Layout Creator to Imposition ICS

Version 1.4

Date: 2009-12-22

File: *ICS-LayCrImp-1.4.doc*, *.pdf*

Origination & Prepress WG

Abstract

This ICS defines the interface between a layout creation tool and a Consumer of the fully-populated **Layout** Resource that it produces. The layout creation tool may be entirely manually driven, or it may be controlled using the JDF ***Stripping*** or ***LayoutPreparation*** Processes. Example Consumers include a tool that produces an imposed content file, or a Device or prepress workflow system that performs a ***RIPing*** Process including the ***Imposition*** Process. When the Consumer is a prepress workflow system, it will be configured in such a way that the JDF instance supplied by the imposition design tool is not merged with JDF from other sources.

The target of Level 1 is mainly RIPs. Level 2 is for workflow systems.

Level 2 of the ICS adds extra information that a workflow system could use (e.g. meta info for double sided proofing). Also, it allows the content to be delivered later, and it is the base for validation for the Detailed-Layout Imposition Profile of [MISPRE-ICS].

For cases where the Consumer (e.g. prepress workflow system) accepts a JDF instance containing an unpopulated, or partially Populated **RunList** from an imposition design tool, or where that JDF instance is merged with JDF data from elsewhere, see alternative ICS documents that may be available from CIP4.

This version applies to interactions using [JDF1.4a].

Copyright Notice

Licenses and Trademarks

Table of Contents

1 Introduction 4
2 Glossary 4
3 Conformance Levels 4
3.1 Certification 5
4 Conformance Tables – JDF Instances 5
4.1 JDF Node 5
5 Conformance Tables – Processes 6
5.1 Imposition 6
6 Conformance Tables – Resources 6
6.1 BindingIntent 7
6.2 DeviceMark 7
6.3 FileSpec 8
6.4 JobField 9
6.5 Layout 9
6.5.1 Abstract PlacedObject 11
6.5.1.1 ContentObject 12
6.5.1.2 MarkObject 12
6.5.2 SourceResource 13
6.6 LayoutElement 14
6.7 Media 15
6.8 RunList 15
6.9 TransferCurvePool 16
6.9.1 TransferCurveSet 16
7 References 17
7.1 Normative References 17
7.2 Informative References 17

Tables

Table 1: Glossary 4
Table 2: Conformance Levels 5
Table 3: JDF Node 5
Table 4: Imposition – Input Resources 6
Table 5: BindingIntent 7
Table 6: DeviceMark 7
Table 7: FileSpec 8
Table 8: JobField 9
Table 9: Layout 9
Table 10: Abstract PlacedObject 11
Table 11: ContentObject 12
Table 12: MarkObject 13
Table 13: SourceResource 13
Table 14: LayoutElement 14
Table 15: Media 15
Table 16: RunList 15
Table 17: TransferCurvePool 16
Table 18: TransferCurveSet 16

1 Introduction

The creation of imposition layouts is often performed by specialist Products. Such Products may be stand-alone, or may be entirely and transparently embedded within a prepress workflow system.

Whether such tools are stand-alone, they are used in two different ways:

- To supply a JDF instance that contains one or more **Layout** Resources, and possibly one or more **RunList** Resources, but where those **RunList** Resources are not fully ***Populated***. Such a file would normally be consumed by a prepress workflow system that uses it as a template for Job data obtained from other sources, or merges it with other JDF Elements from other sources.
- To supply a JDF instance containing one or more **Layout** Resources and one or more associated **RunList** Resources that fully identify all of the content Pages and marks that will be used to fill them. Such a JDF instance is suitable for immediate submission to a Device that performs the ***Imposition*** Process.

This ICS applies only to the latter case where fully-***Populated*** **RunList** Resources are required.

2 Glossary

This section defines terminology used throughout this document. References to other documents are indicated with square brackets, e.g. [JDF1.4a]. For most terms, see the Glossary section in [Base-ICS].

This section contains terms that pertain to this ICS:

Table 1: Glossary

Term	Definition
Populated **RunList**	A **RunList** whose RunList/LayoutElement/FileSpec/@URL Attribute references a file that holds either content Pages or marks.

3 Conformance Levels

This ICS, the *[LayCrImp-ICS]*, defines two Conformance Levels, namely Level 1 and 2.

