ORACLE
INSURANCE

Oracle® Documaker

Docusave Server Administrator's Guide

version 3.0

Part number: E15145-01

January 2009

THIRD PARTY SOFTWARE NOTICES

This product includes software developed by Apache Software Foundation (http://www.apache.org/).
THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes software distributed via the Berkeley Software Distribution (BSD) and licensed for binary distribution under the Generic BSD license.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes software developed by Jean-loup Gailly and Mark Adler. This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

This product includes software developed by the Massachusetts Institute of Technology (MIT).
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This software is based in part on the work of the Independent JPEG Group (http://www.ijg.org/).

This product includes software developed by Sam Leffler of Silicon Graphics.
THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE

This product includes software components distributed by Steve Souza.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS "AS IS'" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright © 2002, Steve Souza (admin@jamonapi.com). All Rights Reserved.

This product includes software components distributed via the Berkeley Software Distribution (BSD).

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS'" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright © 2000-2006 www.hamcrest.org. All Rights Reserved.

This product includes software components developed by the Independent JPEG Group and licensed for binary distribution under the Independent JPEG Group license.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS'" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright © 1994-1998 AIIM International. All Rights Reserved.

This product includes software components developed by Sam Stephenson.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2005-2007 Sam Stephenson

This product includes software components developed by Sun Microsystems.

Copyright (c) 1995-2008 Sun Microsystems, Inc. All rights reserved.

This product includes software components distributed by Vbnet and Randy Birch.

Copyright © 1996-2008 Vbnet and Randy Birch. All Rights Reserved

---

This product includes software components distributed by the Internet Software Consortium and IBM.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 1996 by Internet Software Consortium.

THE SOFTWARE IS PROVIDED "AS IS", AND IBM DISCLAIMS ALL WARRANTIES, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL IBM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE, EVEN IF IBM IS APPRISED OF THE POSSIBILITY OF SUCH DAMAGES.

Portions Copyright (c) 1995 by International Business Machines, Inc.

This product includes software components distributed by RSA.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

Copyright © 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

This product includes software components distributed by Terence Parr.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2003-2007, Terence Parr. All rights reserved.

This product includes software components distributed by Computer Associates.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2002 Computer Associates. All rights reserved.

This product includes software components distributed by MetaStuff.

THE SOFTWARE IS PROVIDED BY METASTUFF, LTD. AND CONTRIBUTORS "AS-IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDINGBUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL METASTUFF, LTD. OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS AND SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, ARISING IN ANY WAY OUT OF THE USE OF THE SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes software components distributed by JSON.

The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product includes software components distributed by OpenSSL (http://www.openssl.org/).

THIS SOFTWARE IS PROVIDED BY THE OPENSSL PROJECT "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OPENSSL PROJECT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA OR PROFITS; OR BUSINESS INTERRUPTION) HOWVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE

This product includes software components distributed by Yahoo! Inc.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS'" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Publication history

Initially released for /NT 2.0.4 in November of 2001.

Revision	For	Date
1	/NT v2.0.4/6000 v2.0.4a	GA: September 2002
2	/NT v2.0.5 GA/6000 v2.0.4e	GA: April 2003
3	/NT v2.0.8 GA/6000 v2.0.4e	GA: August 2003
4	/NT v2.0.8 GA/6000 v2.0.4e GA/ 390 v2.0.9	GA: January 2004
5	/NT v3.0 GA/6000 v2.0.4e GA/390 v2.0.9	GA: May 2006
6	v3.0 for Windows, AIX, Solaris, and Linux	September 2006
7	v3.0 for Windows, AIX, Solaris, and Linux	April 2008
8	v3.0 for Windows, AIX, Solaris, and Linux	July 2008
9	v3.0 for Windows, AIX and Linux	January 2009

Docusave Server documentation roadmap

Administrating

Installation Guide

Server Administrator Guide

Server Help

Programming

Programmer's Guide

Table of Contents

Preface

Welcome

Skywire Software offers a comprehensive range of scalable high-performance products for every step in the life cycle of a document. These include creation solutions to capture data and create forms, publishing solutions to produce large volumes of personalized documents, **archival solutions** to intelligently store and retrieve documents, management solutions to control and network documents, and development tools to customize your solutions implementations and interfaces.

Skywire Software archival solutions store your documents electronically for intelligent retrieval and viewing. Archival solutions facilitate immediate access to your documents for such applications as claims processing, enterprise-wide contract or regulatory document lookup, call center statement reference, and other internal and customer service-based functions. You can implement archival solutions as standalone systems or integrate them with leading imaging systems.

