end, job ticket templates are stored as PJTF files in \\\CDFE_DATA\Resources\jobticket-templates\<value>. <value> is a string.

- **JobTicketTemplate** occurs as a sub-element of **SystemDefaults**.

DTD Representation:

```
<!ELEMENT JobTicketTemplate (#PCDATA)>
```

9.2.54 Jog

Jog identifies the supported jog-offset options. Available jog options and their values are defined as attributes.

- **Jog** may occur as a sub-element of **Jogging**.

Attribute Name or value	Writer	Description
Enable = True	***w?***	Type: *Boolean* Defines the supported jog offset options.
True	***w?***	Printing system supports enabling jogging. Default Value
False	***w?***	Printing system supports disabling jogging.

DTD Representation:

```
<!ELEMENT Jog EMPTY>
    <!ATTLIST Jog
        Enable (True | False) #IMPLIED>
```

9.2.55 Jogging

Jogging defines a list of jog options available to a print job. When jogging is enabled and the job is sent to an output that supports offset stacking, each set of print job output will be jog-offset from the preceding set. Refer to **Collation** for details about set collation.

- **Jogging** may occur as a sub-element of **FinishingOptions**.
- **Jogging** may contain the sub-element **Jog**.

DTD Representation:

```
<!ELEMENT Jogging (Jog*)>
```

9.2.56 LayoutMode

LayoutMode defines whether page content is to be printed on one or both sides of a sheet for the containing **ImpositionTemplate**. Its value is supplied as an attribute.

- **LayoutMode** may occur as a sub-element of **ImpositionTemplate**.

Attribute Name or value	Writer	Description
Enum = Simplex	***w?***	Type: *Enumeration* Defines whether page content is to be printed on one or both sides of a sheet for the containing **ImpositionTemplate**.

Attribute Name or value	Writer	Description
Simplex	***w?***	Single sided printing. Default Value.
Duplex	***w?***	Two sided printing.
Present	***w!***	Type: *Boolean* Not supported.

DTD Representation:

```
<!ELEMENT LayoutMode EMPTY>
    <!ATTLIST LayoutMode
        Enum (Simplex | Duplex) #IMPLIED
        Present (True | False) #IMPLIED>
```

9.2.57 Length

Length defines the length/height component of a rectangular dimension. **Length** can be in units of millimeters or inches; units are provided as an attribute.

- **Length** may occur as a sub-element of **BoundedMediumData**.

Attribute Name or value	Writer	Description
Unit = MM	***w?***	Type: *Enumeration* Unit of measure for **Length**.
MM	***w?***	Millimeters. Default Value
IN	***w?***	Inches.

DTD Representation:

```
<!ELEMENT Length (#PCDATA)>
    <!ATTLIST Length
        Unit (MM|IN) #IMPLIED>
```

9.2.58 ManufacturingCapabilities

ManufacturingCapabilities is the outermost element of the XML hierarchy in the Device Capabilities File. It contains all other elements. There is only one occurrence of **ManufacturingCapabilities**.

- **ManufacturingCapabilities** may contain sub-elements **DeviceID**, **PrinterList**, **MediumList**, **SubstrateSupplyList**, **FontList**, **ImpositionTemplateList**, **ICCProfileList**, **ResourceCheckProfileList**, **ScreeningSystems**, **SpotColorTableList**, **WorkFlow**, **SystemDefaults**, and **FinishingOptions**.

DTD Representation:

```
<!ELEMENT ManufacturingCapabilities (DeviceID?, PrinterList?, MediumList?, SubstrateSupplyList?, FontList?,
    ImpositionTemplateList?, ICCProfileList?, ResourceCheckProfileList?, ScreeningSystems? SpotColorTableList?,
    Workflow?, SystemDefaults?, FinishingOptions?)*>
```

9.2.59 MaxLicensedEngineSpeed

MaxLicensedEngineSpeed defines the maximum speed for which the press is licensed. If this speed is lower than the maximum value found in **EngineSpeeds** the higher speeds will not be used.

- **MaxLicensedEngineSpeed** may occur as a sub-element of **Printer**.

DTD Representation:

```
<!ELEMENT MaxLicensedEngineSpeed (EngineSpeed)>
```

9.2.60 MaxMediumData

MaxMediumData defines the maximum range values for a medium supported by the printer.

- **MaxMediumData** may occur as a sub-element of **MediumConstraints**.
- **MaxMediumData** contains sub-element **BoundedMediumData**.

DTD Representation:

```
<!ELEMENT MaxMediumData (BoundedMediumData)>
```

9.2.61 MaxSize

MaxSize defines the maximum supported rectangular dimensions (width and height) for a sheet of substrate. **MaxSize** is used to characterize both supply and output locations.

- **MaxSize** occurs as a sub-element of **SubstrateSupply** and **Delivery**.
- **MaxSize** contains the sub-elements **Width** and **Height**.

DTD Representation:

```
<!ELEMENT MaxSize (Width, Height)>
<!ATTLIST MaxSize Unit (MM) #IMPLIED>
```

9.2.62 Medium

Medium defines the characteristics of a substrate known to the printing system.

- **Medium** may occur as a sub-element of **MediumList**.
- **Medium** contains sub-element **Name**. It may also contain sub-elements **SystemMedium**, **PrinterMediumDefaults**, and **PrinterMedium**.

DTD Representation:

```
<!ELEMENT Medium (Name, SystemMedium?, PrinterMediumDefaults?, PrinterMedium*)>
```

9.2.63 MediumConstraints

MediumConstraints defines the range of medium sizes, substrate types and surface types supported by the printer.

- **MediumConstraints** may occur as a sub-element of Printer.
- **MediumConstraints** may contain sub-elements **MinMediumData**, **MaxMediumData**, **SubstrateList**, **SurfaceList**, **ColorList, SupportedSizeList**.

DTD Representation:

```
<!ELEMENT MediumConstraints (MinMediumData?, MaxMediumData?, SubstrateList?, SurfaceList?, ColorList?,
    SupportedSizeList?)>
```

9.2.64 MediumList

MediumList defines all printable substrates, or mediums, known to the printing system. Each listed **Medium** defines only those elements for which the value differs from that in **SystemMediumDefaults**.

- **MediumList** may occur as a sub-element of **ManufacturingCapabilities**.
- **MediumList** may contain sub-elements **SystemMediumDefaults** and **Medium**.

NOTE: The contents of **MediumList** are available to all Virtual Printer Hot Folders defined for the printing system.

DTD Representation:

```
<!ELEMENT MediumList (SystemMediumDefaults?, Medium*)>
```

9.2.65 MediumWeight

MediumWeight defines the basis weight of a print substrate. Weight can be in units of grams per square meter, or pounds; units are provided as an attribute.

- **MediumWeight** may occur as a sub-element of **BoundedMediumData**.

Attribute Name or value	Writer	Description
Unit = GSM	***w?***	Type: *Enumeration* Unit of measure for **MediumWeight**.
GSM	***w?***	Grams per square meter. Default Value
LB	***w?***	Pound.

DTD Representation:

```
<!ELEMENT MediumWeight (#PCDATA)>
    <!ATTLIST MediumWeight
        Unit (GSM|LB) #IMPLIED>
```

9.2.66 MinMediumData

MinMediumData defines the minimum range values for a medium supported by the printer.

- **MinMediumData** may occur as a sub-element of **MediumConstraints**.
- **MinMediumData** contains sub-element **BoundedMediumData**.

DTD Representation:

```
<!ELEMENT MinMediumData (BoundedMediumData)>
```

9.2.67 MinSize

MinSize defines the minimum supported rectangular dimensions (width and height) for a sheet of substrate. **MinSize** is used to characterize both supply and output locations.

- **MinSize** occurs as a sub-element of **SubstrateSupply** and **Delivery**.
- **MinSize** contains the sub-elements **Width** and **Height**.

