Job Definition Format (JDF)

An Open, Multi-Vendor Solution

An overview of the new specification format developed by Adobe, Agfa, HEIDELBERG, and MAN Roland.

In the last few years, the demand for greater unification of mechanized and automated systems has led to the advent of specification formats. These formats spawned a generation of systems in the mid-1990's that began to provide the ability to link certain elements of the pre-press, press, and post-press processes. The formats, however, had inherent limitations and proprietary architectures. For example, they could address only certain aspects of a print job, they could do little to help in the authoring and revision processes, and they could not provide production automation systems with the ability to control and track jobs.

These formats, however, limitations and all, have paved the way for improving the existing architecture, and indeed for improving the graphic arts industry as a whole. In the last few years, there has been a call for the printing and publishing industries to create two things: a standards-based, supply-chain infrastructure and a set of protocols specific to the process of creating, manufacturing and distributing printed information. Four companies prominent in the graphic arts industry—Adobe, Agfa, HEIDELBERG, and MAN Roland—have united to create a new format that addresses these and other issues. The idea was to develop a set of open, extensible, XML-based job ticket standards, as well as a mechanism that provides new business opportunities for all individuals and companies involved in the process of creating, managing and producing published documents in the new economy. Building on the existing technologies of CIP3's Print Production Format (PPF) and Adobe's Portable Job Ticket Format (PJTF), the Job Definition Format (JDF) supplies a means for printing businesses to streamline the process of producing printed material.

The three most prominent features of JDF are its ability to carry a print job from genesis through completion, its ability to bridge the communication gap between production and Management Information Services, and its ability to do so under nearly any precondition. In short, JDF is an extremely versatile and comprehensive format. A business of any size can use JDF, from an individual with only a handful of jobs a year to a publication house with daily output. JDF provides a means to set up communication between systems from different vendors with minimum configuration efforts. It also facilitates the execution of every aspect of any print job, from creation through shipping.

This whitepaper provides an introduction to the Job Definition Format, and explains its place in today's pre-press and printing industry. From yesterday's world of mechanized labor to today's world of digital unification, the printing industry has sought to improve upon every advance and move towards the goal of increasing efficiency and versatility. The document elaborates on the background of the development of JDF, explains its specific capabilities, and provides practical examples of how JDF can be used.

Background

All four companies involved in creating this specification entered the project with strong credentials in the graphic arts industry. Adobe has developed key applications in the pre-press area (such as Adobe PageMaker, Adobe InDesign, and Adobe Photoshop) and has set the de facto standards PostScript® and PDF. Adobe has also defined the Job Ticket Format (PJTF) to describe the parameters of important prepress functions. Agfa is a significant player in the consumables and prepress market. Both HEIDELBERG and MAN Roland are the main players in printing presses. HEIDELBERG also works to develop both pre-press and postpress technologies, including automation and information systems, and MAN Roland additionally develops press-room automation and links business information systems to their presses in all areas of printing, from sheet-fed to newspaper.

All of these companies brought their experience and knowledge to the table when working to develop JDF, and all are well versed in the constraints placed on the printing process by current technological limitations. Despite the great strides made in the industry in the last century, some difficulties in the process have thus far proved insurmountable. These difficulties are often characterized as "islands in the stream" problems. This refers to the isolation of all of the diverse elements required to complete a printing job, and the problem has been around pretty much since more than one apparatus and one press operator were used to create a sheet of printed paper.

In the beginning, all was manual. Then came mechanization, which paved the way for automation. Automation permitted press operators to dictate in advance how each separate machine would function. Eventually, production controllers assembled systems involving machines with multiple capabilities run with modular automation, all controlled from a centralized console. The devices themselves were not connected, nor was the process a fluid one, but the possibilities were becoming more clear. Each advance spawned another opportunity for greater growth.