A key component of Skywire Software's archival solutions is Docusave Server™. This manual describes Docusave Server and the procedures for using it.

Using this manual

This *Administering Docusave Server* manual is written for administrators of the Docusave Server application. This manual describes in detail how to use Docusave Server on multiple platforms to receive print data streams as input and store them in a document archive.

Conventions

The *Administering Docusave Server* manual provides consistent typographic conventions and keyboard formats to help you locate and interpret information easily. These conventions are provided below.

Typographic and Keyboard Conventions

Convention	Description
Italics	Command, dialog box, button, and field names
Arial font	Directory, folder, and file names
1 **Numbered lists**	Provide step-by-step procedures for performing an action
◆ Bulleted lists	Provide grouped information, not procedural steps

Introduction

What is Docusave Server?

Introduction

Docusave Server automates document filing and batch retrieval.

When performing document filing, Docusave Server processes jobs, which consist of groups of documents, separates them into individual documents, converts and optionally compresses each document, and then automatically files the documents into an archival imaging or document management system.

When performing batch retrieval, Docusave Server fetches previously archived documents for subsequent processing by another process (like Docucreate IC) or for routing print stream documents back to a production printer. It does this by using two types of records:

- Document RePrint records (DRP records)—Docusave Server retrieves a compressed AFP or Metacode print stream and its dependant printer resources from the archival imaging or document management system for subsequent printing.

- Document Retrieval Requests (DRR records)—Docusave Server retrieves page images from an archival imaging or document

management system and exports them in a format accepted by Docucreate IC.

NOTE: See "Appendix G: Glossary" on page 161, for terms used throughout this book.

Functionality

Docusave Server provides a flexible framework and offers many business solutions such as print stream rasterization, print stream archiving, resource archiving, document reprint and document retrieval.

Solution	Description	Inputs	Outputs
Print Stream Rasterization	Converts print stream files into individual image files, compresses, and files them into the selected storage repository. (Local copies of the printer resources are used during the image file generation, but need not be stored.)	AFP Compound Metacode	Separate (compressed) image files - TIFF
Print Stream Archiving	Divides print stream files into individual printable and viewable documents, optionally compresses, and files them into the selected storage repository. (Check that the printer resources needed to display documents are present in the storage repository.)	AFP Compound Metacode DCD	AFP DCD Separate Compressed Metacode
Resource Archiving	Stores a reference copy of a printer resource into the storage repository prior to print stream archiving.	Fonts Overlays	Fonts Overlays

Solution	Description	Inputs	Outputs
Document Reprint	Extracts a compressed print stream and its dependant resources from the storage repository.	Document RePrint Records	Reprint Package
Document Retrieval	Extracts an individual file from the storage repository.	Document Retrieval Requests	Megafiles

Printer Resources and Docusave Server

Several of the common operations listed above utilize printer resources, but at different times and in different ways:

- Rasterization uses a local copy of printer resources during the rasterizing process, but not afterwards.

- Archiving requires a single reference copy of each resource cited in the print stream file to be archived into the storage repository for later use. When the archived print stream document is viewed, the viewer (e.g., Documanage Workstation or Docuview LFS) includes logic to locate the correct version of the printer resource for accurate display (or printing) of the original document. Before print stream archiving completes, Docusave Server checks and makes sure every needed printer resource is already archived.

- As a convenience, when archiving printer resources into the storage repository, there is an option to also put a copy of each archived printer resource into a specially formatted “resource cache” for subsequent use by a viewer application, such as Docuview LFS. (This is now used primarily for backward compatibility.)

During these operations, printer resources can be stored in several distinct formats and locations:

- In a local disk directory for use during rasterization operations.
- In the Docusave Server queue, as placed by the Addres utility, prior to resource archiving operations.
- Inside a storage repository, after successful resource archiving operations.
- In a "resource cache" created during resource archive operations, for later use by a viewer application (e.g., Docuview LFS).

Use and Structure

Docusave Server can be used as part of any business solution that requires automated processing (typically archiving or batch retrieving) of significant volumes of documents or document oriented transactions. Additionally, Docusave Server provides transaction level logging, error recovery, and custom library support to meet unusual or changing business needs.

