

Software Version 9.0 SP1  
December 2010  
701P31780



# FreeFlow<sup>®</sup> Process Manager

## Manifest Automation from Xerox



©2008-2010 Xerox Corporation. All rights reserved. Xerox and the sphere of connectivity design, are trademarks of Xerox Corporation in the United States and/or other countries.

Changes are periodically made to this document. Changes, technical inaccuracies, and typographic errors will be corrected in subsequent editions.

Document Version: 2.1 (December 2010)

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
	Summary of Chapter.....	1
	Intent.....	1
	Audience and Skill set .....	1
	Terminology.....	1
<b>2</b>	<b>MAX Overview.....</b>	<b>3</b>
	Summary of Chapter.....	3
	What is FreeFlow Process Manager MAX.....	3
	Applications Supported.....	3
	High Level view of MAX.....	4
<b>3</b>	<b>Manifest Processing .....</b>	<b>7</b>
	Summary of Chapter.....	7
	Manifest Files.....	7
	Parsing Manifest Files .....	8
	Mapping Manifest Data To Manifest Fields.....	11
	Bill of Materials Configuration .....	18
	Manifest Submission Status File.....	19
	Using MAX variables in Process Manager workflows.....	21
	Print Ticket Processing .....	24
	JMF Status Signals.....	25
	Best practices .....	26
<b>4</b>	<b>Walk Through Example.....</b>	<b>29</b>
	Summary of Chapter.....	29
	Explore the ESv9 - Personalized Cards example.....	29
	Demonstration of the Example.....	32
	Example discussion .....	35



# Introduction

# 1

Welcome to FreeFlow Process Manager Manifest Automation from Xerox (MAX).

## Summary of Chapter

The purpose of this chapter is to provide you a high level overview of:

- Intent
- Audience and Skill set
- Terminology

### Intent

This document was designed to explain how to submit documents to FreeFlow Process Manager through an interface known as Manifest Automation from Xerox. This document is intended for integration professionals who wish to understand the MAX interface included with FreeFlow Process Manager.

### Audience and Skill set

This guide has been created for Xerox Analysts, software developers, technical developers, and integration professionals.

### Terminology

The major elements of FreeFlow Process Manager MAX are:

- Manifest Automation from Xerox (MAX)  
The process of enabling automated job submission by processing manifest files.
- Manifest file  
The manifest file is a delimited file that contains informational and control data required to submit jobs to Process Manager.
- Manifest record  
A single row in a manifest file.
- Manifest column  
The values in the same position across all manifest records in the manifest file.

- **Manifest field**  
A data field that is used by Manifest processing to pass job information to FreeFlow Process Manager. Manifest columns are mapped into manifest fields based on the MAX configuration.
- **FreeFlow variable**  
A FreeFlow variable is a representation of job metadata within a process node setting. FreeFlow variables can come from both MAX variables as well as JDF variables.
- **Status file**  
A manifest file with two additional columns used to provide manifest submission status.
- **Bill of Materials**  
The Bill of Materials, or BOM for short, is a summary of the jobs submitted as part of a single order in a manifest. The summary is based on the manifest file data and configured workflow parameters.

# MAX Overview

# 2

## Summary of Chapter

The purpose of this chapter is to provide you with an overview of:

- What is FreeFlow Process Manager MAX
- Applications Supported
- High Level view of MAX

### What is FreeFlow Process Manager MAX

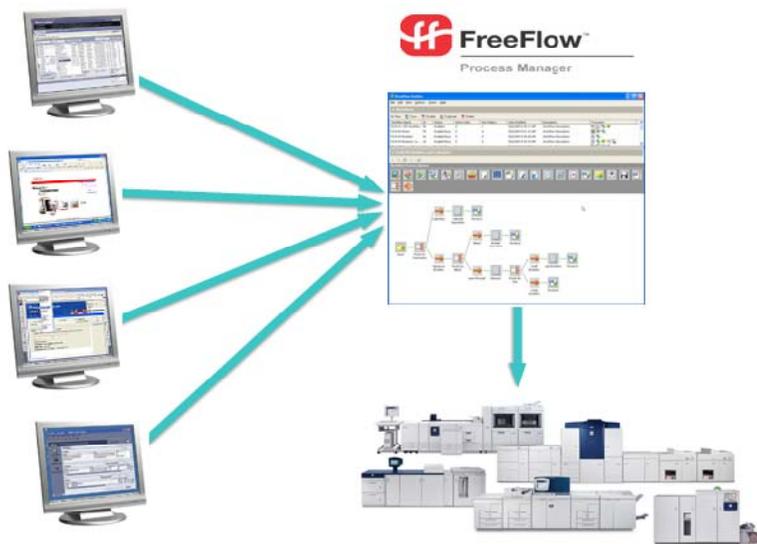
FreeFlow Process Manager automates prepress activities and increases the operation's overall efficiency. MAX (Manifest Automation from Xerox) builds on Process Manager's capabilities, streamlining the order-fulfillment process by connecting order-management systems to automated print production.

The result is a smooth, automated process from work-order submission through print submission. MAX reduces and in many cases eliminates "touch points" between the order-entry process and production, and helps maximize productivity and profit.

### Applications Supported

Process Manager Automation Software, in general, supports all the major Monochrome and Color publishing applications. The supported applications include manuals, directories and technical documentation, flyers, brochures, business cards, books and course materials, proposals and reports, books, newsletters, photo, and postcards, etc.

## High Level view of MAX



Process Manager MAX connects the print provider's order-management system, print MIS, FreeFlow Web Services, or other Web to Print systems with automated print production. It enables a print provider to process orders automatically by submitting a manifest file containing order records to Process Manager.

Each record in the manifest file corresponds to a job or part of a job from the order-management system. Each record might include information such as the order number, copy count, media, filename, barcode information, and production method (an identifier mapped to a Process Manager automated workflow). Records may also contain information for job personalization or for the Bill of Materials.

MAX is capable of accepting input from multiple sources at one time. Manifest files from each source can be processed using a different hot folders and MAX configurations. MAX also supports automated pre-scripting. This enables automatic conversion of input files into compliant manifest files in order to enable integration with systems that may not be able to produce a compliant manifest file.

### MAX Configuration Overview

- Define workflows in Workflow Builder
- Set-up Hot Folder for Manifest processing in Workflow Builder
  - Create a Hot Folder and select the “Enable this folder for use with manifest automation” option

- Use Manifest Automation from Xerox configuration tool to:
  - Define Hot Folder manifest processing options
    - Specify the manifest column that contains the Production Method
  - Map Production Method values to a desired workflow and to a MAX configuration
  - Define a MAX configuration to map the manifest columns to Manifest fields
- Optionally, add mapped manifest fields to process node settings in workflow

### **MAX Processing Overview**

- Manifest file gets dropped into the designated Hot Folder. For each record in the manifest the following happens:
  - The value found in the Production Method is used to select a MAX configuration
  - The MAX configuration is used to map the manifest columns to the Manifest Fields
  - The value found in the Production Method field is used to select a FreeFlow Process Manager workflow
  - The values in the Manifest fields are applied to the workflow process node settings that use MAX variables
  - The job file(s) are retrieved
  - The job files are submitted to the workflow
- A manifest status file is generated



# Manifest Processing

## Summary of Chapter

This chapter provides an overview of the following:

- Manifest Files
- Parsing Manifest Files
- Mapping Manifest Data To Manifest Fields
- Bill of Materials Configuration
- Manifest Submission Status File
- Using MAX variables in Process Manager workflows
- Print Ticket Processing
- Best practices

### Manifest Files

A manifest file is a text file where record fields are separated by a configurable delimiter. Each record may represent a single job or be part of a multi-part job.