Once automation was in place, a job ticket format was the next logical step. With these formats, printers could begin to segment larger sections of the print job and run each more efficiently. CIP3's developed PPF to create a means for jobs to flow between multi-vendor products, so that cluster groups of machines could be controlled by a single job ticket. It handled a comprehensive variety of pre-press, press and post-press processes. Adobe's PJTF offered limited but very precise control of pre-press processes such as layout, imposition, trapping and RIPping as well as significant widespread control over a variety of press processes.

Despite these tremendous strides, however, some of the limitations faced by pre-mechanized printers remain in place today. No technological advance before the development of JDF had yet provided the print industry with the ability to counteract the "islands-in-a-stream" problem. In other words, although production facilities can attend to each individual element of a print job and can even link some of those processes, they are incapable of automating a system to run a job from the moment a customer places the order to the moment the final product emerges.

A significant reason that this continues to be a problem is that multi-vendor cooperation is still possible only on a very limited basis. It still takes significant effort to get machines manufactured by different companies to work together. The encoding capabilities of versatile languages such as XML have been recognized, but, so far, not fully exploited.

Furthermore, two of the biggest and most important islands in the printing stream are the ones that progress has so far had the least success in connecting. Management Information Systems (MIS), generally responsible for the planning and controlling of a job, and production services, responsible for the operation of a job, still toil in relative isolation from a communications standpoint. In other

words, there is no means for consistent, automatic, and effective bi-directional communication between the two indispensable facets of every printing business. All data enumerating planning and scheduling, process results, job status and job tracking must travel from production to MIS so that the latter can process the information and provide instructions on how to continue. This continues to be a manual procedure.

In short, what exists is an entire process that is comprised of multiple machines and functions, all necessary for the completion of a job, but incapable working as a coherent unit. Most industries would find this kind of fragmentation unthinkable in today's age of digital unification because it is inefficient, a cardinal sin in the business world.

JDF provides not only a solution, but a flexible and comprehensive one. It is capable of creating a bridge between each separate island in the printing stream, from the moment a customer places an order to the moment the finished product is placed in the customer's hands, regardless of how many manufacturers contributed machinery to the apparatus, or how complex the task. It can also link the two separate strata essential to the completion of each print job: MIS and production.

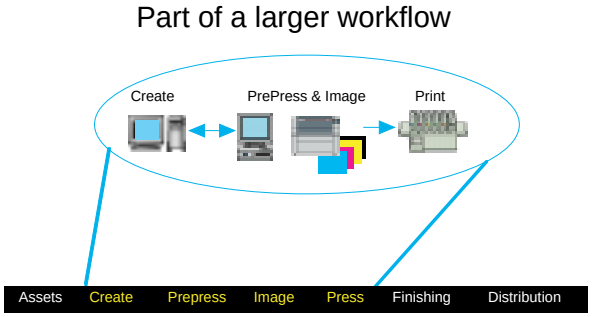

Figure 1. Part of a larger workflow

The advantages of a system capable of uniting the disparate and diverse elements of a printing job are probably obvious. The more control you have over the mechanical workings of the process, the more able you'll be to focus on the quality of the jobs you produce. No less important is that increased efficiency creates faster results. All customers value speed as long as quality is not sacrificed.

What is JDF?

The most significant capabilities of JDF can be divided into three principal categories: its ability to complete every part of a job, from start to finish; its ability to link MIS with production; and its ability to perform both of the first two tasks no matter what tools are used. The following sections provide detail about each of these categories.

JDF provides a mechanism to control all of the processes in print production

Unlike other job ticket formats, JDF allows the description of all the processes needed to complete a print product, from job submission through pre-press, press and post-press. It does this by translating each process step in a job into what is called a node. The entire job is represented by a tree of these nodes. All of the nodes taken together describe the desired printed product and the workflow of its production.

Each individual node—that is, each process—is defined in terms of inputs and outputs. The inputs for a process consist of the resources it uses and the parameters that control it. Inputs in a node describing the process for creating the cover of a brochure, for example, might include the inks, the press sheets, the plates, and a set of parameters that indicate how many sheets should be produced. The output of the process node using these particular inputs will be a set of printed press sheets.