See Appendix A "*How to Read ICS Documents*" in [Base-ICS] for an explanation of Conformance Tables.

Level 1 targets RIPs

Level 2 targets workflow systems and adds extra meta information (needed for double sided and decomposed proofing)

While the Producer in the interface described by this ICS will typically be a JDF Device, the Consumer of the JDF instance so produced may be either another Device or a Controller. In most cases, the Consumer will be a Device, but that behavior is not mandated by this ICS. All descriptions and tables below therefore refer to Managers and Workers.

If a JDF instance in isolation conforms to this ICS, there is no guarantee that the complete file set (comprising the JDF instance plus referenced Page description files) will also conform to this ICS. See FileSpec/@URL in Table 7: FileSpec.

To be conformant to a level of this ICS specified in the first column of Table 2, a Manager MUST conform to the Manager part and a Worker MUST conform to the Worker part of the ICSs and levels specified in Table 2 below.

Table 2: Conformance Levels

Level of this ICS	[Base-ICS]	[JMF-ICS]	[MIS-ICS]	Description
1	0	-	-	Info needed for a RIP
2	0	-	-	Adds info for workflow system

3.1 Certification

The read test for the "r" Attributes are evaluated on the output by comparing to a reference output. A high-resolution TIFF file is acceptable instead of the physical output.

4 Conformance Tables – JDF Instances

4.1 JDF Node

Table 3 specifies the JDF Node structures Supported by this ICS. They include a stand-alone ***Imposition*** Process and a Process Group (also Gray Box) that includes an ***Imposition*** Process. Only JDF Nodes that have a *ICSVersions* Attribute containing `LayCrImp_L1-1.4` or `LayCrImp_L2-1.4` are covered by the next table.

In case of the "Detailed-Layout Imposition Method" as described by the [MISPRE-ICS], the Node is a ***GB ImpositionSoftProofing***, ***GB ImpositionProofing*** or ***GB PlateMaking***.

Table 3: JDF Node

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ICSVersions	w←			r?			
`LayCrImp_L1-1.4`	w←			r?			
`LayCrImp_L2-1.4`		w←		r?			
Type	w			r			.
`Combined`	w←			r			A Worker MUST accept a Combined Process where `contains(@Types, "Imposition")`.
`Imposition`	w←			r			A Worker MUST accept a Node with a single ***Imposition*** Process.
`ProcessGroup`	w←			r			A Worker MUST accept a Process

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
							Group where contains(@Types, "Imposition"). The JDF Node MUST NOT contain any child Nodes.
Types	w←			r			MUST be specified if (@Type="ProcessGroup" or @Type="Combined").
Imposition	w			r			*Types* list MUST contain exactly one "*Imposition*" and MAY contain other Processes.

5 Conformance Tables – Processes

5.1 Imposition

Table 4: Imposition – Input Resources

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Layout	w			r			See Table 9: Layout.
RunList (Document)	w			r			See Table 16: RunList.
RunList (Marks)	w←			r			Manager MUST write if the Subelements of the **Layout** Resource contain any MarkObject Elements with *Ord* Attributes. See Table 16: RunList.

6 Conformance Tables – Resources

Manager MUST set the *Status* Attribute of all Input Resources (direct or indirect) of the ***Imposition*** Process to have the value of *Available*.

6.1 BindingIntent

Table 5: BindingIntent

Referenced by: SourceResource

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
BindingOrder		w			r?		
Collecting							Saddle stitched.
Gathering							Perfect bound.
None							Flat work.
BindingSide/@Actual		w			r?		
Left							
Right							
BindingType/@Actual		w			r?		
SaddleStitched							Saddle stitched.
SoftCover							Mapped to Perfect bound.

6.2 DeviceMark

Table 6: DeviceMark

Referenced by: MarkObject

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Font	!w			r?			Explicit selection of a *Font* makes handling of localized text in different scripts difficult.
FontSize	w?			r			Default="9".

