

# JDF: Smart Flexible Workflows

by James E. Harvey

Executive Director of CIP4 Organization and President of Media4theWorld, LLC

## A Short History of JDF

The International Cooperation for the Integration of Processes in Prepress, Press and Postpress Organization or “CIP4” is a not-for-profit association. CIP4 is established in Switzerland, it has no regular offices; rather is a global organization with representatives several countries.

Currently the CIP4 has about 250 members, but that number is hard to pin down, because the membership is constantly growing and two or three new members join every month. CIP4 lives and breathes over WebEx — there are approximately 20 technical and administrative working groups and some of them “meet” weekly via WebEx; hence, the pace that CIP4 has kept is rather impressive, but would not have been possible just a few years ago. There are approximately six face to face meetings every year at locations around the world, including the annual member meeting, which is always held in conjunction with GraphExpo.

CIP4 was formed in September of 2000. Its predecessor, CIP3, was formed by Heidelberg and was managed by the Fraunhofer Institute for Computer Graphics. CIP3 created the Print Production Format or “PPF,” which has found some success in ink key pre-setting and postpress operations. The PPF format was written in a proprietary format and CIP3 was considering the move to XML at the time. In the meanwhile, Adobe, Heidelberg, Agfa, and MAN Roland had put together an XML based job ticket called the Job Definition Format or “JDF” and they asked CIP3 to take over stewardship of the specification, provided that CIP3 would reorganize as a public not-for-profit entity open to all, which it did.

Adobe’s Portable Job Ticket Format (PJTF) was another early attempt at creating a method for exchanging print metadata, as was Graphic Communications Association’s Industry Architecture Project and IFRA’s ifraTrack. Each metadata program had its own unique failings and lessons learned; hence, JDF is not a “first attempt,” but rather builds on the experience gained from all of these efforts:

1. Embedding metadata into production files and/or PostScript (PJTF and ifraTrack) isn’t going to work for many reasons: the least being that front-end systems cannot handle the size and volume of production files.
2. Metadata must be typed and structured so that it can move from data store to data store (IAP)
3. The basic language needs to be XML because it is open and also widely used by programming tools (all)
4. The environment that JDF is developed in must be a public and open environment (all)

In fact, PJTF and PPF are “mapped” into JDF — In the JDF document you will find appendices that provide explicit instructions for moving from PJTF or PPF to JDF.

Once the four creators of the initial draft of the JDF specification transferred it to the new CIP4 organization in 2001, JDF 1.0 was published. Everyone agrees that it was not possible to implement JDF 1.0, rather it served as a “straw man” document that members of CIP4 could shape, change, and improve ... it was a starting point. JDF 1.1 and 1.1a were published in April and October of 2002. Significant changes were made in the specification and accompanying schema and most of

the equipment that you see on the market today is build to JDF 1.1 and 1.1a. JDF 1.2 is just being released for public review and is expected to be formally published in April of this year. This course is based on JDF 1.2 and many of the products that you will see release at drupa 2004 will include some JDF 1.2 functionality, as members have been developing there systems in conjunction with the development of the new edition of JDF 1.2.

JDF is managed in four simple tiers. Working groups, such as “Prepress” or “Digital Printing” dicuss and agree to changes that are pertinent to their area. These changes are submitted to a Technical Steering Committee (TSC) that is responsible for reviewing and approving all changes to the specification. The TSC is charged with settling issues that arise between working groups and ensuring that the overall architecture of the JDF specification and schema remain sound. Approved changes are then submitted to the Managing Editor who is charged with integrating changes to the document. Finally, a technical specialist is responsible for implementing the approved changes to the schema and ensuring that the schema is reconciled to the specification itself.

JDF is based in the Extensible Markup Language (XML), but not entirely. XML was selected as the underlying standard language for JDF because it is already a widely recognized language and is a standard. The application programming interfaces for programming languages such as Java, C+ and .NET all support XML directly and there are many XML tools, databases, and so on readily available on the market. For instance, XML has found a niche, and is widely used, for communicating content between back-office systems and web servers.

XML provides rules of syntax and default data types; however, XML alone is not enough. XML does not require that a document type definition or schema be defined, against which “instances” of XML can be checked or “validated.” Validation is critical to many types of applications, and especially where data is going to be imported into a database (as in the databases behind your workflow and MIS systems.) JDF is based on the Worldwide Web Consortium’s (W3C) XML Schema Recommendation. By using a schema, JDF allows for validation (think preflighting, but for metadata) and allows CIP4 to create user defined data types, (ex., “Rectangle” or “Matrix” as used in transforms, or “NamedColor”).