A sample manifest file with five columns of data and a comma [,] delimiter is organized as follows:

```
Rec1Col1,Rec1Col2,Rec1Col3,Rec1Col4,Rec1Col5
Rec2Col1,Rec2Col2,Rec2Col3,Rec2Col4,Rec2Col5
Rec3Col1,Rec3Col2,Rec3Col3,Rec3Col4,Rec3Col5
Rec4Col1,Rec4Col2,Rec4Col3,Rec4Col4,Rec4Col5
```

In this example, RecNo. specifies the record number and ColNo. specifies the field number.

A manifest file with actual data could look like this (space added for legibility):

```
Convert-print,      C:\\cust1001\\covers.doc,      1001,  MAKE,  500
Watermark-print,   C:\\cust1001\\intro.pdf,       1001,  MAKE,  500
Review-print,      C:\\cust1001\\chap1-5.pdf,     1001,  MAKE,  500
Review-Save,      C:\\cust1001\\memo.pdf,       1001,  PICK,   1
```

## Parsing Manifest Files

Manifest files are submitted to and parsed by FreeFlow Process Manager Manifest hot folders.

### Adding a Manifest hot folder

Use Workflow Builder to create a hot folder. Then check the **Enable this folder for use with manifest automation** checkbox to associate the hot folder with the **Submitted with Documents** target workflow.

By default, a Manifest hot folder will parse manifests that:

- have the Production Method in column 1

Note: The Production Method is the value that's used to automatically select a process manager workflow for a job.

- are comma separated
- use no text qualifier

### Header records

When the # symbol is the first character in a record, it indicates to FreeFlow Process Manager that the record is a header record. These records are ignored when parsing manifest files.

### Configuring a Manifest hot folder

Manifest hot folders can be configured to support delimited files other than the default CSVs. This allows manifest file parsing to accommodate constraints from upstream systems.

To change the default manifest hot folder options, select **Hot Folders Options** from the **Manifest Automation from Xerox** configuration tool.

### Job Submission

The options in the **Job Submission** tab control how jobs in the manifest are submitted to Process Manager:

The value in the **Production Method** is used to identify the job type that's represented in each manifest record. Different job types are usually submitted to different workflows. The Production Method value is built from up to five columns in the manifest.

The Production Method is also used to select a Manifest configuration that will be used to map the parsed manifest columns to manifest fields.

#### Note

Manifests will not process if the manifest configuration for a given Production Method specifies a column that's not in the manifest.

The **Submission Options** are used to specify whether to create a composite job from contiguous manifest records with the same order ID, Group key, and Production Method values. This hot folder configuration parameter can be set to either **Always submit groups of files as a single job** or **Submit groups of files as a single job only if the workflow has a join process node**.

### Processing Options

The options in the **Processing Options** control how manifest records are processed.

Two parameters define how values are found in each manifest record:

The **Column delimiter at order jacket file** defines the delimiter that separates the values in the manifest file.

The **Text Qualifier** is used to surround the values in the manifest file. It may be set as none, single or double quotes. If a text qualifier is used, it must be used to enclose all values in the manifest records. Use of a text qualifier allows the values in the manifest record to contain the specified delimiter.

For example if the delimiter is a comma and one of the fields in a record contains the text **Xerox, Process Manager, Xerox** and **Process Manager** will be separated into two field values, thus causing an error during parsing. If you use " as the text qualifier, then "Xerox, Process Manager" will be processed as a single field whose value is **Xerox, Process Manager**.

Note: Leading and trailing spaces in the manifest values are ignored by default. However, if a text qualifier is used then leading and trailing spaces are included in the manifest values.

The **Process header row of the order jacket file** option determines whether the first row should be processed. This can be used to ignore a header row that does not start with the # symbol.

#### Note

Any row, including the first row, that begins with a # symbol is ignored when processing the manifest.

Two parameters define error-handling policies:

The **When an error occurs while processing an order jacket file** specifies whether to stop processing a manifest if errors occur.

The **When an error occurs while processing an order** specifies whether to stop processing an order within a manifest if errors occur.

After a manifest is processed, the manifest is saved with two additional status columns appended to each manifest record. This manifest **status file** can be

saved to either a subfolder of the manifest hot folder or to a specified location. For more information see the Manifest Submission Status File section.

### Conversion Script

You can have a script run on any file that is received by the hot folder and use output of the script for MAX processing. The purpose of the script is to transform, normalize, or convert submissions into a manifest that can be processed by MAX. You can define the following manifest attributes:

**Script full path** - This is the script to run. It must include the full path. You can use the File Browser to find the script.

**Parameters** - Specifies the format of the parameters passed to the script. By default the script parameter list must include the following:

- \$in\$ - input file location (the file received by the hot folder)
- \$out\$ - output file location (the full path of the output file to be processed by MAX)
- \$error\$ - error file location

You can specify additional switches needed by the script and re-order the switch and parameters. For example, if your script takes four parameters in the following order:

```
-v -f input_file error_location -o output_file
```

Then the parameter field should be: `-v -f $in$ $error$ -o $out$`

**Error file directory** - This directory is passed to the script as the \$error\$ parameter. This directory is not monitored by MAX and is provided as a convenience.

**Timeout** - The timeout period specifies how long MAX should wait for the script output prior to timeout.

When a file is dropped in the hot folder, MAX will make a CLI call using a string that concatenates the script full path and the parameters. If the CLI call requires quotes around the path or around specific parameters, they should be added in the conversion script configuration.

The manifest hot folder accepts any file type when the conversion script is enabled. The result of running the script must be saving a manifest to the location specified in the \$out\$ variable.

### Manifest Data

The manifest hot folder configuration defines how manifests should be parsed in order to extract values from the delimited file. The output of the manifest hot folder parsing is normalized manifest data. This internal data is then mapped to manifest fields using the selected manifest configuration.

The mapping between the manifest hot folder and the manifest configurations can be a combination of any of the following:

- A single hot folder processes manifests that use a single manifest configuration
- A single hot folder processes manifests that use multiple manifest configurations
- Multiple hot folders process manifests that use a single manifest configuration

- Multiple hot folders process manifests that use multiple manifest configurations
- The following comma separated manifest example contains records within a single manifest that can be processed by multiple configurations:

```
# ProdMethod,      Filename,          Order,  Event,  Quantity
Convert-print,    C:\\cust1001\\covers.doc,    1001,  MAKE,  500
Watermark-print,  C:\\cust1001\\chap1-5.pdf,    1001,  MAKE,  500
Watermark-Save,  C:\\cust1001\\memo.pdf,      1001,  PICK,  1
#ProdMethod, Order, Quantity, Filename, cardName, cardImage
watermark-print,1002, 50, C:\\cust1002\\NameCardTemplate.pdf, John Doe, JD.jpg
```

## Mapping Manifest Data To Manifest Fields

Once the manifest data has been normalized to a coherent internal representation the data is then mapped to specific manifest fields.

### Manifest Processing Constraints

- Up to 30 MAX configurations can be concurrently active
- The first 100 columns in a manifest record can be mapped in a manifest configuration

### Job Submission Manifest Fields

#### Job Name

MAX can leverage manifest fields to customize Process Manager job naming. This can be useful to, for example:

- Use external job IDs in Process Manager and DFEs job names  
This enables mapping of job names in accounting data to job IDs in MIS
- Use of external job IDs when saving jobs for archival
- Use of manifest information for conditional routing within a workflow

Choose one of the following methods to configure the job name:

- Use the file name field as the job name.
- Build the job name using selected fields from the manifest. Combine the values of up to five manifest columns to create the job name.
- Create MAX job name using variable data syntax. This allows the job name to be defined using static text and values from the manifest columns.