Resources produced by one process, however, are modified or consumed by subsequent processes. Therefore, the output described above—the printed sheets—become the input resource for some set of finishing operations, such as folding and cutting. And the finished sheets that are the output of those operations become the input resource for further post-press processes—binding, for example.

So the entire job is defined as a hierarchical network of processes that are linked through the consumption of inputs and production of outputs, which in turn become the inputs for further processing. The end result is the combination of outputs that produce the desired finished product. JDF provides the ability to place individual actions into a greater context, so that each element is regarded by the structure as a part of the whole. For more detail on this topic, see the section entitled *A Practical Example*.

JDF provides a mechanism to allow production automation systems (MIS) to control and track jobs

Essentially, JDF supplies a messenger service to run between MIS and production. As each process in a job executes, the results are recorded into the job to facilitate tracking each aspect. In addition, JDF defines a messaging architecture, which includes message formats, semantics, and message protocols. JDF devices use this architecture to communicate with systems that control print production facilities. System vendors therefore are given a great deal of flexibility in terms of how they choose to use the messaging architecture, and whether they provide both notification and control facilities. Figure (2) depicts how various communication threads might run.

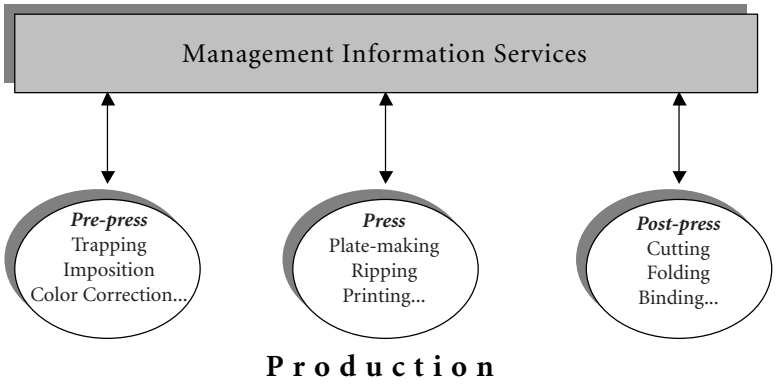

Figure 2. Communication between MIS and production

The system is set up in the following way: a job is defined in a network of nodes that each describes a process. Processes are executed by devices, and devices are grouped together in work centers. A print production control system controls the devices in a work center. The system communicates with the devices to control the flow of jobs through the work center to monitor the process of individual jobs and to detect and correct exception conditions. JDF's role is to specify a messaging architecture to enable communication between the controller and the devices in the work center. The architecture defines a standard set of messages, a format for all messages, and a set of protocols which devices can implement.

Furthermore, a range of messaging capabilities is provided. The minimum available capability is no messaging. When no messaging is selected, the controller must examine the JDF to determine

the results of processing. Independent of messaging capabilities, JDF includes different Audit records for each process that provide details of the planned and actual results of the process.

Most devices, however, will choose to support some level of messaging capability. The most basic level of support is 'notification'. Devices that support notification provide uni-directional messaging. They inform the controller when they begin and complete processing of some procedures within a job. They may also provide notification of some error conditions.

The next level of communication supports queries. Devices that support queries respond to requests from the controller by communicating status, such as current JobIDs, queued JobIDs, or current job progress. Queries require bi-directional communication.

Finally, devices may support commands. If command capability is selected, the controller can issue a directive to interrupt the current job, to restart a job, or to change the priority of jobs in the queue.

Besides messaging, another option JDF provides to controllers is the ability to collect performance data for each process and pass that information to a job-tracking system for use by the job accounting system. This information may be derived from the messages that the controller receives, or from the Audit records in the job. Alternatively, the completed job may be passed to the job accounting system, which examines the Audit records for itself to determine the costs of all the processes in the job. Each individual project manager selects the option or set of options for each individual job.

JDF is a vendor-independent standard

While the first version of JDF is being developed, the four companies involved intend to pass control of the specification to a cross-industry consortium. This will ensure that all vendors can develop systems that use JDF, and that no vendor is disadvantaged with respect to any of its competitors.