The JDF Specification also provides specifications for file naming (e.g., URL/URI usage for hot folder exchange) and methods for packing JDF and content files together in MIME Multipart Messages. Both make use of the hypertext transmission protocol (HTTP), which also assumes TCP/IP networking. The selection of a networking protocol and method of exchange are just as important to your process automation program as is the XML component of JDF.

The JDF Specification is freely available to the public, as is the JDF schema, at [www.cip4.org](http://www.cip4.org). The public schema covers all the requirements of our very flexible environment. In practice, you may want to create a subset of the schema that only uses the components that correspond to your workflow, but more on that later.

No single device (i.e., printer, press, imagesetter, etc.) is likely to implement all that the JDF specification provides for. For instance, if you are in the digital printing business, you may not care to facilitate data on hard case binding. A RIP need not facilitate JDF preflighting. A Stitcher probably doesn’t need to handle image rendering data. To specify exactly what individual classes of devices need to do with JDF, CIP4 members are developing “Interoperability Conformance Specifications” (ICS) that will provide the standard for individual classes of devices. ICS document will later (probably beginning in 2005) be used as the basis for certification testing. CIP4 has signed on GATF as the first certification testing facility and others will later be added in Europe and Asia. Once the certifica-

tion program begins, you will start seeing products that are marked as “JDF Certified” and this will be to a specific ICS document. The ICS documents are all currently in draft form and only in circulation among members of CIP4, but once published, like the standard, they will be freely available to the public and we expect that they will become part of your buying practices.

## The Three Main Functions of JDF

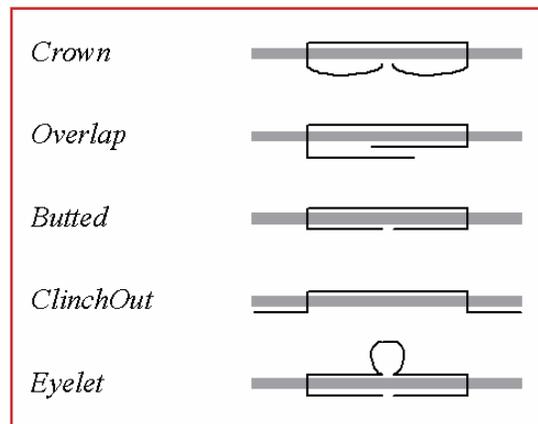
Conceptually JDF has three main functions. First, it provides a single common language that supports the lifecycle of a print job. This is what people mean when they refer to JDF as a “Job Ticket” language, but as you’ll see, it is much more than that. The second function is to provide a command and control language for devices on the shop floor. This aspect of JDF is call the Job Messaging Format or “JMF” and many folks talk about JMF almost as if it is a separate specification, but it is integral to JDF. JMF allows the controlling workflow or MIS system in a process automated environment to tell device to start and stop jobs, reorder the queue, and so on. Finally, there is inherent in JDF a flexible methodology for constructing workflows and providing the command, control and configuration of plant automation and job production. There were several improvements made to JDF 1.2, which include:

- Adding Preflighting functionality
- Improving JMF messaging
- Tightening the specifications for file naming and using MIME encoding
- Adding quality control and color management functionality
- Adding “Device Capabilities”

There may be a fourth main function, which is new to JDF 1.2 and that is automating the handshake, which is accomplished with the new device capabilities functionality. For instance, in JDF a there are five staple folds that a stitcher may use. If a new stitcher is added to your JDF workflow, the governing workflow or MIS system must know which of those five folds the new stitcher supports. Communicating the set of JDF elements and attributes supported by a device to the MIS system or workflow system is creating the “handshake.”

Currently, printers must make this reconciliation or “handshake” for themselves or with the help of their vendors and/or consultants. Some groups, such as NGP or Print City, are constructing the handshakes between devices of participating companies so that come drupa, if you buy from a the companies in one of these groups, you’ll have some assurance that the handshakes have been established and the devices among the partners have been pre-integrated.