DTD Representation:

```
<!ELEMENT MinSize (Width, Height)>
<!ATTLIST MinSize Unit (MM) #IMPLIED>
```

9.2.68 Name

Name identifies a resource known to the printing system. Its value is an ASCII string.

- **Name** occurs as a sub-element of **Medium**, **Font**, **ImpositionTemplate**, **ICCProfile**, **ResourceCheckProfile**, and **SpotColorTable**.

DTD Representation:

```
<!ELEMENT Name (#PCDATA)>
```

9.2.69 OffsetCapable

OffsetCapable defines whether the output location of the containing **Delivery** element supports offset stacking. Its value is provided as an attribute.

- **OffsetCapable** may occur as a sub-element of **Delivery**.

Attribute Name or value	Writer	Description
Enum	***w***	Type: *Boolean* Indicates support for stacker offset capability.
True	***w?***	Has offset stacking capability.
False	***w?***	Has no offset stacking capability.

DTD Representation:

```
<!ELEMENT OffsetCapable EMPTY>
    <!ATTLIST OffsetCapable
        Enum (True | False) #REQUIRED>
```

9.2.70 OperatorSelectable

OperatorSelectable specifies whether the output location of the containing **Delivery** element is operator-selectable at the NexPress Client. Its value is provided as an attribute.

- **OperatorSelectable** may occur as a sub-element of **Delivery**.

Attribute Name or value	Writer	Description
Enum	***w***	Type: *Boolean* Indicates support for operator selection of an output delivery capability from the NexPress Client.
True	***w?***	Has support for operator selection of an output delivery capability.
False	***w?***	Has no support for operator selection of an output delivery capability.

DTD Representation:

```
<!ELEMENT OperatorSelectable EMPTY>
    <!ATTLIST OperatorSelectable
        Enum (True | False) #REQUIRED>
```

9.2.71 OutputDelivery

OutputDelivery specifies the sheet orientation characteristics of the containing **Delivery** location. **OutputDelivery** is a characteristic of device construction and paper path design.

- **OutputDelivery** occurs as a sub-element of **Delivery**.

Attribute Name or value	Writer	Description
Enum	***w***	Type: *Enumeration* Indicates orientation of printed surface in the output tray.
SheetFlip	***w?***	Printed sheets are flipped as they are placed into the output location.
NoSheetFlip	***w?***	Printed sheets are not flipped as they are placed into the output location.

DTD Representation:

```
<!ELEMENT OutputDelivery EMPTY>
    <!ATTLIST OutputDelivery
        Enum (SheetFlip | NoSheetFlip) #REQUIRED>
```

9.2.72 PageOrder

PageOrder identifies the supported page order and orientation options. Available page ordering options and values are defined as attributes.

- **PageOrder** may occur as a sub-element of **PageOrderList**.

Attribute Name or value	Writer	Description
Enum	***w***	Type: *Enumeration* Defines the supported **PageOrder** options.
FaceUp	***w?***	Sheets can be output face-up and in forward order.
FaceDown	***w?***	Sheets can be output facedown and in forward order.
FaceUpReverseOrder	***w?***	Sheets can be output face up and in reverse order.
FaceDownReverseOrder	***w?***	Sheets can be output face down and in reverse order.

DTD Representation:

```
<!ELEMENT PageOrder EMPTY>
    <!ATTLIST PageOrder
        Enum (FaceUp | FaceDown | FaceUpReverseOrder | FaceDownReverseOrder) #REQUIRED>
```

9.2.73 PageOrderList

PageOrderList defines the combinations of face-up or facedown, and forward or reverse sheet ordering that are available to a print job.

- **PageOrderList** may occur as a sub-element of **FinishingOptions**.
- **PageOrderList** may contain the sub-element **PageOrder**.

DTD Representation:

```
<!ELEMENT PageOrderList (PageOrder*)>
```

9.2.74 PagesPerSheet

PagesPerSheet defines the number of PDL pages consumed by a single application of the containing **ImpositionTemplate**. Its value is an integer.

- **PagesPerSheet** may occur as a sub-element of **ImpositionTemplate**.

NOTE: As an example, if the imposition template is defined as a two-sided, two-up imposition; **PagesPerSheet** would be 4 because it requires two placed objects, or pages, per side.

DTD Representation:

```
<!ELEMENT PagesPerSheet (#PCDATA)>
```

9.2.75 PixelRectangle

PixelRectangle contains the dimensions of a rectangular imaged area and its offset from the substrate edges. It is used when defining a printing medium or substrate tray. Sub-elements of **PixelRectangle** are specified in units of pixels.

- **PixelRectangle** occurs as a sub-element of **PrinterMedium**.
- **PixelRectangle** may contain sub-elements **PixelsPerUnit**, **RectLengthOffset**, **RectLengthCount**, **RectWidthOffset**, and **RectWidthCount**.

DTD Representation:

```
<!ELEMENT PixelRectangle (PixelsPerUnit?, RectLengthOffset?, RectLengthCount?, RectWidthOffset?,
    RectWidthCount?)>
```

9.2.76 PixelsPerUnit

PixelsPerUnit defines the vertical and horizontal resolution of the printing system as it pertains to the other elements of this **PixelRectangle**. **PixelsPerUnit** can be in units of pixels per inch, or pixels per centimeter; units are provided as an attribute. To convert other elements of **PixelRectangle** to inches or centimeters as appropriate, divide the corresponding element by the value of **PixelsPerUnit**.

- **PixelsPerUnit** may occur as a sub-element of **PixelRectangle**.

Attribute Name or value	Writer	Description
Unit = CM	***w?***	Type: *Enumeration* Unit of measure for **PixelPerUnit**.
CM	***w?***	Pixels per centimeter. Default Value
IN	***w?***	Pixels per inch.

DTD Representation:

```
<!ELEMENT PixelsPerUnit (#PCDATA)>
    <!ATTLIST PixelsPerUnit
        Unit(CM | IN) #IMPLIED>
```

9.2.77 PrecollatedSetSize

PrecollatedSetSize defines the set size when a **PrinterMedium** is collated

- **PrecollatedSetSize** may occur as a sub-element of **PrinterMedium**.

9.2.78 Printer

Printer defines a Virtual Printer and Virtual Printer Hot Folder configured on the NexPress front end.

- **Printer** may occur as a sub-element of **PrinterList** and **PrinterDefaults**.
- **Printer** contains sub-element **PrinterName**. It may also contain sub-elements **Comment**, **PrinterType**, **PrinterPlatform, PrinterSerialNumber**, **I2PPlatform, PrinterVersion**, **PrinterSpeed**, **ExternalControllerType**, **ColorSpaceList**, **FifthColor**, **MediumConstraints, SupportedEngineSpeedList** and **MaxLicensedEngineSpeed.**

DTD Representation:

```
<!ELEMENT Printer (PrinterName, Comment?, PrinterType?, PrinterPlatform?, PrinterSerialNumber? I2PPlatform?,
    PrinterVersion?, PrinterSpeed?, ExternalControllerType?, ColorSpaceList?, FifthColor?, MediumConstraints?,
    SupportedEngineSpeedList?, MaxLicensedEngineSpeed?)>
```

9.2.79 PrinterDefaults

PrinterDefaults defines capabilities common to all Virtual Printers and their Virtual Printer Hot Folders in the containing **PrinterList**.

- **PrinterDefaults** may occur as a sub-element of **PrinterList**.
- **PrinterDefaults** contains sub-element **Printer**.

NOTE: **PrinterDefaults** is different from other patterns of **ItemList** (**ItemDefaults**, **Item***) found in the Device Capabilities File. Each Virtual Printer shares all common elements of **Printer**, as defined in **PrinterDefaults**,

except **PrinterName** and **Comment**. None of the remaining elements are redefined; each **Printer** shares this same configuration.

NOTE: **PrinterName** will be empty within **PrinterDefaults**. Each Virtual Printer must have a unique name; a default value for **PrinterName** is not applicable.