Note: The manifest job naming will automatically remove the following invalid characters from the job name: ASCII Control Characters (\0 - \31), “, <, >, |, backspace (\b) and tab (\t)

## Production Methods

The **Production Method** represents a job type identifier for a specific upstream system such as an MIS or a Web to Print system. The **Production Method** field is defined in the manifest hot folder options. The values in this field are used to select a manifest configuration and a workflow for each manifest record.

**Production Methods and Manifest Configurations** – Manifest configurations typically map columns in manifests from a specific upstream system to Process Manager Manifest Fields. Different upstream systems typically have different job type identifiers that are used as the Production Method values. Hence Production Method values are mapped to workflows within a manifest configuration. This manifest configuration then determines how the rest of the columns in the specific manifest record are mapped to the Manifest Fields.

Each manifest configuration has its own set of unique Production Method values that must be unique across all manifest configurations. The Manage Mappings dialog in the Production Methods displays every production method across all manifest configurations. You can move production method mappings between manifest configurations within this dialog.

**Production Methods and Workflows** – Workflows are typically built to automate prepress for specific job types, Hence Production Method values are mapped to specific workflows. However, the job types as defined by the Process Manager workflows may be broader or narrower in scope than the job types as defined by upstream systems. To deal with this disparity:

- Multiple Production Method values may be mapped to the same workflow. This allows a single workflow to process a variety of job types from one or more upstream systems.  
For example, a single “saddle stitch workflow” may be mapped to booklets, calendars and folded brochures.
- Multiple manifest fields may be used for the Production Method. This allows different variants of the same job type to be submitted to different workflows.  
For example, paperback books for a specific account may be routed to a workflow that processes those books based on unique preferences from that account.

Individual manifest hot folders can use different combinations of manifest columns to define Production Method values. As such, Process Manager can concurrently integrate with multiple systems that may have very different job type definitions.

## Additional Manifest Fields

### Mandatory Fields

In addition to the Production Method, MAX requires data for two additional Manifest Fields in order to enable automated job submission. FreeFlow Process Manager nodes that support MAX variables can also reference these fields.

- **Order ID**

Line items that share a common **Order ID** value are considered part of the same order. All line items that share a common **Order ID** value must be contiguous in the manifest.

The **Order ID** field controls MAX processing in the following ways:

- Composite jobs  
The **Order ID**, when used in conjunction with the **Group Key** and the **Production Method**, determines which items are submitted together as a composite job.
- Bill of Materials  
A Bill of Materials is generated for each unique **Order ID**.

- **File name**

The **File name** manifest field and the optional **Folder name** manifest field specify the name and location (URI) of the source files to be submitted for processing.

#### File URI Constraints

For FTP, HTTP, HTTPS and Repository URIs, the entire URIs must be in the **File name** manifest field. These URIs must also be URL encoded (e.g. spaces are represented as % 20).

Paths to files in the manifest file must be one of the following URI types:

- Local path – e.g. c:\My Files\MyFile.pdf
  - Mapped drives cannot be used.
- UNC – e.g. \\ServerName\ShareName\My Files\MyFile.pdf
- FTP – e.g. ftp://ServerName/My % 20Files/MyFile.pdf
- HTTP – e.g. http://ServerName/My % 20Files/MyFile.pdf
- HTTPS – e.g. https:// ServerName/My % 20Files/MyFile.pdf
- Repository URLs – e.g. rms://ConnectorName/DocumentID

#### File Access Credentials

UNC – the user XDL\_ADMIN must have access rights to the file resources to be retrieved.

FTP – the credentials must be in the URI

ftp://user:password@ServerName/My % 20Files/MyFiles.pdf

Note: A dedicated account with limited permissions should be used for FTP logins since the credentials are visible in the manifest.

HTTP/HTTPS – by default the XDL\_ADMIN account credentials will be used if authentication is required. Alternatively the credentials can be added to the URI

http://user:password@ServerName/My % 20Files/MyFile.pdf

### Notes

Only HTTP/HTTPS basic authentication is supported.

A dedicated account with limited permissions should be used for HTTP/HTTPS logins since the credentials are visible in the manifest.

### Repository

The repository credentials are defined in the **Repository Credentials Setup** in Workflow Builder. These credentials are used for all `rms://` URIs.

If retrieving files from the repository defined as the “Active” connector in the FreeFlow Administration Tool the ConnectorName is not needed in the URI. Otherwise, the ConnectorName in the URI should be an URL encoded version of the Connector Name in the FreeFlow Administration Tool.

DocuShare example – `rms://ConnectorName/File-#####`

SharePoint example –

`rms://ConnectorName/http%3A%2F%2FServerName%2FMy%20Files%2FMyFile.pdf`

### Note

SharePoint files may also be retrieved using standard HTTP urls – e.g.  
`http://ServerName/My%20Files/MyFile.pdf`

### Network configuration

File retrieval is performed using the XDL\_ADMIN account. As such, any required network settings must be configured in this account.

For example, HTTP, HTTPS and FTP proxies, if needed, must be added to the XDL\_ADMIN account in order to enable job retrieval using these protocols.

### JDF Submission

Unlike XPIF tickets, JDF tickets must be submitted using the **File name** manifest field. During job submission Process Manager will retrieve the files specified in the JDF RunList. If a JDF RunList specifies multiple files, these are submitted as a composite job.

Paths to files within the JDF file support the same URI types as paths in the Filename manifest field.

### Note

Manifest records that specify a JDF file in the **File name** manifest field cannot be grouped with other manifest records.

The location or URL can be specified as follows:

File Name specified?	Folder Name specified?	Result	Example
Yes	No	The File Name field must include the file location.	File Name field = c:\MyFiles\MyFile.pdf
Yes	Yes	The folder name and file name are combined to specify the file location.	File Name Field = MyFile.pdf Folder Name Field = c:\MyFiles\
No <b>Note</b> The File Name field can be blank but it must be mapped because it is a mandatory field.	Yes	All supported file types in the folder are submitted.  <b>Note</b> When files are submitted to a workflow using only the <b>Folder name</b> , there's no guaranteed join order for the files in the folder.	File Name Field = Folder Name Field = c:\MyFiles\ or File Name Field = c:\MyFiles\MyFile.pdf Folder Name Field =

### Optional Control Fields

MAX supports a number of optional fields that can be used to enhance job submission. FreeFlow Process Manager nodes that support MAX variables can also reference these fields.

- **Event type**  
The **Event type** field specifies whether the item is submitted to Process Manager (make job) or whether the item is a previously printed or non-print job (pick job). Pick jobs are not submitted to the Process Manager workflow, and do not require file information, but PICK items appear in the Bill of Materials.  
Constraints
  - Values must be either MAKE or PICK. MAKE is used if the field is blank.
  - Even though Pick jobs are not submitted to Process Manager, a production method is still required so that MAX can determine which configuration to use when mapping the Pick line item.
- **Folder name**
  - See File name for information on how to use the **File name** and **Folder name** fields to specify file locations.
  - Folder names are only valid for Local and UNC URIs.
- **Quantity**  
The **Quantity** field specifies the copy count.