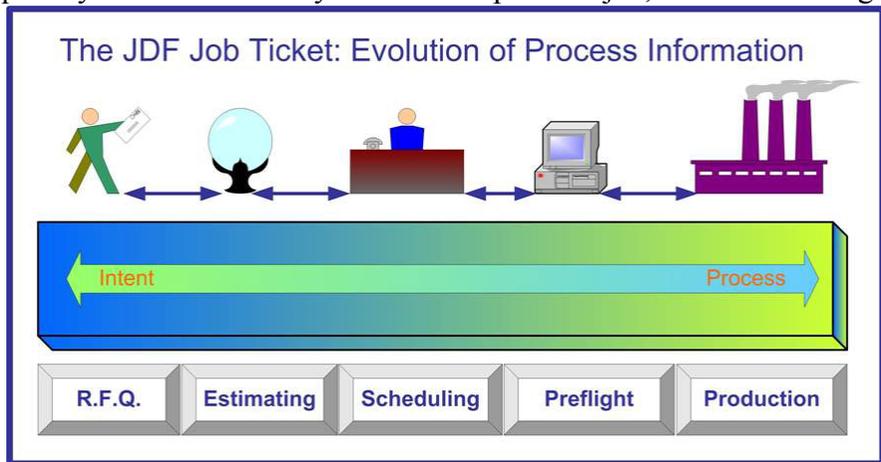
Device capabilities in JDF 1.2 allows for 1.2 capable devices to be automatically queried for the details of what aspects of JDF they can and cannot manage. This is an important step towards total “plug-n-play” interoperability, but it may be a year or more before there are enough JDF 1.2 products on the market for buyers to specify and rely on this automated handshake functionality.



## The Role of JDF as a “Job Ticket Format”

The analog version of a job ticket may be an envelope or folder with routing information that contains instructions and specifications about a job. It may start in sales with the purchase and customer information and progress through estimating, scheduling, and into production, and through the shop floor. As each employee does their part, they sign off on work completed and the job ticket moves on. JDF does provide a digital method for accomplishing the same functions. Like analog job ticket, it may begin with metadata about the job provided by the customer or gleaned from sales. This customer “intent” information may not be enough to drive the production process and it must be added to. The customer may specify “80 lbs. offset matte white,” but in estimating your staff will add a brand name and perhaps confirm availability and the actual quantity of sheets or rolls you’ll need to print the job, to include overage to account for inter-process spoilage and so on.

Likewise, in JDF there are two basic categories of job input data and materials: Intent and Process. Customer Intent is preserved but process data is added as the job progresses through its life cycle until the job can (and is) processed; hence, to use JDF does not mean that you must collect all data about a



job upfront, a JDF instance can be added to throughout the lifecycle of a job and the data does not all have to be keyed in either. Part of the idea behind JDF is to preserve job metadata at its point of entry in order to eliminate re-keying and its inherent problems of miscommunication, redundancy, and confusion. Some of the likely sources of JDF metadata include:

- Customer files and metadata — The customer’s own files may be a source of metadata that can be extracted automatically. They also may be using JDF-enabled tools that will pass along JDF data.
- JDF-enabled preflight software — Several of the most prominent preflight software companies are adding the ability to provide JDF data derived from preflighting functions.
- Customers entry via your company’s website or via an e-commerce partner — If you are using PrintTalk-based ecommerce tools or services, you may already have JDF data available to you and you don’t even know it.<sup>1</sup>
- Defaults and presets on your own equipment — Several production specifics may be established as systems defaults. For instance, that staple fold that I mentioned, you may establish one fold as the default, knowing that others are rarely used.
- Job “profiles” and defaults set in your “MIS systems” — You may have predefined job types that can be used to fill in many of the blanks, particularly if you focus on jobs with similar attributes such as printing paperback books. Once established for one job, you can focus on what is unique for each job.

<sup>1</sup> PrintTalk Schema’s make use of JDF and JDF incorporates several PrintTalk objects, including support for Request for Quote, Quote, Purchase Order, Order Confirmation, Cancellation, Refusal, Order Status Request, Order Status Response, Proof Approval Request, Proof Approval Response, and Invoice transactions. Also see [www.printtalk.org](http://www.printtalk.org).

- Direct entry — The method of last choice is to provide direct entry for operators.

Although some companies may literally use JDF as “tickets” in that they maintain a single file for a job and keep adding to it, in most cases the “job ticket” is a virtual construct maintained in a database within the software that controls your workflow. At its heart, JDF is an exchange format for use between systems. When a workflow system wishes to drive a device on the shop floor it need only prepare and deliver a JDF instance with the specifics for that particular step of the process.