DTD Representation:

```
<!ELEMENT PrinterDefaults(Printer)>
```

9.2.80 PrinterList

PrinterList defines the printer configuration and a list of available Virtual Printers and their Virtual Printer Hot Folders.

- **PrinterList** may occur as a sub-element of **ManufacturingCapabilities**.
- **PrinterList** may contain sub-elements **PrinterDefaults** and **Printer**.

NOTE: All Virtual Printers identified in **PrinterList** have identical capabilities in terms of the resources available to them.

DTD Representation:

```
<!ELEMENT PrinterList (PrinterDefaults?, Printer*)>
```

9.2.81 PrinterMedium

PrinterMedium defines substrate-specific imaging characteristics. For historical reasons it is distinct from **PrinterMediumDefaults**, but in practice **PrinterMedium** exists only as a sub-element of **PrinterMediumDefaults**.

- **PrinterMedium** may occur as a sub-element of **PrinterMediumDefaults** and **Medium**.
- **PrinterMedium** may contain sub-elements **CertifiedMedium, GlossUnitCompatible**, **PixelRectangle, HasGlosserSettings, PrecollatedSetSize, FeedEdge** and **SupportedEngineSpeedList.**

DTD Representation:

```
<!ELEMENT PrinterMedium (CertifiedMedium?, GlossUnitCompatible?, PixelRectangle?, HasGlosserSettings?,
                         PrecollatedSetSize?, FeedEdge?, SupportedEngineSpeedList?)>
```

9.2.82 PrinterMediumDefaults

PrinterMediumDefaults defines substrate-specific imaging characteristics. For historical reasons it is distinct from **PrinterMedium**, but in practice **PrinterMedium** exists only as a sub-element of **PrinterMediumDefaults**.

- **PrinterMediumDefaults** may occur as a sub-element of **Medium**.
- **PrinterMediumDefaults** contains sub-element **PrinterMedium**.

DTD Representation:

```
<!ELEMENT PrinterMediumDefaults (PrinterMedium)>
```

9.2.83 PrinterName

PrinterName provides the name of a Virtual Printer and its Virtual Printer Hot Folder. Its value is an ASCII string.

- **PrinterName** occurs as a sub-element of **Printer** and **PrinterMedium**.

DTD Representation:

```
<!ELEMENT PrinterName (#PCDATA)>
```

9.2.84 PrinterPlatform

PrinterPlatform identifies the NexPress product model or model series.

- **PrinterPlatform** may occur as a sub-element of **Printer**.

Attribute Name or value	Writer	Description
Enum	***w***	Type: *Enumeration* Printing system platform type.
NexPress_Classic	***w?***	NexPress Classic 2100
NexPress_NPP	***w?***	NexPress NPP Series with New Paper Platform
NexPress_M700	***w?***	NexPress M700
Unknown	***w?***	

DTD Representation:

```
<!ELEMENT PrinterPlatform EMPTY>
    <!ATTLIST PrinterPlatform
        Enum (NexPress_Classic | NexPress_NPP | NexPress_M700 | Unknown) #REQUIRED>
```

9.2.85 PrinterSerialNumber

PrinterSerialNumber reports the serial number from the engine to which the DFE is connected.

- **PrinterSerialNumber** may occur as a sub-element of **Printer**.

DTD Representation:

```
<!ELEMENT PrinterType (#PCDATA)>
```

9.2.86 PrinterSpeed

PrinterSpeed reports the printing system productivity. Productivity is reported as the number of A4-simplex pages printed per minute; units are provided as an attribute.

- **PrinterSpeed** may occur as a sub-element of **Printer**.

Attribute Name or value	Writer	Description
Unit = PPM	***w?***	Type: *Enumeration* Unit of measure for **PrinterSpeed**.
PPM	***w***	A4 Equivalent Pages Per Minute. Default Value.

DTD Representation:

```
<!ELEMENT PrinterSpeed (#PCDATA)>
    <!ATTLIST PrinterSpeed
        Unit (PPM) #IMPLIED>
```

9.2.87 PrinterType

PrinterType contains an ASCII string supplied by the print engine to identify model or platform. The format and content of this string are not specified by this interface.

- **PrinterType** may occur as a sub-element of **Printer**.

DTD Representation:

```
<!ELEMENT PrinterType (#PCDATA)>
```

9.2.88 PrinterVersion

PrinterVersion contains firmware version and machine configuration information. Firmware version is supplied as an ASCII string. The format and content of this string are not specified by this interface. Machine configuration information is provided as an attribute.

- **PrinterVersion** may occur as a sub-element of **Printer**.

NOTE: The format and content of **PrinterVersion** is controlled by the print engine firmware. The content reported in a simulation environment differs from that reported by an actual print engine.

Attribute Name or value	Writer	Description
MachineConfiguration	***w***	Type: *CDATA* A bit-encoded description of selected machine configuration information from the printer. The format of this description is not defined by this interface.

DTD Representation:

```
<!ELEMENT PrinterVersion (#PCDATA)>
    <!ATTLIST PrinterVersion
        MachineConfiguration CDATA #REQUIRED>
```

9.2.89 PunchPattern

PunchPattern describes the punching pattern of an attached finisher if punching is supported by the finishing device.

DTD Representation:

```
<!ELEMENT PunchPattern EMPTY>
    <!ATTLIST PunchPattern Enum (None | 2_Hole | 2_AND_3_Hole | 4_Hole_France | 4_Hole_Sweden |
Unknown) #REQUIRED>
```

9.2.90 RectLengthCount

RectLengthCount specifies the length for a rectangular imageable area defining the containing **PixelRectangle**. The edge of the cut sheet media corresponding to **RectLengthCount** depends upon the size of the media. As a rule, length corresponds to the dimension of the media that is along the in-track direction (not the lead edge) as it is fed through the printer. Since the smaller media sizes of A4 and Letter are fed through the NexPress digital production color press with long edge as the lead edge, the short edge corresponds to the value of **RectLengthCount**. For larger size cut sheet media, including A3, Tabloid, and A3+, the short edge is the lead edge as it is fed through the print engine, so the long edge of these larger media corresponds to the value of **RectLengthCount**. **RectLengthCount** is specified in units of pixels as defined by **PixelsPerUnit**.

- **RectLengthCount** may occur as a sub-element of **PixelRectangle**.

DTD Representation:

```
<!ELEMENT RectLengthCount (#PCDATA)>
```

9.2.91 RectLengthOffset

RectLengthOffset specifies the length-dimension offset for the rectangular imageable area of the enclosing **PixelRectangle**. This offset is in the same dimension as **RectLengthCount**; refer to the description of **RectLengthCount** for a description of which edge of the physical media corresponds to the length dimension. **RectLengthOffset** is specified in units of pixels as defined by **PixelsPerUnit.**

- **RecLengthOffset** may occur as a sub-element of **PixelRectangle**.

DTD Representation:

```
<!ELEMENT RectLengthOffset (#PCDATA)>
```

9.2.92 RectWidthCount

RectWidthCount specifies the width for a rectangular imageable area defining the containing **PixelRectangle**. The edge of the cut sheet media corresponding to **RectWidthCount** depends upon the size of the media. As a rule, width corresponds to the dimension of the media that is the lead edge as it is fed through the printer. Since the smaller media sizes of A4 and Letter are fed through the NexPress digital production color press with long edge as the lead edge, the long edge corresponds to the value of **RectWidthCount**. For larger size cut sheet media, including A3, Tabloid, and A3+, the short edge is the lead edge as it is fed through the print engine, so the short edge of these larger media corresponds to the value of **RectWidthCount**. **RectWidthCount** is specified in units of pixels as defined by **PixelsPerUnit**.

- **RectWidthCount** may occur as a sub-element of **PixelRectangle**.