- **Group key**

The **Group key** field directs MAX to submit multiple line items as a composite job to Process Manager. All contiguous items with the same **Order ID**, **Group Key**, and **Production Method** may be submitted as a composite job. If the composite job is joined, the files are joined in the order listed in the manifest. Whether the files are submitted as a composite job is determined by the **Job Submission** settings in the **Hot Folder Options**.

- **Xpif file**

The **Xpif file** field specifies location of the xpif job ticket to be associated with the job.

When using FTP, HTTP, HTTPS or Repository URLs to retrieve an XPIF file, the full path to the file must be in the **Xpif file** manifest field.

The location can be specified as follows:

XPIF File Name specified?	Folder Name specified?	Result	Example
Yes	No	The XPIF File Name field must include the file location.	File Name field = c:\MyFiles\MyFile.xpf
Yes	Yes	The folder name and XPIF file name are combined to specify the file location. <b>Note</b> The folder name that's combined with the file name is also combined with the XPIF file name.	File Name Field = MyFile.xpf Folder Name Field = c:\MyFiles\
No	Yes	If the folder contains a job file with an XPIF that has a matching name then the XPIF is automatically associated with the job file. <b>Note</b> Example of matched PDF and XPIF filenames: c:\MyFiles\MyFile.pdf c:\MyFiles\MyFile.pdf.xpf	File Name Field = Folder Name Field = c:\MyFiles\ or File Name Field = c:\MyFiles\MyFile.pdf Folder Name Field =

- **Priority**

The **Priority** field specifies the priority of the jobs in the manifest file. High priority items are submitted to FreeFlow Process Manager before Normal priority items. This manifest field controls which line-items in the manifest get submitted to the workflows first, but it does not guarantee which jobs complete first.

- **Constraints**

Values must be either High or Normal. Normal is used if the field is blank.

- **Paper stock name**

The value from the **Paper stock name** is used as the **Main Stock** for the job.

In order to use the **Paper stock name**, FreeFlow Print Server's Stock Library must have a stock with the same name as the value in the **Paper stock name** manifest field.

While processing the manifest, MAX will attempt to retrieve and use all paper stock properties from the printer information within Process Manager. If the specified **Paper stock name** is not found then only the stock name is changed in the print ticket's Main Stock selection. All other stock parameters will remain at their defaults.

During print submission, FreeFlow Process Manager will submit the stock name and the stock attributes. If FreeFlow Print Server receives a stock name in a print ticket it will ignore other attributes when matching the specified media to items in its stock library.

**Note**

The DFE Stock Library is retrieved and stored by Process Manager whenever the print settings for a given printer are changed.

### Optional Data Fields

MAX supports a number of data fields that become part of the Bill of Materials. FreeFlow Process Manager nodes that support MAX variables can also reference these fields.

- **Title** - the title of an item
- **Part ID** - the part number of an item
- **PO number** - the purchase order of an item
- **Rush** - the priority of an item to be used by the operator

**Note**

MAX does not use the Rush field to determine processing priority.

- **Due date** - the due date of an order

**Note**

Use MM/DD/YYYY format.

- **Barcode value** - the number to be printed as a barcode on the Bill of Materials

**Notes**

A large font size or too many characters can cause the barcode to overflow and print off the page. Also, this barcode is not related to the barcode process node.

A valid barcode font must be installed on the Process Manager system.

MAX processing does not generate start, stop, or checksum for the barcode in the Bill of Materials.

### Optional Custom Fields

You can configure up to 50 custom fields. Custom fields appear on the Bill of Materials and are passed through to the status file. FreeFlow Process Manager nodes that support MAX variables can also reference these fields.

#### Note

Custom fields should be named using alphanumeric characters, spaces and underscores.

### Unconfigured Fields

Not all of the fields in the manifest file need to be defined as Manifest fields in the MAX configuration. FreeFlow Process Manager nodes that support MAX variables can reference undefined fields by their manifest column number.

#### Note

If referencing manifest fields using column positions, it can be helpful to open a sample CSV file in a spreadsheet and add a header row with column numbers. For example:

#1,2,3,4,5,6,...

## Bill of Materials Configuration

The Bill of Materials is a summary of the orders submitted through MAX to Process Manager and includes much of the data in the manifest. A separate bill of materials is generated for each order in the manifest.

### Bill of Materials output

**PDF Bill of Materials** - By default a PDF Bill of Materials (BOM) is generated for each order in the manifest file. When generating a PDF BOM you can:

- Specify the workflow that will process the BOM.
- Specify the font that should be used to create the BOM.

**XML or Flat Files** – The Bill of Materials (BOM) can also be saved to a predefined location as either XML data or as a flat file.

### Bill of Materials Name

There are two ways to configure the Bill of Materials name:

- Use the **Order ID** and timestamp as the Bill of Materials name.
- Combine up to five manifest columns to create the Bill of Materials name.

### Bill of Materials data

**Standard Fields** – Specify which manifest fields' values should be included in the XML or flat file Bill of Materials. This also allows customization of the default manifest field names in exported BOMs.

**Note**

Bill of Materials exported as PDFs include all active manifest fields.

**Custom Fields** - Custom fields appear on the Bill of Materials along with standard fields. To add a custom field, provide a custom field name, the column position in the manifest.

## Manifest Submission Status File

A status file is created after a manifest is processed. The status file contains the original manifest content and two additional status columns. The first column indicates success or failure. The second column contains a status code.

The name of the status file is composed of:

- the original filename
- *.status* if there were no submission errors or *.error.status* if there were errors
- the original extension
- Example

The status filename for the file 'MyOrderJacketFile.csv' would be:

- 'MyOrderJacketFile.status.csv' if no error occurs
- Or 'MyOrderJacketFile.error.status.csv' if an error occurred.

A status file indicates the success or failure of the job submission. The status file does not indicate the status of the jobs in the workflow.

### Status Codes

Code	Description
B1	PDF Bill of Materials list was not generated
B2	XML Bill of Materials was not generated
B3	Flat file (.txt) Bill of Materials was not generated
C1	The manifest configuration was not loaded
D1	Directory access error – the folder does not exist
E1	Unknown exception
F1	File access error – file is missing
F2	File access error – file exists, but it cannot be retrieved (typically a permissions error)
F3	A file with an invalid file type was found
F4	The file is an RDO and it is not ready for submission (typically one or more files in the CON directory do not have read/write access or they are missing)

F5	File access error – cannot log into server (typically incorrect credentials)
G1	Generic error, malformed manifest record (typically the manifest contains fewer columns than are expected by the manifest configuration)
G2	The Order ID field is empty
G3	The Job name field is empty after illegal characters have been removed
G4	The same Order IDs is used in non-contiguous line items
G5	Manifest data could not be stored as job metadata (typically a custom manifest field name contains illegal characters)
H1	Manifest variable data processing encountered an error
HTTP1	Repository access failed when retrieving files (typically invalid URIs, permission errors or invalid repository credentials)
I1	The Event value was invalid – the value must be 'MAKE' or 'PICK'
J1	Composite job contains a JDF file
J2	When one or more items in a composite job encounter an error, all other items in the composite job will fail with a J2 error
JDF1	Invalid JDF file
O1	If an error occurs and manifest processing is set to stop processing an order when an order items error, then all order items except the problematic item will fail with an O1 error
P1	The Priority type was invalid – the value must be 'High' or 'Normal'
PM1	The Production method value was not found in any manifest configuration
Q1	The Quantity value was invalid – it must be blank or must be a positive integer
S1	Success
SM2	Submission of manifest jobs to Process Manager failed
W1	The workflow is invalid – the workflow mapped to the production method is disabled or does not exist
X1	Invalid XPIF file
X2	Failed to apply XPIF to job
X3	Failed to apply XPIF when using folder submission

## Using MAX variables in Process Manager workflows

MAX variables enable use of values from manifest fields in process node parameters. When processing a record from a manifest, FreeFlow Process Manager will replace the MAX variables in process node settings with the values from the manifest record.