So JDF is more than a job ticket, it is really a standard language for preserving job data throughout the lifecycle of a print job. In fact, two of its important functions is providing for the collection of both quality control (Q.C.) data as well as audit information about each process (i.e., start time, operator/shift, errors/waste, end time, and so on.) Since audit and Q.C. data is provided electronically and directly from devices, it can be used to troubleshoot the difference between actual and estimated costs, bottlenecks in the workflow, and automatic adjustments to inter-process overage, plus it can be used to improve management’s overall view of production to help them plan improvements, maximize resource allocation, and provide customers with better job information.

## The Role of the “MIS” System

The term “MIS” system, as used in the JDF specification, is a bit a misnomer. MIS stands for “Management Information System,” a term in the 1960s that meant a reporting system, that later in the 1970s and 1980s grew to incorporate accounting, billing, inventory, and other disparate management oversight functions. “MIS” as used in JDF is for the most part mean to cover workflow and production management systems — the business and production functions that one would expect to incorporate in a process automation project — and in our industry there are a variety of sizes and types of MIS systems ranging from enterprise-wide systems that may incorporate management reporting, accounting, and so on, to departmental workflow systems that may only control prepress operations, for instance.

At its most basic, these systems must be able to accept JDF input and decompose the JDF and store the data in its internal database, and they must be able to compose JDF from its database for output. (Beware of one-way workflow systems — in the long-term, systems that take in JDF, but cannot output JDF may prove to be roadblocks to your process automation program.) This requirement implies that these systems must be able to read and validate JDF as well as write and validate JDF.

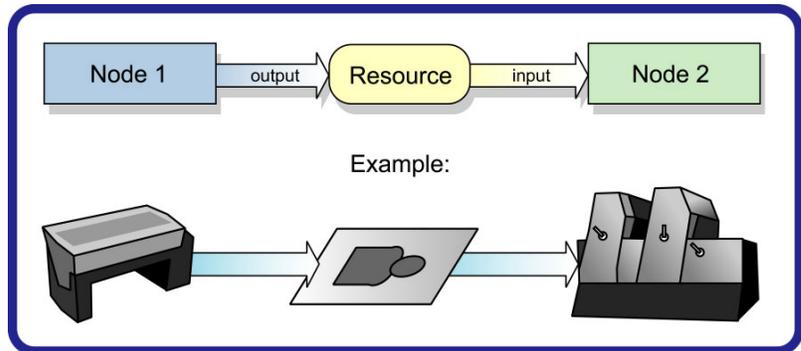
As you may recall, validation is the process of checking a JDF instance against the public JDF schema or a working subset to ensure that the rules of structure are adhered to. Some companies may integrate validation functions, while others may use a third-party tool. It’s important to know that not any off-the-shelf XML schema validation tool can be used. In the JDF specification there are certain “if-then” conditions that an off-the-shelf generic validator will not be able to validate. For instance, if a output ICC color profile is associated to an image that is incorporated into a document file that also has an output ICC color profile, and those two profile don’t match, you and I know that the document’s output color profile takes precedence and you may have to apply a transformation to the image. Conditions such as these are common to the graphic arts and in the JDF specification you’ll see many cases where a default is established at one point if a certain value exists elsewhere in the JDF instance. JDF validators available from CIP4 (to its members), Objective Advantage, and Adobe take these “if-then” conditions into account, as well as embedded systems that many vendors are building into their JDF “MIS” systems.

Your JDF MIS or workflow system must also be able to command the devices on the shop floor to act upon a job, after all, that is one of the main objectives of process automation and computer in-

tegrated manufacturing. To provide this command and control, your JDF MIS or workflow system must understand, read, and write JMF (more on that below) and must be able to store and utilize the input and parameter requirements of each device in your workflow.

The most important requirement is that your JDF MIS or workflow system must be able to organize a JDF job. JDF uses a very “Lego-like” approach to workflow. The specification identifies approximately 80 separate processes individually to which you can associate several “resources.” There approximately 160 “resources” specified in JDF and they are the input and output materials and parameters (metadata) that can be associated to a process. In JDF workflows, the output of one

process “node” is the input to the next; hence, the singular term “resource.” Once all of the required inputs for a particular job are determined to be available by a system, then it can proceed. For instance, the output of a platesetter is an image plate and that image plate is the input to a press. The press cannot begin operating until the plate is imaged. Even legacy systems with no on board computers and manual tasks can be managed this way. (You’d provide the operator with a terminal or pad which would communicate job specifics and instructions, and which would take input such as “time completed,” or “operator ID.”)