DTD Representation:

```
<!ELEMENT RectWidthCount (#PCDATA)>
```

9.2.93 RectWidthOffset

RectWidthOffset specifies the width-dimension offset for the rectangular imageable area of the enclosing **PixelRectangle**. This offset is in the same dimension as **RectWidthCount**; refer to the description of **RectWidthCount** for a description of which edge of the physical media corresponds to the width dimension. **PixelsPerUnit** defines the resolution of **RectWidthOffset** in units of pixels.

- **RectWidthOffset** may occur as a sub-element of **PixelRectangle**.

DTD Representation:

```
<!ELEMENT RectWidthOffset (#PCDATA)>
```

9.2.94 ResourceCheckProfile

ResourceCheckProfile identifies the name of an available resource check profile.

- **ResourceCheckProfile** may occur as a sub-element of **ResourceCheckProfileList**.
- **ResourceCheckProfile** contains the sub-element **Name**.

NOTE: Names of resource check profiles are not specified by this interface. Typical names include "*Medium*", "*Severe*", and "*Tolerant*".

DTD Representation:

```
<!ELEMENT ResourceCheckProfile (Name)>
```

9.2.95 ResourceCheckProfileList

ResourceCheckProfileList defines a list of severity options that control resource check. The job-specified profile determines which resouce checks are performed during job submission. The resource check profile of a Virtual Printer and its Virtual Printer Hot Folder can be set to one of the values lised in **ResourceCheckProfileList**. JDF jobs always use the resource check profile of the default Job Ticket, which is not configurable.

- **ResourceCheckProfileList** occurs as a sub-element of **ManufacturingCapabilities**.
- **ResourceCheckProfileList** may contain sub-element **ResourceCheckProfile**.

DTD Representation:

```
<!ELEMENT ResourceCheckProfileList (ResourceCheckProfile*)>
```

9.2.96 RGBGraphicICCProfile

RGBGraphicICCProfile defines the name of an input ICC profile suitable for use with three component PDF graphics and text data that are specified using the *DeviceRGB* color space. This name should correspond to an entry in **ICCProfileList**. Its value is a character string.

- **RGBGraphicICCProfile** occurs as a sub-element of **SystemDefaults**.

DTD Representation:

```
<!ELEMENT RGBGraphicICCProfile (#PCDATA)>
```

9.2.97 RGBImageICCProfile

RGBImageICCProfile defines the name of an input ICC profile suitable for use with three component PDF image data that are specified using the *DeviceRGB* color space. This name should correspond to an entry in **ICCProfileList**. Its value is a character string.

- **RGBImageICCProfile** occurs as a sub-element of **SystemDefaults**.

DTD Representation:

```
<!ELEMENT RGBImageICCProfile (#PCDATA)>
```

9.2.98 ScreeningSystems

ScreeningSystems defines half-tone screen types supported by the printing system. Each screen type is selectable from a job ticket.

- **ScreeningSystems** may occur as a sub-type of **ManufacturingCapabilities**.
- **ScreeningSystems** may contain the sub-type **ScreenType**.

DTD Representation:

```
<!ELEMENT ScreeningSystems (ScreenType*)>
```

9.2.99 ScreenType

ScreenType defines a half-tone screen type. Its identity and type are supplied as attributes.

- **ScreenType** may occur as a sub-element of **ScreeningSystems**.

Attribute Name or value	Writer	Description
ScreenType = Classic	***w?***	Type: *Enumeration* Defines a half-tone screen type.
Classic	***w?***	Default Value.
Optimum	***w?***	
Line	***w?***	
Supra	***w?***	
Staccato DX	***w?***	
ClassicHD	***w?***	
None	***w?***	
Other	***w?***	

DTD Representation:

```
<!ELEMENT ScreenType EMPTY>
    <!ATTLIST ScreenType
```

Enum (Classic | Optimum | Line | Supra | Stochastic | None | Other) #IMPLIED>

9.2.100 SecuredColorant

SecuredColorant identifies colorants that support secure printing. When **SecuredColorant** is defined for a **FifthColorName** in the **InstalledFifthColorList**, the colorant is available for use with secured printing modes. **SecuredColorant** is not repeated for **CurrentLoadedFifthColor** even if the loaded colorant is a secured colorant.

- **SecuredColorant** may occur as a sub-element of **FifthColorName**.

DTD Representation:

<!ELEMENT SecuredColorant EMPTY>

<!ELEMENT FifthColorName (SecuredColorant?)>

<!ATTLIST FifthColorName Enum (#PCDATA) #IMPLIED>

<!ELEMENT InstalledFifthColorList (FifthColorName*)>

9.2.101 SheetsPerSignature

SheetsPerSignature defines the number of sheets generated by a single application of the containing **ImpositionTemplate**. Its value is an integer.

- **SheetsPerSignature** may occur as a sub-element of **ImpositionTemplate**.

DTD Representation:

<!ELEMENT SheetsPerSignature (#PCDATA)>

9.2.102 SpotColorTable

SpotColorTable defines a spot color to CMYK process color translation table.

- **SpotColorTable** may occur as a sub-element of **SpotColorTableList**.
- **SpotColorTable** contains the sub-element **Name**.

DTD Representation:

<!ELEMENT SpotColorTable (Name)>

9.2.103 SpotColorTableList

SpotColorTableList defines the list of named spot color to CMYK process color translation tables available on the printing system.

- **SpotColorTableList** may occur as a sub-element of **ManufacturingCapabilities**.
- **SpotColorTableList** may contain the sub-element **SpotColorTable**.

DTD Representation:

<!ELEMENT SpotColorTableList (SpotColorTable*)>

9.2.104 StapleCapable

StapleCapable specifies whether the output location of the containing **Delivery** element supports stapling. Its value is provided as an attribute.

- **StapleCapable** may occur as a sub-element of **Delivery**.

Attribute Name or value	Writer	Description

Attribute Name or value	Writer	Description
Enum	***w***	Type: *Boolean* Indicates support for stapling capability.
True	***w?***	Has stapling capability.
False	***w?***	Has no stapling capability.

DTD Representation:

```
<!ELEMENT StapleCapable EMPTY>
    <!ATTLIST StapleCapable
        Enum (True | False) #REQUIRED>
```

9.2.105 StopPoint

StopPoint defines the conditions under which the printing system stops execution of a job and requires operator intervention. Its value is supplied as an attribute.

- **StopPoint** occurs as a sub-element of **StopPointList**.

Attribute Name or value	Writer	Description
Enum	***w***	Type: *Enumeration* Job execution policy option. Indicates condition for which execution of a job is stopped and requires operator intervention. **StopPoint** may be specified in the job ticket.
StopAfterSubmission	***w?***	Stop execution of the job immediately after job ticket submission.
StopWhenPDFAvailable	***w?***	Stop execution of the job immediately after all PDF files referenced by the submitted job ticket have been made available.
StopAfterResourceCheck	***w?***	Stop execution of the job immediately after the resource checker has verified resource files required by the job ticket.
StopAfterRIP	***w?***	Stop execution of the job immediately after the job has been ripped.

DTD Representation:

```
<!ELEMENT StopPoint EMPTY>
    <!ATTLIST StopPoint
        Enum (StopAfterSubmission | StopWhenPDFAvailable | StopAfterResourceCheck | StopAfterRIP)
    #REQUIRED>
```

9.2.106 StopPointList

StopPointList defines a list of job execution policy options supported by the printing system. These policies control the conditions under which execution of a job is stopped.

- **StopPointList** occurs as a sub-element of **Workflow**.
- **StopPointList** may contain the sub-element **StopPoint**.

DTD Representation:

```
<!ELEMENT StopPointList (StopPoint*)>
```

9.2.107 SubstrateDeliveryList

SubstrateDeliveryList defines a list of delivery options for printed output.

- **SubstrateDeliveryList** may occur as a sub-element of **FinishingOptions**.
- **SubstrateDeliveryList** may contain the sub-element **Delivery**.