The MAX variable syntax requires that variables begin with the letters '\$CSV' or '\$FF' (called the prefix) and end with the '\$' character (called the suffix). The MAX variable is specified using either the manifest field name or the manifest column number between the prefix and suffix. Additional syntax allows extraction of only a portion of the manifest field value.

- Using a manifest field name - Specify a MAX variable by placing the field name between the MAX variable prefix and suffix.

Examples:

- \$CSVFile name\$ - Requests the File name manifest field
- \$FFQuantity\$ - Requests the Quantity manifest field

### Note

Field names are case sensitive. They must match the name displayed in the Manifest Automation from Xerox configuration tool.

- By column position - Specify a MAX variable using the column position by placing the manifest column number between the prefix and suffix.

Example:

- \$CSV2\$ - Requests column 2 of the manifest

- Extracting only portions of field value - you can choose to extract a specified number of characters from the left, right, or from the middle of the string. The following keywords are supported:

- LF(n) - Extracts the first *n* characters from the string
- RT(n) - Extracts the last *n* characters from the string
- MID(s, n) - Extracts *n* characters from the string, starting with the character at position *s*

The manifest field name, followed by the '.' symbol must precede the keyword.

Examples:

- \$FFFile name.LF(3)\$ - Requests the first three characters of the File name manifest column. If the File name column contained the text, 'XeroxFreeFlow.tif', the characters extracted from text would be 'Xer'.
- \$FFFile name.RT(8)\$ - Requests the last eight characters of the File name manifest column. For example, with 'XeroxFreeFlow.tif' in the File name column, the characters extracted from text would be 'Flow.tif'.
- \$FFFile name.MID(6, 4)\$ - Requests four characters of the File name manifest column, starting with the sixth character. For example, with 'XeroxFreeFlow.tif' in the File name column, the characters extracted from text would be 'Free'.

## Manifest variable precedence

There are three sources of runtime variables in FreeFlow Process Manager: MAX, PDF and JDF.

The following variables are automatically populated from PDF metadata whenever a PDF is submitted:

- \$FFTitle\$
- \$FFAuthor\$
- \$FFSubject\$
- \$FFKeywords\$
- \$FFApplication\$
- \$FFProducer\$
- \$FFPDF Version\$
- \$FFNumber of Pages\$

### Note

These are the same variables that are populated when a PDF file is submitted via Hot Folders, JMF or the Workflow Submission Client.

The following variables are automatically populated whenever a JDF job ticket is submitted:

- \$FFFfile name\$
- \$FFFfile path\$
- \$FFQuantity\$
- \$FFJob Name\$
- \$FFOrder ID\$
- \$FFCustomer ID\$
- \$FFBilling Code\$

When using MAX to submit JDF jobs to Process Manager, the manifest field variables have higher precedence than the JDF variables. As such, the following variables will be populated with manifest values if these manifest fields are mapped in the configuration. Otherwise these values are populated using JDF information:

- \$FFFfile name\$
- \$FFQuantity\$
- \$FFJob Name\$
- \$FFOrder ID\$

Note: The \$FFFfolder name\$ and \$FFFfile name\$ variables contain the URI to the JDF file. The \$FFFfile path\$ contains the URI to the job file within the JDF ticket.

When submitting JDF jobs to Process Manager, the JDF variables have higher precedence than the PDF variables. As such, the manifest field variables also have a higher precedence than the PDF variables.

## Supported process nodes

The following process nodes support use of MAX Variable Syntax:

Node	Locations
Watermark	<ul style="list-style-type: none"> <li>in the Text Watermark's content</li> <li>in the Image Watermark's filename</li> <li>as the Page Range</li> </ul>
Bates Numbering	<ul style="list-style-type: none"> <li>in the prefix</li> <li>as the Page Range</li> </ul>
Page Numbering	<ul style="list-style-type: none"> <li>in the Prefix</li> <li>in the Suffix</li> <li>as the Page Range</li> </ul>
Barcode	<ul style="list-style-type: none"> <li>in the text portions of the data string</li> <li>as the Page Range</li> </ul>
Rotate	<ul style="list-style-type: none"> <li>as the Page Range</li> </ul>
Flip	<ul style="list-style-type: none"> <li>as the Page Range</li> </ul>
Insert Pages	<ul style="list-style-type: none"> <li>in the Insert Filename when inserting pages from a file</li> <li>as the Page Range</li> </ul>
Delete Pages	<ul style="list-style-type: none"> <li>as the Page Range</li> </ul>
Insert Duplicate Pages	<ul style="list-style-type: none"> <li>as the Page Range</li> </ul>
Resize	<ul style="list-style-type: none"> <li>as the Page Range</li> </ul>
Crop	<ul style="list-style-type: none"> <li>as the Page Range</li> </ul>
Pressmark	<ul style="list-style-type: none"> <li>in the Image Pressmark's filename</li> <li>as the Page Range</li> </ul>
Notify	<ul style="list-style-type: none"> <li>in the From email address</li> <li>in the To email address</li> <li>in the Custom Subject line</li> <li>in the Message</li> </ul>
Save	<ul style="list-style-type: none"> <li>in the Location</li> </ul>
Conditional	<ul style="list-style-type: none"> <li>in conditions that compare text</li> </ul>
External	<ul style="list-style-type: none"> <li>in the CLI Profile Parameters</li> </ul>

## Print Ticket Processing

When submitting jobs via MAX there are multiple sources of job ticketing that must be potentially reconciled. These are:

1. Print process node defaults
2. XPIF tickets embedded within an RDO file
3. XPIFs referenced in the Xpif file field or JDFs referenced in the File name field
4. Manifest fields – Job Name, Quantity and Stock name
5. Runtime print ticket adjustments - adjustments made by the Imposition, Duplicate Page Insert or External process nodes

Nominally, Process Manager will blend print tickets from all these sources into a single set of reconciled print instructions. The items with a larger number in the list above have higher precedence with item #5 having the highest precedence and item #1 having the lowest precedence. If there are conflicting job ticket parameters then the parameters from the item with the highest precedence are used.

Job tickets parameters are blended at several points during execution.

**Job Ticket Blending During Job Submission** - During job submission, Process Manager blends job ticketing from embedded XPIF files, referenced XPIF files and from mapped manifest fields. The result of this blending is a job print ticket that is used as the basis for further blending during execution. This job print ticket is not associated with a Print process node.

### Note

Process Manager job submission always results in the creation of a job print ticket regardless of whether a print ticket was submitted (e.g. when only manifest fields were used to specify print instructions) or whether the workflow contains Print process nodes.

**Job Ticket Blending During Job Processing** – Certain process nodes have the ability to manipulate the job print ticket during execution. Specifically:

- The imposition process node adjusts the print quantity based on the Nup number when doing repeated imposition. This behavior can be enabled / disabled.
- The imposition process node can also adjust the media size, sides imaged and orientation based on the imposed layout size. This behavior can also be enabled / disabled.
- The duplicate page insert changes the print quantity to 1 when inserting duplicate pages based on the print quantity.
- The Review process node copy count changes the job print ticket quantity.
- The External process node can export an XPIF ticket for external editing. When the External process node exports the job print ticket it expects an updated job print ticket from the external process. The internal job print ticket is replaced with the updated one from the external script or application.