6.3 FileSpec

Table 7: FileSpec
Referenced by: LayoutElement

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Compression	w?			r?			*Compression* other than "*None*", the Default, are outside the scope of this ICS.
None	w			r?			
MimeType	w			r?			
URL	w			r			When RunList[LayoutElement/FileSpec] is an Input Resource to the ***Imposition*** Process, the Manager MUST write *URL* values that can be resolved by the Worker to the page description language files to be used. See [FileURL-AN] for more discussion.
"" (empty URL)	!w			r?			If a ContentObject or MarkObject is blank, the Manager MUST specify such with LayoutElement/@IsBlank ="*true*" and MUST NOT specify it with LayoutElement/FileSpec/@URL =""".
./…	w←			r			This value represents any Relative ***URL*** that is relative to the Hot Folder. Relative ***URLs*** MUST be Supported relative to a "file" scheme and MAY be Supported relative to other schemes.
file:…	w←			r			Any ***URL*** whose scheme is "file", except as restricted below.
file://localhost/...	!w			r?			A ***URL*** MUST be resolvable by the Worker. A localhost ***URL*** can only be resolved by a Worker that is running on the same computer as the Manager, and which therefore sees the file system from the same viewpoint. Note also that many early Products based on Mac OS/X create badly constructed ***URLs*** for files stored on other computers, starting "*file://localhost/Volumes/*", or "*file:///Volumes/*".
file:///...	!w			r?			See above row about localhost.
all remaining values	w?			r?			

6.4 JobField

The **JobField** specifies content only. Its format depends on the Worker implementation.

Table 8: JobField

Referenced by: MarkObject

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
OperatorText	w←			r			Manager MUST write if *ShowList* includes `OperatorText`.
ShowList	w			r			
`DeviceID`	w?			r			
`OperatorText`	w?			r			
`StartTime`	w?			r			
`UserText`	w?			r			
all remaining values	w?			r?			A Worker MAY ignore any of these values and it MUST NOT treat any as an error condition.
UserText	w←			r			Manager MUST write if *ShowList* includes `UserText`.

6.5 Layout

Table 9: Layout

Input to: *Imposition*

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Automated	!w			r?			Disallow automated **Layout** Resources. (PJTF: PL). That is, allow only fully specified **Layout** Resources.
PartIDKeys	w			r			Manager MUST write into the outermost **Layout** Element of the Resource. When a Manager supplies *PartIDKeys*, the order of Partition Keys in its value (other than those marked with a "!w" value) MUST follow the order shown in the following table rows. The Manager MUST include all three Partition keys.
`SignatureName`	w			r			
`SheetName`	w			r			

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Side	w			r			Jobs intended to be printed on only one side MUST include only a single *Side* Partition within each *SheetName* Partition. Work & Turn and Work & Tumble Jobs MUST include only a single *Side* Partition within each *SheetName* Partition, and those *Side* Partitions MUST set *Side*= "*Front*".
all remaining values	!w			r?			
SourceWorkStyle	w?	w		r?	r?		Only allowed in **Layout** *SheetName* Partition.
SurfaceContentsBox	w←			r			Manager MUST write *SurfaceContentsBox* in enough **Layout** *Side* or *SheetName* Partitions so that each **Layout** *Side* Partition either contains this Attribute, or inherits it from its immediate parent (*SheetName*) Partition. The first two numbers in the rectangle MUST be 0, 0.
ContentObject	w←			r			Manager MUST write a ContentObject in a **Layout** *Side* Partition if it places content Pages from input files on the Surface. Manager MUST NOT write a ContentObject into a **Layout** *SignatureName* or *SheetName* Partition. Manager MUST write at least one PlacedObject in a **Layout** *Side* Partition. Each PlacedObject MUST be either a ContentObject or a MarkObject. See Table 11: ContentObject.
MarkObject	w←			r			Manager MUST write a MarkObject in a **Layout** *Side* Partition if it places marks other than content Pages on the Surface. Manager MUST NOT write a MarkObject into a **Layout** *SignatureName* or *SheetName* Partition. Manager MUST write at least one PlacedObject in a **Layout** Element Partitioned by *Side*. Each PlacedObject MUST be either a ContentObject or a MarkObject. See Table 12: MarkObject.
Media (Plate)	w?			r?			Manager SHOULD provide this Resource if it has the appropriate information. See Table 15: Media.

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Media (Paper)	w?			r?			Manager SHOULD provide this Resource if it has the appropriate information. See Table 15: Media.
SourceResource		w			r?		Points to binding information. See Table 13: SourceResource.
TransferCurvePool	w?			r?			Manager SHOULD provide this Resource if it provides one or more TransferCurveSet Elements as specified in Table 17: TransferCurvePool. This Resource provides the relationship between the coordinate system of the **Layout**, the physical printed **Media**, and any intermediate **Media** such as film or plates, to enable in-line plate-punching, etc. See Table 17: TransferCurvePool.