DTD Representation:

```
<!ELEMENT SubstrateDeliveryList (Delivery*)>
```

9.2.108 SubstrateList

SubstrateList defines the set of **SubstrateType** used by the printer when characterizing a medium. This is different from **MediumList** that identifies the complete list of media available and qualified for use in the printer.

- **SubstrateList** may occur as a sub-element of a **MediumConstraints**.
- **SubstrateList** may contain sub-element **SubstrateType**.

DTD Representation:

```
<!ELEMENT SubstrateList (SubstrateType*)>
```

9.2.109 SubstrateSizeName

SubstrateSizeName identifies a name associated with a substrate's size. Values of **SubstrateSizeName** are not specified by this interface.

- **SubstrateSizeName** may occur as a sub-element of **BoundedMediumData**.

DTD Representation:

```
<!ELEMENT SubstrateSizeName (#PCDATA)>
```

9.2.110 SubstrateSupply

SubstrateSupply defines the characteristics of an input substrate supply in the printing system. The ID identifying the **SubstrateSupply** is provided as an attribute.

- **SubstrateSupply** may occur as a sub-element of **SubstrateSupplyList**.
- **SubstrateSupply** contains sub-elements **MinSize**, **MaxSize**, **Capacity**, **BayID**, **DoorNumber**, **FeederType, SupportedEngineSpeedList**.

Attribute Name or value	Writer	Description
ID	***W***	Type: *CDATA* The unique identifier of the supply tray as known by the NexPress digital production color press.

DTD Representation:

```
<!ELEMENT SubstrateSupply (MinSize, MaxSize, Capacity, BayID, DoorNumber, FeederType?,
    SupportedEngineSpeedList?)>
    <!ATTLIST SubstrateSupply
        ID CDATA #REQUIRED>
```

9.2.111 SubstrateSupplyList

SubstrateSupplyList defines the substrate supply trays installed on the printing system. The identity and capabilities of each supply are provided.

- **SubstrateSupplyList** may occur as a sub-element of **ManufacturingCapabilities**.
- **SubstrateSupplyList** may contain the sub-element **SubstrateSupply**.

DTD Representation:

<!ELEMENT SubstrateSupplyList (SubstrateSupply*)>

9.2.112 SubstrateType

SubstrateType defines basic characteristics of a substrate. The value of **SubstrateType** is supplied as an attribute.

- **SubstrateType** may occur as a sub-element of **SubstrateList** and **SystemMedium**.

Attribute Name or value	Writer	Description
Enum = Paper	***w?***	Type: *Enumeration* Defines the type of a substrate.
Paper	***w?***	Default Value
Foil	***w?***	
Transparency	***w?***	
OpaqueFoil	***w?***	
Self-AdhesivePaper	***w?***	
TranslucentFoil	***w?***	
SyntheticPaper	***w?***	
TranslucentPaper	***w?***	
Plain	***w?***	
SingleCoated	***w?***	
DoubleCoated	***w?***	
Recycled	***w?***	
Texture	***w?***	
Film	***w?***	
Label	***w?***	
Vellium	***w?***	
Bond	***w?***	
Other	***w?***	
Present	***!w***	Type: *Boolean* Not supported.

DTD Representation:

```
<!ELEMENT SubstrateType EMPTY>
    <!ATTLIST SubstrateType
        Enum (Paper | Foil | Transparency | OpaqueFoil | Self-AdhesivePaper | TranslucentFoil | SyntheticPaper |
            TranslucentPaper | Plain | SingleCoated | DoubleCoated | Recycled | Texture | Film | Label |
            Vellium | Bond | Other) #IMPLIED
        Present (True | False) #IMPLIED>
```

9.2.113 SupportedEngineSpeedList

SupportedEngineSpeedList defines the set of **EngineSpeed** for which the **PrinterMedium** is supported. If a job selects a substrate not supported at the configured engine speed the job will not be permitted to print.

- **SupportedEngineSpeedList** may occur as a sub-element of **Printer, PrinterMedium, CurrentlyLoadedFifthColor, SubstrateSupply, DeliveryTypeDetails, Delivery**.
- **SupportedEngineSpeedList** may contain zero or more **EngineSpeed** enumerations.

DTD Representation:

```
<!ELEMENT SubstrateEngineSpeedList (EngineSpeed*)>
```

9.2.114 SurfaceList

SurfaceList defines the set of **SurfaceType** used by the printer when characterizing a medium. This is different from **MediumList** that identifies the complete list of media available and qualified for use in the printer.

- **SurfaceList** may occur as a sub-element of a **MediumConstraints**.
- **SurfaceList** may contain sub-element **SurfaceType**.

DTD Representation:

```
<!ELEMENT SurfaceList (SurfaceType*)>
```

9.2.115 SurfaceType

SurfaceType defines the surface characteristics of a print medium. The value of **SurfaceType** is supplied as an attribute.

- **SurfaceType** may occur as a sub-element of **SurfaceList** and **SystemMedium**.

Attribute Name or value	Writer	Description
Enum = Uncoated	***w?***	Type: *Enumeration* Defines the type of a substrate.
Uncoated	***w?***	Default Value
Parchment	***w?***	
CastCoated	***w?***	
SingleSideCastCoated	***w?***	
TransparentWatermarked	***w?***	
LatexImpregnated	***w?***	
Carbonless	***w?***	
LabelStock	***w?***	
Transfer	***w?***	
Tinted	***w?***	
Embossed	***w?***	
SatinCoated	***w?***	
MatteCoated	***w?***	
GlossyCoated	***w?***	
Textured	***w?***	

Attribute Name or value	Writer	Description
FeltMarked	***w?***	
Colored	***w?***	
Normal	***w?***	
Tabbed	***w?***	
Punched	***w?***	
Other	***w?***	
Present	***!w***	Type: *Boolean* Not supported.

DTD Representation:

```
<!ELEMENT SurfaceType EMPTY>
    <!ATTLIST SurfaceType
        Enum (Uncoated | Parchment | CastCoated | SingleSideCastCoated | TransparentWatermarked |
            LatexImpregnated | Carbonless | LabelStock | Transfer | Tinted | Embossed | SatinCoated |
            MatteCoated | GlossyCoated | Textured | FeltMarked | Colored | Normal | Tabbed |
            Punched | Other) #IMPLIED
        Present (True | False) #IMPLIED>
```

9.2.116 SystemDefaults

SystemDefaults defines default settings for options not explicitly controlled or specified in a job ticket. When a job-specific value is not supplied, default settings determine system behavior.

- **SystemDefaults** may occur as a sub-element of ManufacturingCapabilities.
- **SystemDefaults** contains sub-elements **CMYKGraphicICCProfile**, **CMYKImageICCProfile**, **RGBGraphicICCProfile**, **and RGBImageICCProfile**. It may contain the sub-element **JobTicketTemplate**.

DTD Representation:

```
<!ELEMENT SystemDefaults (CMYKGraphicICCProfile, CMYKImageICCProfile, RGBGraphicICCProfile,
    RGBImageICCProfile, JobTicketTemplate?)>
```

9.2.117 SystemMedium

SystemMedium defines the physical characteristics of a print media.

- **SystemMedium** may occur as a sub-element of **Medium**.
- **SystemMedium** may contain sub-elements **Comment**, **ColorName**, **BoundedMediumData**, **SubstrateType**, **SurfaceType**, and **ColorType**.

DTD Representation:

```
<!ELEMENT SystemMedium (Comment?,ColorName?,BoundedMediumData?,SubstrateType?,SurfaceType?,
    ColorType?)>
```

9.2.118 SystemMediumDefaults

SystemMediumDefaults defines default values for elements within **SystemMedium**..

- **SystemMediumDefaults** occurs as a sub-element of **MediumList**.
- **SystemMediumDefaults** contains sub-element **SystemMedium**.

NOTE: In general the "***Unit***" attribute of elements within **SystemMedium** is the same as the default, and will only be defined within **SystemMediumDefaults**.