To this end, JDF will be encoded in XML, a standard controlled by the World-Wide Web Consortium (W3C). Additionally, features of XML have been chosen to allow easy extension of the specification to support processes and devices not anticipated in version 1.0 of the specification.

A Practical Example

Now that you have a general understanding of what JDF has to offer, it might help you to envision a scenario with JDF at work. This example provides a detailed description of how a typical job is represented in JDF. In addition, it describes how progress and results data are communicated between a production system and an MIS system.

The job

The job in this example is fairly simple—a 12-page product-marketing brochure. The cover is printed on a different stock than the inside pages, using different inks. The brochure will be stapled.

Figure (3) depicts the hierarchy of JDF nodes that comprise this job.

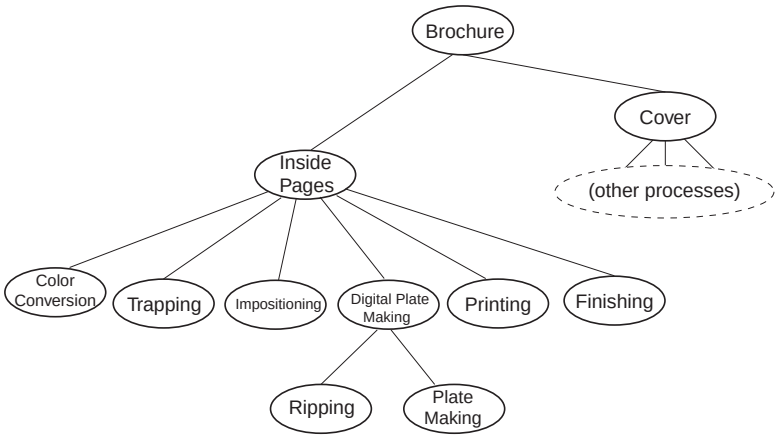

Figure 3. A tree of nodes

Process Details

Figure (4) depicts the details of one set of process nodes that describe the creation of plates for the inside pages of the brochure. There are three nodes in this part of the job.

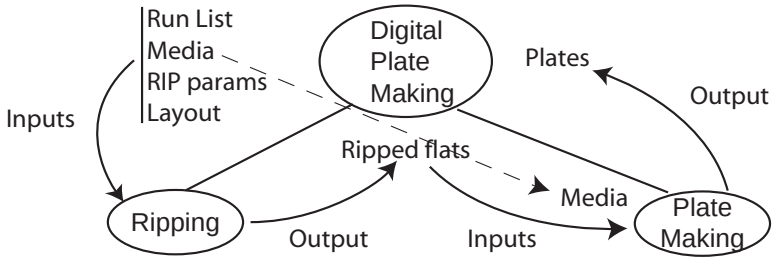

Figure 4. A subsection of the larger tree of nodes, depicting inputs and outputs

The parent node, described as 'Digital Plate Making', represents the entire process of taking the files, which have been generated through various pre-press processes, and using them to create the plates, which will subsequently be mounted on the press for printing.

The first child node, described as 'RIPping', represents the process of RIPping the pages and simultaneously imposing them onto the 'digital flats' that will be the input for the plate-making process.

The RIPping node has the following input resources:

- A runlist resource that describes the pages. This resource identifies the set of pages to be RIPped. It identifies the files in which they occur, and which pages are to be used.
- A media resource that describes the media which will be used to make plates. This is necessary to dictate the dimensions of the media.
- A RIP-parameters resource that describes all the device-specific parameters of the RIPping process.
- A Layout resource that describes the placement of the source pages onto the plates, and, eventually, onto the press sheets.

The RIPping node has a single output resource: the digital flats.

The second child node, described as 'Plate-Making', represents the process of producing the press plates. The Plate-Making node has two input resources:

- The digital flats. These are the same flats that were the output resource of the RIPping node.

- A Media resource which describes the media to be used. In this case, the media resource is used to identify the physical resource to be loaded into the plate-maker.

The Plate-Making node has a single output resource: the press plates.