6.5.1 Abstract PlacedObject

The descriptions of *CTM* and *SourceClipPath* in [JDF1.4a] refer to the "the native coordinate system of the object" and "the coordinates of the source Page" respectively. In both cases the lower left of the object (**MediaBox** in case of PDF, the whole of the raster for TIFF, etc) is used as the reference; X values increase to the right, and Y values increase towards the top. For those source file formats that define tags to indicate extra transformations, a Consumer conforming to this ICS MUST apply these transformations before identifying the lower left of the object. For example, the rotation defined by the PDF Rotate key MUST be applied first; or the orientation defined by the TIFF Orientation tag MUST be applied first. The Consumer MUST determine the location of the lower left corner before any transformation is applied by the *CTM*. The Consumer MUST apply all translation, scaling and rotation defined by *CTM* to the *SourceClipPath*.

A Worker conforming to this ICS MUST process the PlacedObject Elements (i.e., ContentObject and MarkObject Elements) in such a way that the final rendered appearance of the Surface concurs with the order of those Elements within the **Layout** *Side* Partition. Thus when two graphic objects defined in different PlacedObject Elements (i.e. MarkObject Elements or ContentObject Elements) overlap each other, the one specified later in the **Layout** *Side* Partition will appear to knock out or overprint the one specified earlier, depending on whether the graphic element is set to overprint or not.

Table 10: Abstract PlacedObject

Superclass of: ContentObject, MarkObject

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ClipBox	w?			r			
ClipPath	w?			r			Note that this is new in [JDF1.4a].

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
CTM	w			r			See notes above.
SourceClipPath	w?			r			See notes above.

6.5.1.1 ContentObject

See Section 6.5.1 *Abstract PlacedObject* for additional information.

Table 11: ContentObject
Referenced by: Layout
Subclass of: Abstract PlacedObject

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
DocOrd	!w			r?			
Ord	w			r			*Ord* MUST NOT be equal to or greater than the total number of Pages in the document **RunList**.
OrdExpression	!w			r?			
SetOrd	!w			r?			

6.5.1.2 MarkObject

A MarkObject may contain a number of Attributes and Elements, such as **CIELABMeasuringField**, **RegisterMark**, etc, alongside graphical content, referenced via the *Ord* index into the **RunList**. Such Attributes and Elements may provide metadata about the marks that could be useful to downstream closed loop control systems for color or registration, but a Worker conforming to this ICS is not required to act on them directly.

See Section 6.5.1 *Abstract PlacedObject* for additional information.

Table 12: MarkObject
Referenced by: Layout
Subclass of: Abstract PlacedObject

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Ord	w←			r			The Manager MUST write either *Ord* or **JobField**, but not both. *Ord* MUST NOT be equal to or greater than the total number of Pages in the marks **RunList**. The Manager MAY specify additional information to place on a Sheet by using the *Ord* value to reference text, icons etc. on a Page of a marks **RunList**.
DeviceMark	w←			r			See Table 6: DeviceMark.
DynamicField	!w			r?			See [JDF1.4a].
JobField	w←			r?			The Manager MUST write either *Ord* or **JobField**, but not both. See Table 8: JobField.

6.5.2 SourceResource

Table 13: SourceResource
Referenced by: Layout

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
BindingIntentRef		w			r		Refers to **BindingIntent** Resource with more details, See Table 5: BindingIntent.