DTD Representation:

```
<!ELEMENT SystemMediumDefaults (SystemMedium)>
```

9.2.119 Thickness

Thickness defines the thickness of a print substrate. It is reported in micrometers; units are provided as an attribute.

- **Thickness** may occur as a sub-element of **BoundedMediumData**.

Attribute Name or value	Writer	Description
Unit = UM	***w?***	Type: *Enumeration* Unit of measure for **Thickness**.
UM	***w***	Micrometers. Default Value

DTD Representation:

```
<!ELEMENT Thickness (#PCDATA)>
    <!ATTLIST Thickness
        Unit (UM) #IMPLIED>
```

9.2.120 Transparency

Transparency defines the opacity of a print substrate. It is expressed as a percentage of transmittance; units are provided as an attribute. Value may range from 0 to 100; 0 is opaque and 100 is transparent. Values between 0 and 100 indicate the substrate is semitransparent.

- **Transparency** may occur as a sub-element of **BoundedMediumData**.

Attribute Name or value	Writer	Description
Unit = PCT	***w?***	Type: *Enumeration* Unit of measure for **Transparency**.
PCT	***w***	Percent opacity. Default Value

DTD Representation:

```
<!ELEMENT Transparency (#PCDATA)>
    <!ATTLIST Transparency
        Unit (PCT) #IMPLIED>
```

9.2.121 Trap

Trap defines a control option for the raster trapping engine. Its value is supplied as an attribute.

- **Trap** may occur as a sub-element of **Trapping**.

Attribute Name or value	Writer	Description
Enable	***w***	Type: *Boolean* Defines a control option for the raster trapping engine.
True	***w?***	The job ticket can enable the raster-trapping engine.
False	***w?***	The job ticket can disable the raster-trapping engine.

DTD Representation:

```
<!ELEMENT Trap EMPTY>
    <!ATTLIST Trap
```

Enable (True | False) #REQUIRED>

9.2.122 Trapping

Trapping defines a list of trapping control options supported by the raster-trapping engine of the printing system. Trapping results in an altered image to accommodate registration variability in the print engine.

- **Trapping** occurs as a sub-element of **Workflow**.
- **Trapping** may contain the sub-element **Trap**.

DTD Representation:

<!ELEMENT Trapping (Trap*)>

9.2.123 Width

Width defines the width component of a rectangular dimension. **Width** can be in units of millimeters or inches.

- **Width** may occur as a sub-element of **BoundedMediumData**, **MinSize**, and **MaxSize**.

Attribute Name or value	Writer	Description
Unit = MM	***w?***	Type: *Enumeration* Unit of measure for **Width**.
MM	***w?***	Millimeters. Default Value
IN	***w?***	Inches.

DTD Representation:

<!ELEMENT Width (#PCDATA)>

9.2.124 Workflow

Workflow defines some of the printing system supported workflow options that can be controlled by the job ticket. These job execution policies and processes include control of the conditions under which job execution is stopped, and availability of raster-based trapping control.

Workflow may occur as a sub-element of **ManufacturingCapabilities**.

Workflow contains sub-elements **StopPointList** and **Trapping**.

DTD Representation:

<!ELEMENT Workflow (StopPointList, Trapping)>

9.3 Device Capabilities File DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ********************************************************************************
    * This is the DTD for the Device Manufacturing Capabilities XML data
    * structure.
    *
    * Values used to specify units of measure:
    * UM:   Micrometer
    * MM:   Millimeter
    * CM:   Centimeter
    * IN:   Inch
    * PCT:  Percent
    * GSM:  Grams per square meter
    * LB:   Pound
    * KB:   Kilobyte
```

```
    * PPM:  Pages per minute
    * NUM:  Number
    ********************************************************************** -->

<!-- **********************************************************************
    * Define elements for general use.
    ********************************************************************** -->
<!ELEMENT DeviceID (#PCDATA)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT PrinterName (#PCDATA)>
<!ELEMENT Comment (#PCDATA)>
<!ELEMENT SubstrateSizeName (#PCDATA)>
<!ELEMENT Length (#PCDATA)>
        <!ATTLIST Length Unit (MM|IN) #IMPLIED>
<!ELEMENT Width (#PCDATA)>
        <!ATTLIST Width Unit (MM|IN) #IMPLIED>
<!ELEMENT Transparency (#PCDATA)>
        <!ATTLIST Transparency Unit (PCT) #IMPLIED>
<!ELEMENT Thickness (#PCDATA)>
        <!ATTLIST Thickness Unit (UM) #IMPLIED>
<!ELEMENT ColorSpace EMPTY>
        <!ATTLIST ColorSpace Enum (Gray|CMYK|RGB|LAB|XYZ|
                                   LUV|YXY|3CLR|4CLR|5CLR) #IMPLIED
                             Present (True|False) #IMPLIED>
<!-- **********************************************************************
    * Define the remaining elements to specify a medium's physical
    * characteristics.
    ********************************************************************** -->
<!ELEMENT MediumWeight (#PCDATA)>
        <!ATTLIST MediumWeight Unit (GSM|LB) #IMPLIED>
<!ELEMENT Creep (#PCDATA)>
        <!ATTLIST Creep Unit (UM|IN) #IMPLIED>
<!ELEMENT SubstrateType EMPTY>
        <!ATTLIST SubstrateType Enum (Paper|Foil|Transparency|OpaqueFoil|
                                      Self-AdhesivePaper|TranslucentFoil|
                                      SyntheticPaper|TranslucentPaper|
                                      Plain|SingleCoated|DoubleCoated|
                                      Recycled|Texture|Film|Label|Vellium|
                                      Bond|Other) #IMPLIED
                                Present (True|False) #IMPLIED>
<!ELEMENT SurfaceType EMPTY>
        <!ATTLIST SurfaceType Enum (Uncoated|Parchment|CastCoated|
                                    SingleSideCastCoated|TransparentWatermarked|
                                    LatexImpregnated|Carbonless|LabelStock|
                                    Transfer|Tinted|Embossed|SatinCoated|
                                    MatteCoated|GlossyCoated|Textured|
                                    FeltMarked|Colored|Normal|Tabbed|Punched|
                                    Other) #IMPLIED
                               Present (True|False) #IMPLIED>
<!ELEMENT ColorType EMPTY>
        <!ATTLIST ColorType Enum (Unknown|WhiteNeutral|GrayNeutral|
                                  Yellowish|Blueish|Reddish|Greenish|White|
                                  Blue|Cream|BrightYellow|Gray|Green|
                                  Ivory|Orange|Pink|Red|Yellow|Transparency|
                                  Other) #IMPLIED
                             Present (True|False) #IMPLIED>
<!-- **********************************************************************
    * Define the medium elements that can be constrained by a min/max range.
    ********************************************************************** -->
<!ELEMENT BoundedMediumData
     (Length?,Width?,SubstrateSizeName?,MediumWeight?,Transparency?,Thickness?,
                  Creep?)>
```

```
<!ELEMENT MinMediumData (BoundedMediumData)>
<!ELEMENT MaxMediumData (BoundedMediumData)>

<!-- ********************************************************************************
    * Define the medium elements that can be constrained by a list of valid
    * values.
    ******************************************************************************** -->
<!ELEMENT SubstrateList (SubstrateType*)>
<!ELEMENT SurfaceList (SurfaceType*)>
<!ELEMENT ColorList (ColorType*)>

<!-- ********************************************************************************
    * Define element MediumConstraints as an aggregate of elements that
    * constrain the media that can be used on a particular printer.
    ******************************************************************************** -->
<!ELEMENT MediumConstraints (MinMediumData?,MaxMediumData?,
                    SubstrateList?,SurfaceList?,ColorList?)>