Even legacy systems with no on board computers and manual tasks can be managed this way. (You’d provide the operator with a terminal or pad which would communicate job specifics and instructions, and which would take input such as “time completed,” or “operator ID.”)

There are several features in JDF that can be used optionally to accommodate even the most complex workflows. For instance, JDF allows for the spawning and merging of jobs. This feature may be used where several different signatures print concurrently on different presses, or a cover is printed concurrently and the pieces are assembled later in bindery. You can create “combined process” that act as one with one set of inputs and outputs in a pipeline. For instance, many digital printing devices consist of a combined RIP, printer, and finishing unit and yet there is only one set of inputs and outputs. A conventional printing is actually a combined process, in that you may have ink, paper, cutting, color measurement, and other processes running concurrently. An ink system may use a special variant of pipelining that incorporates maximum and minimum values — when the ink well or tank hits a lower level, the inking system goes off-line until it is refilled, and refilling may be defined as to hit a certain maximum quantity. These are options that you may want to be aware of (and you’ll learn more about) when selecting a JDF MIS or workflow system.

Some vendors are providing options for JDF MIS and workflow systems such as middleware that can convert back and forth between JDF and legacy device controller languages. For instance, several of the major press manufacturers have been providing fairly automated presses for several years and such middleware will be important to making your transition from proprietary process automation languages to a JDF-enabled environment. Optionally, you may want your JDF MIS or workflow system to either include, or have bridges to estimating, CRM, scheduling, inventory, management and customer reporting, accounting, billing, ecommerce, and ERP systems. For instance, an integrated paper inventory system may allow you to reduce inventory levels, minimize spoilage, and get closer to a “just in time” system.

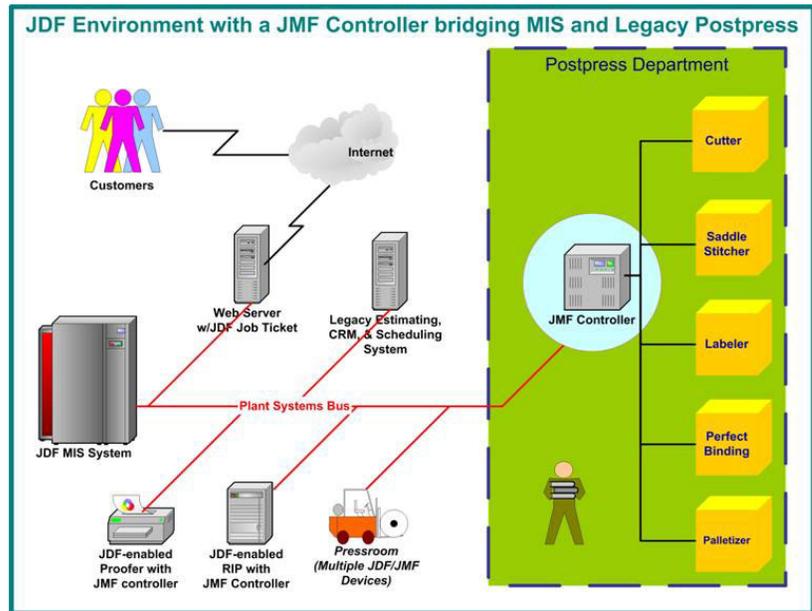
Possibly the most important option to consider is your JDF MIS or workflow system’s ability to handle extensions. Although the JDF specification provides a fairly exhaustive catalog of all the XML elements and attributes you may want to associate to each process and resource, it’s doubtful

that the specification incorporates every conceivable possibility. You may encounter vendors who have unique functions that require extensions to the JDF specification, and you may have even developed your own internal systems for dealing with customer or market specific data requirements and may find it useful to employ JDF extension of your own.

Whereas a front-end system or device on the floor only needs to pass along extension that it does not understand, a JDF MIS or workflow system must be able to understand and make use of extensions. If you add a device to your shop floor that requires a certain job parameter that is only available as a JDF extension, then your JDF MIS or workflow system can only drive that device if it can provide the required value for the extension, otherwise, it can never know when a job is ready to proceed in the workflow! Extensions are important, but should be used with caution. After you've complete this course, you will be able to determine if a vendor has a legitimate need for an extension or if they are creating synonyms for JDF elements and attributes that are already specified in order to create a proprietary hook. Your vendors should be able to provide you with documentation for extensions and CIP4 encourages both vendors and users to submit extensions for possible inclusion in the JDF specification.