Note how these nodes are related to nodes that were executed before 'Plate-Making', and to others that will occur subsequently.

One of the inputs to the RIPping node, the runlist resource, identifies a set of pages that occur in some file or files. These files were processed previously, possibly for color correction, possibly to insert trapping.

Another resource, the Layout resource, is created by an imposition tool. The tool may be able to apply a pre-defined template to the job, or an operator may specify the layout. In either case, the imposition tool adds the resource to the job, allowing the RIPping node to perform In-RIP imposition.

Finally, the output of the Plate-Making node becomes an input to a subsequent node, identified as 'Printing'. This node completes the process for the example shown.

Interaction between MIS and production

Throughout the execution of the processes needed to build the brochure, devices communicate with the MIS system. For this discussion, it is assumed that the devices support only notification. Obviously, additional messages might be exchanged between devices and the MIS system if those devices supported queries, or queries and commands.

Figure (5) depicts the chronological progress of the job through the system as the 'Digital Plate Making' node, and its children are executed with time increasing from left to right.

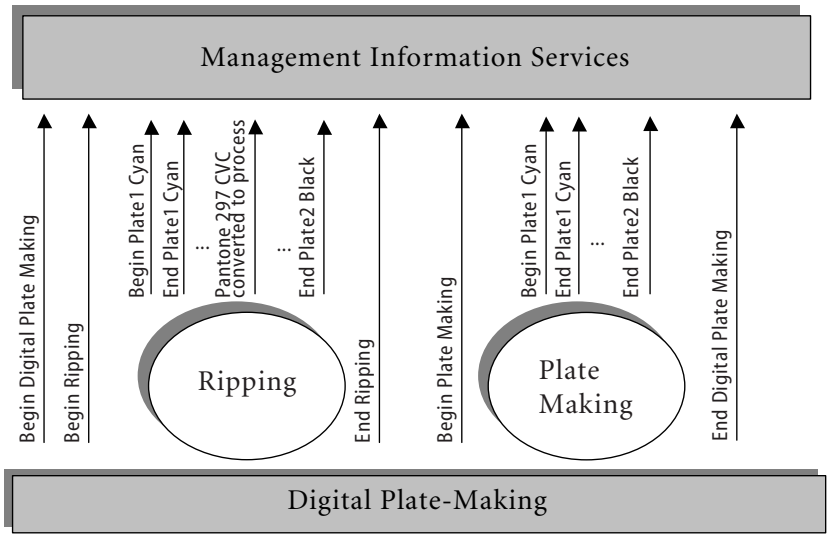

Figure 5. Uni-directional communication between MIS and production

Note that for each node, and for each device, there are pairs of notifications indicating the beginning and end of the processing for that node. In addition, there are notifications for each of the color plates produced during the RIPping process, and again during Plate Making.

Finally, note that the RIPping process generated a notification when one of the pages contained a Separation colorspace for a colorant that was not requested in the Separations resource. Depending on system configuration, this condition could have been an error, or an exception that would be identified during job preflight. In the example here, the colorspace was imaged in its alternate colorspace, and the job was executed to completion.

The level of detail provided in the notifications shown here is meant to be illustrative only. Devices may provide greater or lesser degrees of detail in their notifications. In addition, devices may choose to provide additional detail in the Audit records they write into the process nodes when they complete or fail to complete.

Conclusion

The Job Definition Format is on its way. Before long, it will be possible for pre-press, press and post-press business of any size and configuration to execute jobs quickly, seamlessly, and easily, from imagination to realization. The process of controlling and overseeing each job will require conspicuously less effort.

JDF is comprehensive—it intends to provide a mechanism to the entire range of processes that comprise print production. It facilitates automated production control. And it is open—all vendors can benefit.

Most importantly, the customer wins. Systems will work together accurately and consistently to produce the product the customer wants. Production will be controlled so that production facilities can accurately estimate and fairly assess the costs of producing work for customers. Progress will be facilitated as vendors focus less energy on making systems work together and more energy on adding value to the product. And who knows what new evolution it will stimulate in its turn?