<!-- ********************************************************************************
    * Define the remaining elements that make up the Printer Element.
    ******************************************************************************** -->
<!ELEMENT PrinterType (#PCDATA)>
<!ELEMENT PrinterPlatform EMPTY>
        <!ATTLIST PrinterPlatform Enum (NexPress_Classic|NexPress_NPP|NexPress_M700|Unknown)
      #REQUIRED>
<!ELEMENT I2PPlatform EMPTY>
        <!ATTLIST I2PPlatform Enum (I2P|I2P2|Unknown) #REQUIRED>
<!ELEMENT PrinterVersion (#PCDATA)>
        <!ATTLIST PrinterVersion MachineConfiguration CDATA #REQUIRED>

<!-- ********************************************************************************
    * Legacy support for element PrinterSpeed is supported. Engines capable of
    * multiple speeds are characterized by SupportedEngineSpeedList and
    * MaxLicensedEngineSpeed. PrinterSpeed is derived from MaxLicensedEngineSpeed.
    ******************************************************************************** -->
<!ELEMENT PrinterSpeed (#PCDATA)>
        <!ATTLIST PrinterSpeed Unit (PPM) #IMPLIED>
<!ELEMENT EngineSpeed EMPTY>
<!ATTLIST EngineSpeed Enum (70|83|100|120|150|Auto|Unknown|Other) #REQUIRED>
<!ELEMENT SupportedEngineSpeedList (EngineSpeed*)>
<!ELEMENT MaxLicensedEngineSpeed (EngineSpeed) >
<!ELEMENT ColorSpaceList (ColorSpace*)>

<!-- ********************************************************************************
    * When SecuredColorant is defined for a FifthColorName in the
    * InstalledFifthColorList, the colorant is available for use with Secured
    * printing modes. It is not repeated in CurrentLoadedFifthColor even if
    * the loaded colorant is a Secured colorant.
    ******************************************************************************** -->
<!ELEMENT SecuredColorant EMPTY>

<!ELEMENT FifthColorName (SecuredColorant?)>
        <!ATTLIST FifthColorName Enum (#PCDATA) #IMPLIED>
<!ELEMENT InstalledFifthColorList (FifthColorName*)>
<!ELEMENT CurrentColorMode EMPTY>
        <!ATTLIST CurrentColorMode Enum (Auto|Black|CMYK|5CLR|Unknown) #IMPLIED>
<!ELEMENT CurrentLoadedFifthColor (FifthColorName, SupportedEngineSpeedList?)>
<!ELEMENT FifthColor (InstalledFifthColorList*,CurrentColorMode?,CurrentLoadedFifthColor?)>
        <!ATTLIST FifthColor Supported (True|False) #IMPLIED>
<!ELEMENT ExternalControllerType EMPTY>
        <!ATTLIST ExternalControllerType Enum (NONE|FIERY|BRISQUE|CREO_PODS) #REQUIRED>

<!-- ********************************************************************************
```

```
    * Define the Printer element as an aggregate element, and the PrinterList
    * element as a list of Printer elements.  Allow a printer list to
    * specify default printer settings.
    ********************************************************************** -->
<!ELEMENT Printer (PrinterName,Comment?,PrinterType?, PrinterPlatform?, I2PPlatform?, PrinterVersion?,
                PrinterSpeed?,ExternalControllerType?,ColorSpaceList?,FifthColor?,MediumConstraints?,
                SupportedEngineSpeedList?, MaxLicensedEngineSpeed?)>

<!ELEMENT PrinterDefaults (Printer)>

<!ELEMENT PrinterList (PrinterDefaults?,Printer*)>


<!-- **********************************************************************
    * Define an aggregate element that specifies a medium's physical
    * characteristics.
    ********************************************************************** -->
<!ELEMENT SystemMedium (Comment?,ColorName?,BoundedMediumData?,SubstrateType?,
                    SurfaceType?,ColorType?)>

<!-- **********************************************************************
    * Define the elements that make up a pixel rectangle.
    ********************************************************************** -->
<!ELEMENT PixelsPerUnit (#PCDATA)>
        <!ATTLIST PixelsPerUnit Unit (CM|IN) #IMPLIED>
<!ELEMENT RectLengthOffset (#PCDATA)>
<!ELEMENT RectLengthCount (#PCDATA)>
<!ELEMENT RectWidthOffset (#PCDATA)>
<!ELEMENT RectWidthCount (#PCDATA)>

<!ELEMENT PixelRectangle (PixelsPerUnit?,RectLengthOffset?,RectLengthCount?,
                    RectWidthOffset?,RectWidthCount?)>

<!-- **********************************************************************
    * Define the remaining elements that specify printer-dependent
    * characteristics of a medium for a particular printer.
    ********************************************************************** -->
<!ELEMENT CertifiedMedium EMPTY>
        <!ATTLIST CertifiedMedium Present (True|False) #IMPLIED>

<!-- **********************************************************************
    * When HasGlosserSettings is defined for a PrinterMedium,
    * glosser settings are defined for this substrate/medium.
    ********************************************************************** -->
<!ELEMENT HasGlosserSettings EMPTY>

<!-- **********************************************************************
    * When GlossUnitCompatible is defined for a PrinterMedium,
    * the medium is compatible with the external glosser.
    ********************************************************************** -->
<!ELEMENT GlossUnitCompatible EMPTY>

<!-- **********************************************************************
    * Define an aggregate element that specifies printable characteristics
    * of a medium.
    ********************************************************************** -->
<!ELEMENT PrinterMedium (CertifiedMedium?,GlossUnitCompatible?,PixelRectangle?, HasGlosserSettings?,
                                    PrecollatedSetSize?, FeedEdge?, SupportedEngineSpeedList?)>

<!-- **********************************************************************
    * Define the Medium element as an aggregate element, and the MediumList
    * element as a list of Medium elements.  Allow a Medium element to specify
    * default PrinterMedium elements.  Allow a MediumList element to specify
```

```
    * default SystemMedium elements.
    ********************************************************************** -->
<!ELEMENT PrinterMediumDefaults (PrinterMedium)>
<!ELEMENT Medium (Name,SystemMedium?,PrinterMediumDefaults?,PrinterMedium*)>
<!ELEMENT SystemMediumDefaults (SystemMedium)>
<!ELEMENT MediumList (SystemMediumDefaults?,Medium*)>

<!-- ************************************************************************
    * Define the SubstrateSupply element as an aggregate element, and the
    * SubstrateSupplyList element as a list of SubstrateSupply elements.
    ********************************************************************** -->
<!ELEMENT Height (#PCDATA)>
        <!ATTLIST Width Unit (MM|IN) #IMPLIED>
<!ELEMENT MinSize (Width, Height)>
<!ELEMENT MaxSize (Width, Height)>
<!ELEMENT Capacity (#PCDATA)>
<!ELEMENT BayID EMPTY>
        <!ATTLIST BayID Enum
      (bayA|bayB|bayC|bayD|bayE|bayF|bayG|bayH|ClassicMain|ClassicUpper|ClassicLower|
                    Unknown) #REQUIRED>
<!ELEMENT FeederType EMPTY>
        <!ATTLIST FeederType Enum (1K_Drawer|4.5KDrawer|RollFeed|Unknown) #IMPLIED>
<!ELEMENT DoorNumber (#PCDATA)>
<!ELEMENT SubstrateSupply (SupportedSizeList?, MinSize,MaxSize,Capacity,BayID,DoorNumber,FeederType?,
                                    SupportedEngineSpeedList?)>
        <!ATTLIST SubstrateSupply ID CDATA #REQUIRED>
<!ELEMENT SubstrateSupplyList (SubstrateSupply*)>

<!-- ************************************************************************
    * Define the remaining elements that make up a Font Element.
    ********************************************************************** -->
<!ELEMENT FontType EMPTY>
        <!ATTLIST FontType Enum (PostScriptType1|PostScriptType1MM|
                        PostScriptType3|PostScriptType42|
                        PostScriptTTF|PostScriptType0CID)
                        #IMPLIED
                    Present (True|False) #IMPLIED>
<!ELEMENT CoreFont EMPTY>
        <!ATTLIST CoreFont Present (True|False) #IMPLIED>