## The Role of the Job Messaging Format

So far we've talked about JDF MIS or workflow systems and devices on the floor. The MIS or Workflow system acts as the JDF *agent* and it may communicate with and command a *controller*. The *controller* is an important concept in JDF. A device on the floor may have an embedded controller, the controller may be a separate physical device, or the controller may actually control more than one device. We have even seen a couple of "departmental" controllers enter the market that deals with all the devices in a department (ex., Postpress).



The language used to communicate between JDF agents and controllers is the Job Messaging Format or "JMF." JMF is part of the JDF specification. JMF also is built in XML and is part of the JDF schema. JMF allows a controller to communicate to a JDF MIS or workflow system information such as events (start, stop, error), status (available, offline, stalled, etc.), results (count, waste, etc.) and other details such as who the current operator is.

A controller may also "register" with a JDF MIS or workflow system, letting it know it is available, and where a controller controls multiple devices, it can provide registration information for the devices it supports. Note: This is information like make and model and is not a substitute for device capabilities as described above.

An MIS or workflow system may use JMF to command devices on the shop floor and may even be able to change the order of jobs in the queue. You may have noticed the heavy use of the word

“may” ... in the JMF session you’ll learn more about the options in JMF and the various level of support a controller may provide for JMF. JMF messages may also be unidirectional (the MIS system provides the commands, but the controller does not respond) or bidirectional. Determining what JMF options will be required of your process automation strategy is an important part of creating an effective JDF equipment buying policy.

JMF can also be used by one controller to provide commands to another controller. This is an important feature for supporting combined process and pipelining as describe above.

## **Strategies for implementing JDF**

The first step towards process automation and JDF implementation is to make someone responsible for developing a strategy and overseeing implementation. Odds are you are enrolled in this program because you will either be leading in, or participating in, your company’s process automation and JDF implementation program. Odds are one person isn’t going to do it! You should also be sure that you have the support of senior management and the availability of departmental heads and key employees in the planning and implementation program. I also strongly suggest that you involve select customer representatives and vendors in your team.

In the case of vendors, you will want to learn what your current vendor’s JDF implementation plans are. What products (that you already own) will include JDF-enabled upgrades? Will those upgrades be automatic or are they paid for options? What about extensions?

Document your current environment and compare your vendor’s upgrade path to equipment and systems on the shop floor. Are there any holes? Vendors who don’t have timely JDF implementation plans or products that will be un-supported in the future? Do you have custom built systems that may require JDF functionality? Or perhaps you need a middleware to communicate between some legacy devices and your JDF systems? Do any of your vendors provide such middleware?

Enterprise-wide JDF implementation is probably not achievable in the short-term and odds are you’ll need to figure out how to phase JDF in to your operations over the course of a couple of years. The key to success is capturing a return on your investment as soon as possible. Every company will have different priorities and different situations. Consider what customer niches or production lines, or departments, or plants (for the big companies) are due for capital replacement or are in the greatest need for improvement. Also consider your company’s objectives and where it is most likely to realize a return in the short-term. By focusing on a quick return from a limited implementation, you will benefit from the experience and learn to maximize your return as you approach enterprise-wide implementation.

Profectus did a study in 2002 of the potential benefits of implementing process automation and found that a \$10,000,0000 per year printer in North America could realize positive benefits to the tune of \$1.2 to \$5 million dollars over a five year period. Those savings may include:

- **Web Interface** — By implementing a web interface for both the purchasing of printing and for the communication of job information, you can capture customer data and reduce data re-keying and more cost effectively communicate with customers.
- **Improved Communications** — By implementing JDF you may be able to reduce miscommunications between customers and your staff, as well as between departments, resulting in less re-work, delays, and lost jobs.
- **Increased Customer Satisfaction** — By improving communications and providing customers with more timely and accurate job information, you may be able to improve customer retention.