6.6 LayoutElement

Table 14: LayoutElement
Referenced by: RunList

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ElementType	w?			r?			This Attribute is present in this table to specify that a Manager MUST NOT include the value below if it supplies this Attribute.
"Reservation"	!w	w?		r?			The **RunList** (Document) row of Table 6-90: "Stripping – Input Resources" in Section 6.4.37 "Stripping" of [JDF1.4a] explains that this value is not suitable for a **RunList** that is an Input Resource of an ***Imposition*** Process. Note: a Manager MUST denote that a ContentObject is blank by using *IsBlank* (see below), and not by using *ElementType* = "*Reservation*".
IsBlank	w←			r			Manager MUST write either **FileSpec** or *IsBlank* = *"true"*, but not both. Manager MUST write *IsBlank* = *"true"* if and only if the corresponding MarkObject or ContentObject contains no graphical data.
FileSpec (MarkObject)	w←			r			Manager MUST write either **FileSpec** or *IsBlank* = *"true"*, but not both. Manager MUST write **FileSpec** if and only if the corresponding MarkObject or ContentObject contains graphical data. See Table 7: FileSpec.
FileSpec (ContentObject)	w←	w?		r			For level 1: Manager MUST write either **FileSpec** or *IsBlank* = *"true"*, but not both. Manager MUST write **FileSpec** if and only if the corresponding MarkObject or ContentObject contains graphical data. For level 2: Manager MUST write **FileSpec** if the Content data is available and contains graphical data. Manager MUST NOT write both **FileSpec** and *IsBlank* = *"true"*. See Table 7: FileSpec.

6.7 Media

Table 15: Media
Referenced by: Layout

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Dimension	w			r?			
MediaType	w			r?			Manager MUST supply one of the values below, allowing transformations of coordinate systems from that of the **Layout** using the *CTM* in TransferCurveSet Elements.
Film	w←			r?			
Paper	w←			r?			
Plate	w←			r?			

6.8 RunList

Table 16: RunList
Input to: *Imposition*

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
NPage	w			r			Manager MUST write a value for *NPage* that: • **if** *Pages* **is present**, is equal to the number of Pages listed in *Pages*, , • **otherwise if** *FirstPage* **is present**, should be equal to the number of Pages in the referenced **LayoutElement** as specified by *FirstPage* and *SkipPage*, • **otherwise**, should not exceed the number of Pages in the referenced **LayoutElement**
ByteMap	!w			r?			Manager MUST reference common raster formats that might be used in post-RIP imposition (e.g. TIFF) via **LayoutElement**, not **ByteMap**, which is a JDF-specific format. See [JDF1.4a].
InsertSheet	!w			r?			See [JDF1.4a].
InterpretedPDLData	!w			r?			See [JDF1.4a].
LayoutElement	w			r			See Table 14: LayoutElement.

6.9 TransferCurvePool

Table 17: TransferCurvePool
Referenced by: Layout

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
TransferCurveSet	w?			r			Manager SHOULD supply one TransferCurveSet Element for each value of TransferCurveSet/@Name listed in this ICS where the corresponding TransferCurveSet/@CTM would not be the identity matrix. See Table 18: TransferCurveSet.

6.9.1 TransferCurveSet

Table 18: TransferCurveSet
Referenced by: TransferCurvePool

Name	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
CTM	w			r			The first 4 numbers of the *CTM* Attribute Value MUST match one of four values as defined by *Rotate0*, *Rotate90*, *Rotate180* and *Rotate270* in [JDF1.4a] Table 2-4 “Matrices and Orientation values for describing the orientation of a Component”.
Name	w			r			The Supported values are the same as for Media/@MediaType. That is, they are the values of *Name* that overlap with Media/@MediaType.
Film	w←			r?			
Paper	w←			r?			
Plate	w←			r?			

7 References

7.1 Normative References

[Base-ICS] Base ICS, Version 1.4, published December 2009. Available at: http://www.cip4.org.

[JDF1.4a] JDF Specification, Version 1.4a, published December 17, 2009. Available at: http://www.cip4.org.

7.2 Informative References

[FileURL-AN] "CIP4 Application Note: Use of the File URL in JDF", 12 November 2003. Available at: http://www.cip4.org/intern/document_archive/document_details.php?did=546.

[JMF-ICS] JMF ICS, Version 1.4, published December 2009. Available at: http://www.cip4.org.

[MIS-ICS] MIS ICS, Version 1.4, published December 2009. Available at: http://www.cip4.org.

[MISPRE-ICS] MIS to Prepress ICS, Version 1.4, published December 2009. Available at: http://www.cip4.org.


CIP4 THANKS ITS PARTNER LEVEL MEMBERS
Adobe
HEIDELBERG
RICOH | IBM
InfoPrint Solutions Company
manroland
RICOH
efi print to win.
HIFLEX
MIS · JDF · Web2Print
Business Automation Systems
hp
Kodak
MULLER MARTINI
xerox