**Job Ticket Blending During Job Printing or Saving** – When a job enters a Print process node the current settings in the job print ticket are blended with the Print process node defaults. This blended ticket is then used to submit the job to print. Any Save and External process nodes that follow the Print process node will save / export the blended job print ticket thus archiving the job as it was sent to the printer.

If the job enters another Print process node then the attributes that were blended in the previous Print process node are discarded. Only the job ticketing that was blended from the Print process node is discarded. Any updates to the job print ticket that happened after the previous Print process node (e.g. quantity adjustments by imposition) are retained.

After the previous Print process node attributes are removed from the job print ticket, the job print ticket is then blended with the attributes of the current Print process node. Save and External process nodes that follow this Print process node will save / export the job print ticket as it was submitted to the last printer.

Finally, if an operator edits the settings in a Print process node then there's no “blending” behavior. Subsequent Print process nodes will not discard job ticketing that was first introduced in a Print process node that was edited by an operator.

Note: The print process node has two options for disregarding the incoming job ticket. The **Always use these settings** option will ignore the incoming job ticket when printing. Similarly, if **Printer Default** option is enabled for the Quantity then the job print ticket quantity will not be sent to the printer.

## JMF Status Signals

FreeFlow Process Manager will send JMF status signals to an upstream system when submitting JDF jobs that contain JMF subscription information. JMF status signals are sent back whether the JDF submission is done via JMF, Hot Folder or Manifest processing.

Supported JMF status signals:

- InProgress – job is executing in a process node or printing
- Waiting – job is waiting for the printer
- Stopped – job is in the Review process node, ineligible for printing or held at the printer
- Completed – job successfully completed
- Aborted – job failed in a process node or was cancelled at the printer

JMF status signals are sent when:

- The status of a job changes
- The print counts have changed since the printer was last polled

### Note

JMF status signals for composite jobs reflect the status of the composite job and not the status of the individual components within the composite job.

## Best practices

### Use Well Understood Naming Conventions

JMF Status signals are sent when the status of a job changes. Avoid confusion by using descriptive workflow names, production methods, and file names.

### Document Your Implementation

Once you have completed an implementation, take the time to document what you have developed. This will reduce misunderstandings, save time, and eliminate waste in the future.

### Use Manifest Field Names for MAX Variables

When using MAX variables in a workflow's process node settings, use MAX field names instead of column positions. This prevents workflows from breaking if the manifest format changes. This also enables use of multiple MAX configurations with one workflow.

### Apply an End-of-Record Marker

When encoding an order jacket file, put an end-of-record marker in the column beyond the last possible data column. In the included example, the word 'end' is in column 16. This end-of-record marker ensures the shape of the manifest data, and forces status and status codes to be written in adjacent columns.

### Microsoft Excel and CSV Files

If possible, avoid opening CSV files in Microsoft Excel because it will automatically make the following changes to CSV field data:

- Remove leading zeroes from CSV fields that contain numerical values
- Reformat any value that happens to match a date

These changes may prevent the workflow from functioning as expected.

### Manifest File Size

For optimal processing, large order jacket files should be split into smaller files that represent fewer jobs.

## File Retrieval

FreeFlow Process Manager retrieves files from network servers using the XML\_ADMIN account. To ensure that this account can retrieve files from the network, perform one of the following:

- Add an XDL\_ADMIN account to the file server.
- If using Active Directory credentials, then add the XDL\_ADMIN account to the directory.
- Log in as XDL\_ADMIN and add credentials to the server using the Windows Stored Usernames and Passwords functionality.

### Note

Expired passwords prevent file retrieval.

## Troubleshooting File Retrieval Problems

If having trouble retrieving files, log into the Process Manager system as XML\_ADMIN and try one of the following:

- UNC – Perform a CLI directory listing to validate access to the shared folder (e.g. dir \\ServerName\ShareName\My Files\).
  - If you have access to the directory, copy the file from the CLI.
- HTTP/HTTPS – Open the URI in a web browser.
  - If the web browser prompts for authentication enter the credentials in the manifest. If the manifest does not have credentials, use the windows XDL\_ADMIN credentials.
- FTP – Connect to the server using the CLI FTP client. If successful then attempt to download the file from the manifest.
  - If using an FTP proxy, first connect to the proxy and then connect to the remote FTP server per the specific proxy's configuration.
- RMS – Use the “Test” functionality in the FreeFlow Administration Tool to validate connectivity to the repository.



# Walk Through Example

# 4

## Summary of Chapter

- This section provides a tutorial, an end-to-end walk through example. Topics for this example include:

Explore the Esv9 - Personalized Cards example

- Demonstration of the Example
- Example discussion

### Explore the Esv9 - Personalized Cards example

The walkthrough example uses the Esv9-07 Personalized Cards workflow from the FreeFlow Process Manager Easy Start workflows. Sample manifests and other resources for the Easy Starts are in the C:\PM Easy Starts v9\ folder.

This sample workflow uses MAX for job submission.

### Manifest file

Open Esv9-07 Personalized Cards.csv. This example uses a delimited file to enable automated production of personalized postcards. Automated production can be enabled from any upstream system that can produce a manifest file with the required data.

**Note:** Do not make any changes or the example may not work properly.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
	#Production Method	Order Number	File name	Folder		Copy Count	Job Join					PO Number								Shc Orc Nar
1	Esv8-07	1	Postcard Furn C:\PM Easy Starts v8\Sa			1	1					Furniture-0001								ES-
2	Esv8-07	1	Postcard Furn C:\PM Easy Starts v8\Sa			1	1					Furniture-0001								ES-
3	Esv8-07	1	Postcard Furn C:\PM Easy Starts v8\Sa			1	1					Furniture-0001								ES-
4	Esv8-07	1	Postcard Furn C:\PM Easy Starts v8\Sa			1	1					Furniture-0001								ES-
5	Esv8-07	1	Postcard Furn C:\PM Easy Starts v8\Sa			1	1					Furniture-0001								ES-
6	Esv8-07	1	Postcard Furn C:\PM Easy Starts v8\Sa			1	1					Furniture-0001								ES-
7	Esv8-07	1	Postcard Furn C:\PM Easy Starts v8\Sa			1	1					Furniture-0001								ES-
8	Esv8-07	1	Postcard Furn C:\PM Easy Starts v8\Sa			1	1					Furniture-0001								ES-
9	Esv8-07	1	Postcard Furn C:\PM Easy Starts v8\Sa			1	1					Furniture-0001								ES-
10	Esv8-07	1	Postcard Furn C:\PM Easy Starts v8\Sa			1	1					Furniture-0001								ES-

When the Esv9-07 Personalized Cards.csv file is put in the Esv9-MAX hot folder located at C:\PM Easy Starts v9\HFfs\, MAX uses the hot folder options to determine which columns should be used to find the Production Method. In this example, the hot folder is configured to use the value from column 1.

**Manifest Automation from Xerox - Hot Folder Options**

Hot Folder: C:\PM Easy Starts v9\HFfs\ESv9-MAX

Job Submission | Processing Options | Conversion Script

**Production Method Field:**  
Select the columns in the order jacket file to use in determining the Production Method.

Enter a column position:  
0 Add

Fields are joined in the order shown:  
1 Move up Move down Delete

**Submission Options:**  
Select how order items groups and multiple files in folders are submitted to workflows.  
Submit groups of files as a single job only if the workflow includes a join node

**Production Methods:** Production methods values in the manifest are mapped to MAX configurations and to Process Manager Workflows. These mappings determine which MAX configuration to use to parse a manifest record and which Process Manager workflow to use to process the job submitted from processing this manifest record.