<!-- ************************************************************************
    * Define the Font element as an aggregate element, and the FontList
    * element as a list of Font elements.  Allow a font list to specify
    * default font settings.
    ********************************************************************** -->
<!ELEMENT Font (Name,Comment?,FontType?,CoreFont?)>
<!ELEMENT FontDefaults (Font)>
<!ELEMENT FontList (FontDefaults?,Font*)>


<!-- ************************************************************************
    * Define the remaining elements that make up the ImpositionTemplate
    * Element.
    ********************************************************************** -->
<!ELEMENT PagesPerSheet (#PCDATA)>
<!ELEMENT SheetsPerSignature (#PCDATA)>

<!ELEMENT LayoutMode EMPTY>
        <!ATTLIST LayoutMode Enum (Simplex|Duplex) #IMPLIED
                    Present (True|False) #IMPLIED>

<!-- ************************************************************************
    * Define the ImpositionTemplate element as an aggregate element,
```

```
    * and the ImpositionTemplateList element as a list of ImpositionTemplate
    * elements.  Allow an imposition template list to specify default
    * imposition template settings.
    ********************************************************************** -->
<!ELEMENT ImpositionTemplate (Name,Comment?,PagesPerSheet?,SheetsPerSignature?,
                        LayoutMode?)>
<!ELEMENT ImpositionTemplateDefaults (ImpositionTemplate)>
<!ELEMENT ImpositionTemplateList (ImpositionTemplateDefaults?,
                          ImpositionTemplate*)>

<!-- **********************************************************************
    * Define the ICCProfile element as an aggregate element,
    * and the ICCProfileList element as a list of ICCProfile
    * elements.
    ********************************************************************** -->
<!ELEMENT ICCProfile (Name,ColorSpace?)>
        <!ATTLIST ICCProfile SubstrateICCProfile (True|False) #IMPLIED>
<!ELEMENT ICCProfileDefaults (ICCProfile)>
<!ELEMENT ICCProfileList (ICCProfileDefaults,ICCProfile*)>

<!-- **********************************************************************
    * Define the ResourceCheckProfileList element as an aggregate element.
    ********************************************************************** -->
<!ELEMENT ResourceCheckProfile (Name)>
<!ELEMENT ResourceCheckProfileList (ResourceCheckProfile*)>

<!-- **********************************************************************
    * Define the ScreeningSystems element as an aggregate element.
    ********************************************************************** -->
<!ELEMENT ScreenType EMPTY>
        <!ATTLIST ScreenType Enum (Classic|Optimum|Line|
                           Supra|Stochastic|None|Other) #IMPLIED>
<!ELEMENT ScreeningSystems (ScreenType*)>

<!-- **********************************************************************
    * Define the SpotColorTableList element as an aggregate element.
    ********************************************************************** -->
<!ELEMENT SpotColorTable (Name)>
<!ELEMENT SpotColorTableList (SpotColorTable*)>

<!-- **********************************************************************
    * Define the Workflow element as an aggregate element.
    ********************************************************************** -->
<!ELEMENT StopPoint EMPTY>
        <!ATTLIST StopPoint Enum (StopAfterSubmission|StopWhenPDFAvailable|
                          StopAfterResourceCheck|StopAfterRIP) #REQUIRED>
<!ELEMENT StopPointList (StopPoint*)>
<!ELEMENT Trap EMPTY>
        <!ATTLIST Trap Enable (True|False) #REQUIRED>
<!ELEMENT Trapping (Trap*)>
<!ELEMENT Workflow (StopPointList,Trapping)>

<!-- **********************************************************************
    * Define the SystemDefaults element and its sub-elements.
    ********************************************************************** -->
<!ELEMENT CMYKGraphicICCProfile (#PCDATA)>
<!ELEMENT CMYKImageICCProfile (#PCDATA)>
<!ELEMENT RGBGraphicICCProfile (#PCDATA)>
<!ELEMENT RGBImageICCProfile (#PCDATA)>
<!ELEMENT JobTicketTemplate (#PCDATA)>

<!ELEMENT SystemDefaults (CMYKGraphicICCProfile,CMYKImageICCProfile,
                    RGBGraphicICCProfile,RGBImageICCProfile,
```

```
                    JobTicketTemplate?)>

<!-- ***************************************************************************
    * Define the elements that make up the FinishingOptions Element.
    *************************************************************************** -->
<!ELEMENT CoverType EMPTY>
        <!ATTLIST CoverType Enum (None|Front|Back|Both) #IMPLIED
                    Present (True|False) #IMPLIED>

<!ELEMENT CoverList (CoverType*)>

<!ELEMENT Inserts EMPTY>
        <!ATTLIST Inserts Present (True|False) #IMPLIED>

<!ELEMENT InsertList (Inserts*)>

<!ELEMENT Collate EMPTY>
        <!ATTLIST Collate Enable (True|False) #IMPLIED>

<!ELEMENT Collation (Collate*)>

<!ELEMENT Jog EMPTY>
        <!ATTLIST Jog Enable (True|False) #IMPLIED>

<!ELEMENT Jogging (Jog*)>

<!ELEMENT DeliveryTypeDetails SupportedEngineSpeedList?>
        <!ATTLIST DeliveryTypeDetails Enum (Deposit|Rotary|HighVolumeTray|
                                LowCapacityProofWaste|LowCapacityProof|
                                Vertical|Dfa|HighCapacity|ProofWaste|
                                Invalid|Unknown) #REQUIRED
                Position CDATA #REQUIRED>

<!ELEMENT DeliveryType (DeliveryTypeDetails)>
        <!ATTLIST DeliveryType Enum (stacker|proof|waste|finisher|unknown) #REQUIRED>

<!ELEMENT OffsetCapable EMPTY>
        <!ATTLIST OffsetCapable Enum (True|False) #REQUIRED>

<!ELEMENT StapleCapable EMPTY>
        <!ATTLIST StapleCapable Enum (True|False) #REQUIRED>

<!ELEMENT OperatorSelectable EMPTY>
        <!ATTLIST OperatorSelectable Enum (True|False) #REQUIRED>

<!ELEMENT OutputDelivery EMPTY>
        <!ATTLIST OutputDelivery Enum (SheetFlip|NoSheetFlip) #REQUIRED>

<!ELEMENT Delivery (DeliveryType, DeliveryLocation, OffsetCapable,StapleCapable, InsertCapable,
      OperatorSelectable, OutputDelivery, SupportedSizeList?, MinSize,MaxSize, DefaultDestination?, PunchPattern?,
      SupportedEngineSpeedList?)>
        <!ATTLIST Delivery ID CDATA #REQUIRED>

<!ELEMENT SubstrateDeliveryList (Delivery*)>

<!ELEMENT PageOrder EMPTY>
        <!ATTLIST PageOrder Enum (FaceUp|FaceDown|
                        FaceUpReverseOrder|FaceDownReverseOrder) #REQUIRED>

<!ELEMENT PageOrderList (PageOrder*)>

<!-- ***************************************************************************
    * Define the FinishingOptions element as an aggregate element.
```

```
    ************************************************************************ -->
<!ELEMENT FinishingOptions (CoverList?,InsertList?,Collation?,Jogging?,
                    SubstrateDeliveryList?,PageOrderList?)>

<!-- ************************************************************************
    * Define the ManufacturingCapabilities element as an aggregate element.
    ************************************************************************ -->

<!ELEMENT ManufacturingCapabilities (DeviceID?,PrinterList?,MediumList?,
                            SubstrateSupplyList?,FontList?,
                            ImpositionTemplateList?,ICCProfileList?,
                            ResourceCheckProfileList?,ScreeningSystems?,
                            SpotColorTableList?,Workflow?,
                            SystemDefaults?,FinishingOptions?)*>
```