- Improved Productivity — You may be able to eliminate some labor, balance loads across resources, and flush out bottlenecks in your workflow, resulting in greater productivity.
- Reduced Waste — You may be able to minimize overages, actual waste, and inter-process waste by getting better real-time counts and usage data throughout each step of the workflow and for each job.
- More Accurate Billing — With a JDF-enabled workflow you can better handle (respond to, estimate, and bill for) customer variation from job specifications and provide customers with bills that are more accurate, resulting in quicker payment and fewer disputed charges.

What your priorities are and how the above will impact your company will vary from company to company. Certainly, some companies have implemented a degree of process automation with proprietary systems, and in the short-term, less will be gained in those areas; however, in the long-term, maintenance and support for proprietary systems will become a burden compare to maintenance and support for JDF-enabled systems — one of the universal advantages of having a common language for all devices in our industry! The situation is similar to the move to CTP. If you were already accepting digital files from customers and outputting imposed film, the move was easy and the gains relatively marginal compared to companies that were still using process cameras and stripping film ... they had much farther to go and more to gain from the change.

Once you've established your priorities and objectives for your return on investment, don't forget to establish how you will measure and evaluate success. Finally, as you go through this course, consider the options in JDF, take notes, and prepare a purchasing policy and benchmarking criteria that you can apply to JDF purchases ... buy smart!

## **Where to find JDF-enabled Tools, Applications, and Resources**

There are several free and publicly available resources on that can be found at [www.cip4.org](http://www.cip4.org), including a CheckJDF application and a JDF Editor, as well the specification and schema. Anticipating that users will need more information on products and services, a "JDF Marketplace" is being published as a quarterly PDF available for your download. The JDF Marketplace includes information on products of all types, as well as information on development tools, consulting services, and educational programs and training.

You can also sign-up at [www.cip4.org](http://www.cip4.org) for a free JDF Bulletin — an informational email newsletter that includes updates on changes and developments, product announcements, case-studies, and articles written by members of CIP4.

CIP4 membership for users (Printers, publishers, buyers, consultants, and prepress services) is only \$150 per year per company and all of your staff can gain access to member-only material, including access to email forums and working groups, sample code, an archive of documents and email messages, and the JDF programmers API (for you do-it-yourself types.) The forums provide a great resource where you can get answers to technical questions.

If you are going to Graph Expo 2004, CIP4 will be hosting the "JDF Pavilion" at Graph Expo. Once attendees arrive at the JDF Pavilion, they can sign up for JDF Pavilion Tours where they will see demonstrated JDF-enabled interoperability between groups of four or five JDF solutions providers.

There will be 18 participating companies in the JDF Pavilion demonstration area and these companies include both large and small innovators offering JDF-enabled products for virtually all aspects of the print workflow. Furthermore, there will be a theater in the JDF Pavillion that will feature educa-

tional presentations throughout the duration of Graph Expo. This will be an excellent opportunity for you see JDF applications in action, ask questions, and learn.

Don't forget to ask your vendors (including your distributors) about technical support and training they may be able to provide. Several vendors are providing seminars around the globe on JDF topics. There is a 13 session JDF Expert Certificate program offered by CIP4 through the International Prepress Association that is available via the Internet on demand (See <http://www.ipa.org/jdf/> for more details.) Also, the Graphic Arts Technical Foundation ([www.cip4.org](http://www.cip4.org)) and the Fraunhofer Institute for Computer Graphics (contact Stefan Daun at [Stefan.Daun@igd.fhg.de](mailto:Stefan.Daun@igd.fhg.de)) have various JDF programs or studies and may be able to provide additional support.

## About the Author

James E. Harvey is the founder and president of Media4theWorld, LLC, in Crofton, Maryland, which provides management and technical consulting to the media industries. He was responsible for the edit and production of the Job Definition Format specification versions 1.1, 1.1a and 1.2, and also serves as CIP4 Organization's Executive Director. For eight years was VP of Spectrum Operations at Graphic Communications Association (now IDEAlliance), a technical and management association for the print and Internet publishing industries. For eight years Harvey was also a Business Development Manager at Volt Information Sciences where he organized and pioneered document and data warehousing, SGML services and applications, CD-ROM publishing, hypermedia, and object-oriented document constructs. He has held numerous industry posts including VP & Board member of TAGA, Secretary and member of the Executive Committee of the Digital Ad Lab, Secretary of ANSI/PIMI IT2 (Densitometry), has authored or co-authored several key industry documents, and has spoken at over 100 industry events. Jim earned his M.S. in Business from John Hopkins University. (See <http://www.media4theworld.com>)