**Manifest Automation from Xerox - Configuration 2 ( Esv9-01, Esv9-06, Esv9-07 and Esv9-10 )**

Configuration 1 | Configuration 2 | Configuration 3 | Configuration 4 | Configuration 5 | Configuration 6 | Configuration 7 | Configuration 8 | Cor

Manifest Automation from Xerox

- Order Jacket File
  - Mandatory Fields
  - Optional Fields
- Job Submission
  - Job Name
  - Production Methods
- Bill of Materials
  - Bill of Materials Output
  - Bill of Materials Name
  - Standard Fields
  - Custom Fields

**Production Methods**

Production methods in the order jacket file must be associated with a Process Manager workflow. Printjobs and orders associated with a given production method will be submitted to the workflow you associate with it here.

Enter a production method name:  
\_\_\_\_\_

Choose a workflow to map to this production method:

- 1:Simple Workflow (Disabled)
- 2:Main Workflow (Disabled)
- 3:Reprint Workflow (Disabled)
- 5:Bill of Materials Workflow (Disabled)
- 6:ESv9-01 CSR Workflow (Disabled)
- 7:ESv9-02 Books (Disabled)
- 8:ESv9-03 Booklets (Disabled)
- 9:ESv9-04 Business Cards (Disabled)
- 10:ESv9-05 Ganged Business Cards (Disabled)

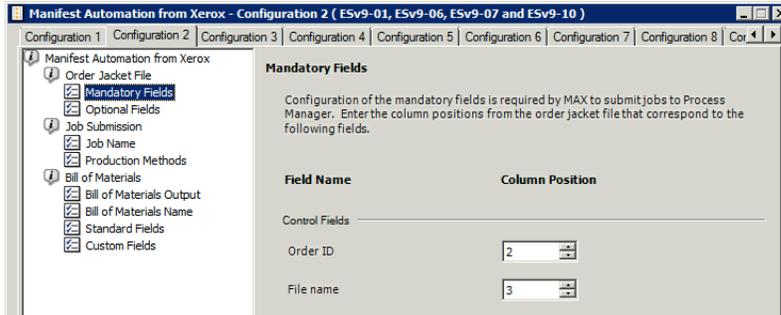
Add Mapping

Production Method Mappings:

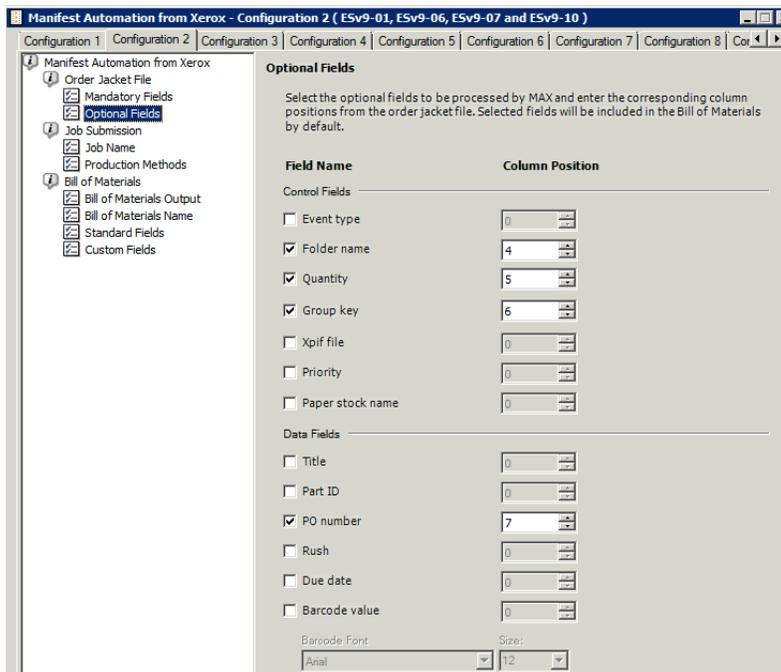
Production Method	Workflow
ESv9-07	12:ESv9-07 Personalized Cards (Disabled)
ESv9-06 Booklets	13:ESv9-08 Booklets and Calendars (Disabled)
ESv9-10	15:ESv9-10 Color Cover, Mono Block Books (Disable
ESv9-01	6:ESv9-01 CSR Workflow (Disabled)
ESv9-06 Cards	11:ESv9-06 Cards (Disabled)

Manage Mappings Delete Mapping

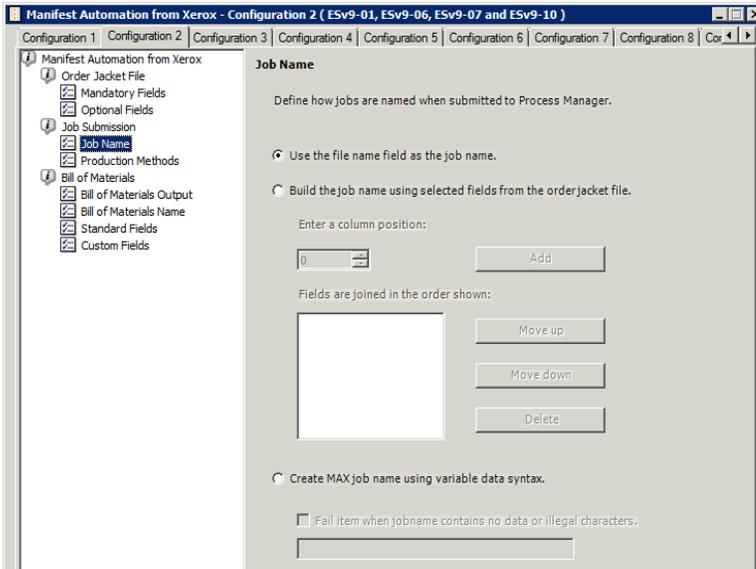
**Mandatory Fields:** This screen is where you map columns from your manifest to mandatory Manifest Fields. In this example for the Esv9-07 Personalized Cards configuration the Order Id is mapped to column 2 and the File name is mapped to column 3.



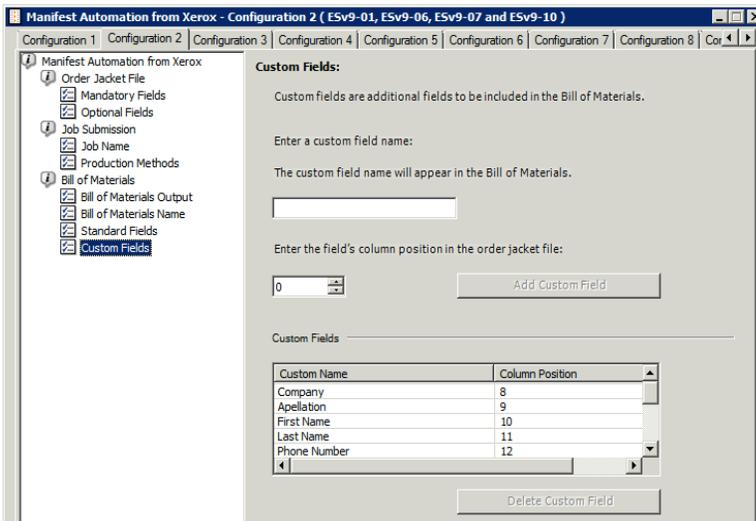
**Optional Fields:** This screen is where you map columns from your manifest to Optional Manifest Fields. Notice that the Manifest Fields are mapped to column positions in the manifest.



**Job Name:** This screen is where you configure the job name to be displayed within Job Manager. You can use the file name as the job name, or build a job name from any column, or combination of columns from the manifest or create a MAX job name using variable data syntax.



**Custom Fields:** This screen is where custom manifest fields are mapped from the manifest. Custom manifest fields can be used within Process Manager for a variety of reasons. Here we will map custom fields named: Last Name, First Name, Company Name, etc. This example shows the usage of the custom fields as MAX variables in the Watermark and Barcode process nodes.



## Demonstration of the Example

### Watch

- Open the FreeFlow Job Manager to watch the jobs you submit through MAX.

### Invoke

- Copy the **ESv9-07 Personalized Cards.csv** sample order jacket file from the directory **C:\PM Easy Starts v9\** into the hot folder directory **C:\PM Easy Starts v9\HFs\ESv9-MAX**.

## Witness

After a short period MAX processes the manifest and jobs appear in the Workflow Job Manager. Notice that the jobs are aggregated into a single composite job – you can see the jobs in the composite job by double clicking on the composite job.



---

## Completion

- In this example, Review nodes were placed at the end of the workflow. Once the jobs in the Job Manager have completed, an Attention Required status is displayed. You can click on any job to see the job's workflow, and then click the Review button to see the document in its final state.

- Here is an example of the job output:



## Example discussion

### What did the manifest example do?

By copying the manifest named Esv9-07 Personalized Cards.csv into the hot folder you invoked the FreeFlow Process Manager Manifest processing interface.

### Hot Folders

FreeFlow Process Manager monitors the hot folders associated with the special Submitted With Documents target workflow. Once a manifest is detected in the hot folder's directory, MAX reads the records of the manifest and processes them.

**Manifest Automation from Xerox - Hot Folder Options**

Hot Folder: C:\PM Easy Starts v9\HFa\ESv9-MAX

Job Submission | Processing Options | Conversion Script

**Production Method Field:**  
Select the columns in the order jacket file to use in determining the Production Method.

Enter a column position:

0 [Add]

Fields are joined in the order shown:

1 [Move up] [Move down] [Delete]

**Submission Options:**  
Select how order items groups and multiple files in folders are submitted to workflows.

Submit groups of files as a single job only if the workflow includes a join node

### Production Methods

In this example, the Esv9-MAX hot folder was configured to look for a Production Method in column 1 of the CSV file. MAX reads the production method value for each record in the manifest file and determines which MAX configuration contains a Production Method mapping that uses the value from the manifest file. This manifest configuration is used to process the manifest record. The resulting job is submitted to the workflow specified in the Production Method mapping.

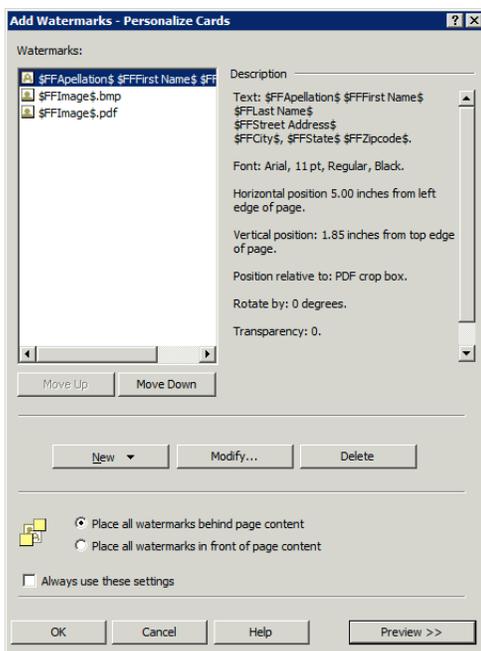
## Manifest Records

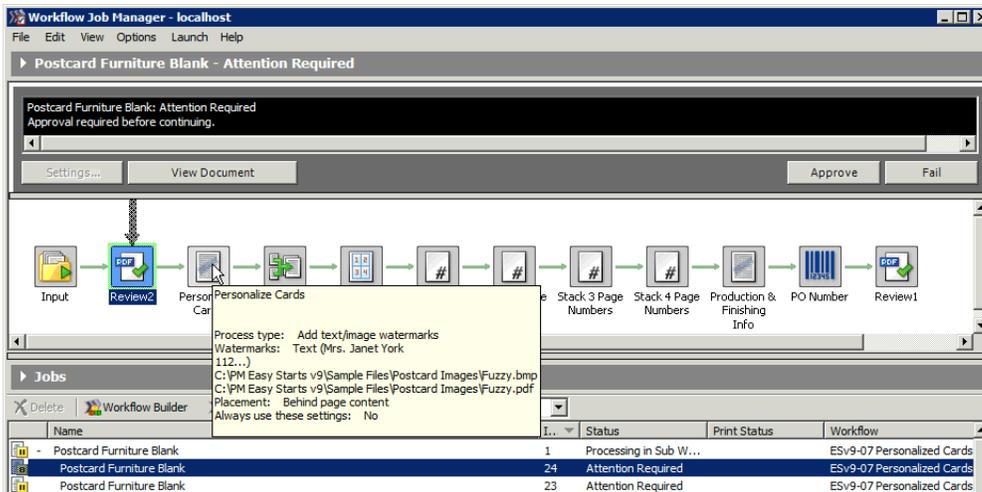
Each manifest record is read from the manifest and processed according to the rules established in the Manifest Configuration Tool. Field values from the record enable FreeFlow Process Manager to use the appropriate workflow, and to apply record-specific details to a variety of FreeFlow Process Manager process nodes.

In this example, multiple records are submitted as a single job. Contiguous records in a Manifest file that share a common production method, order number and file group key are submitted as a single composite job to FreeFlow Process Manager. Individual files in a composite job are processed independently in the workflow. In this example, the files are joined into a single PDF.

## MAX Variables

In Workflow Builder, locate the workflow named ESv9-07 Personalized Cards and open the first Watermark node. You will see MAX variables (\$FF...\$) defined in the image watermark image path and in the watermark text. During workflow execution, these MAX variables are replaced with the values from the configured manifest fields. The resulting watermark in the output document is personalized with content from each manifest record.





Watermarks placed before the join process node are applied to each the PDF in the composite job individually. These watermarks use data from the Manifest record that was used to submit the specific PDF file. Watermarks placed after the join process node are applied to the joined PDF. These watermarks use data from the Manifest record used to submit the first PDF in the composite job.

MAX variables may be specified as either `$$$Field Name$`, `$$$CSVField Name$` or as `$$$CSV##$`. For example:

- `$$$PPO number$` creates a watermark using data from the manifest column that was defined as the source for the PO number Manifest Field in the MAX configuration Optional Fields configuration.
- `$$$CSVFirst Name$` creates a watermark using data from the manifest column that was defined as the source for the First Name in the Bill Of Materials Custom Fields.

MAX variables should be defined using field names whenever possible. This allows multiple configurations to submit jobs to the same workflow. Also, changes in manifest can be accommodated by changing the configuration without changing the process nodes.

## Status

Once MAX has completed parsing the manifest, a status file appears within a sub folder (Manifest Status) of the Hot Folder. The status file has the same name of the submitted file but adds status columns to the order jacket file.

Open your status file, named: “ESv9-07 Personalized Cards.status.csv” which can be found in **C:\PM Easy Starts v9\Hot Folders\ESv9-MAX\Manifest Status**. The manifest status file now has 2 additional columns of data. The first column is a status code, in the example below it is Success. The second column is a textual description of the status code, in the example below it is S1. Here they are displayed as columns R, and S.