The custom capabilities of Docusave Server enable novel solutions and allow easy development of:

- Custom output destinations (e.g., new imaging or archival system destinations)
- Custom index processing including validation, "pre-flighting," and/or possibly correcting indexes before a document is filed
- Custom pre-, post-, and processing steps, such as file type conversion, data validation, and error recovery

Docusave Server takes its primary input from a queue. Use of a queue allows business solutions to span separate computing platforms, where the applications that create documents reside on a different computer system (or

even a different type of system) than Docusave Server itself. Another advantage of queues is that the separate processes feeding a queue can be started and stopped independently of Docusave Server. Using queues also allows the number of instances of Docusave Server to differ from the number of processes feeding the queue, which permits load balancing and system scaling. In short, use of an input queue provides Docusave Server a single interchange point for improved platform connectivity, independent operation, load balancing, and system scaling.

The elements that reside in a Docusave Server input queue are groups of documents or transactions in a single unit, called a job. A job can contain one or many documents. One important function of Docusave Server is to split jobs into separate documents for individual processing, and to make sure the job process is complete. An axiom of Docusave Server design is that entire jobs either succeed or fail. Error recovery processing is provided to make sure a job does not partially succeed.

The “all or none” approach to document processing within a job provides important operational simplicity. It eliminates messy recovery in situations where part of a job was successfully processed, and part of it failed. “Roll-back” logic is included in Docusave Server so that jobs either succeed entirely, or the system is rolled back to its state before the job was started. (If some documents were processed in a job that fails, roll-back makes sure the documents are deleted from the output destination).

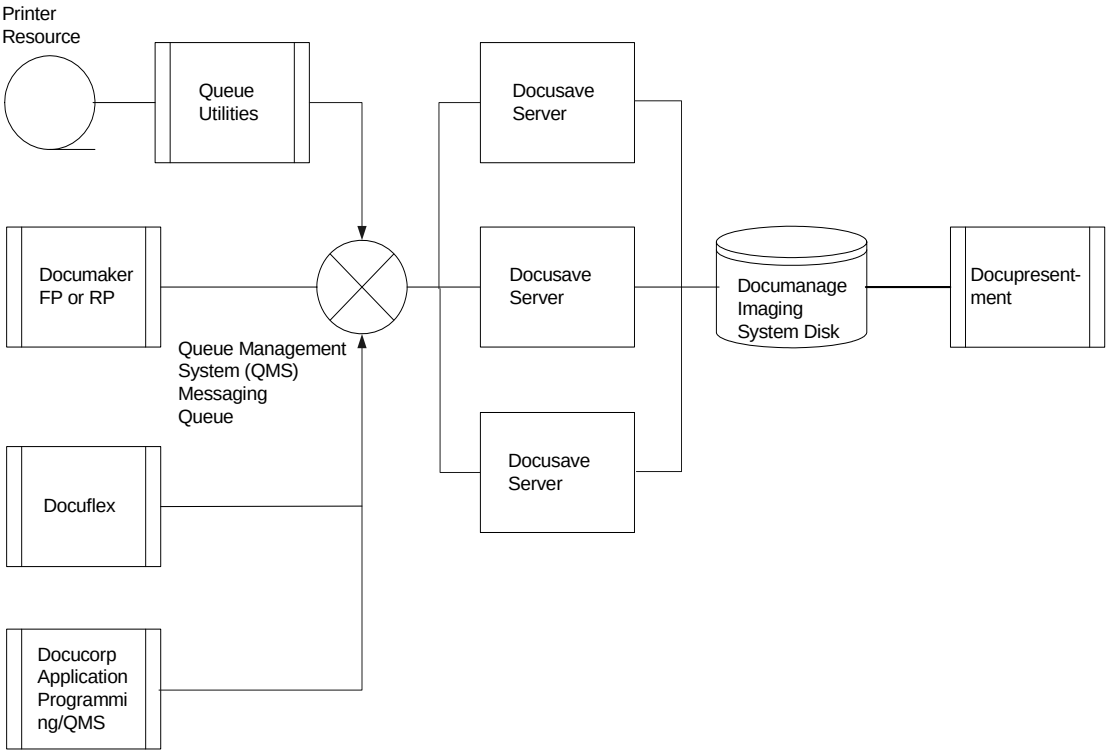
If documents are intended to be processed independently, and not as a combined job, they can be queued in separate jobs of one document. (Note: Some queue implementations have performance implications or system limitations on the number of jobs in a queue. Keep this in mind when putting many small jobs in a queue.)

Docusave Server in Context

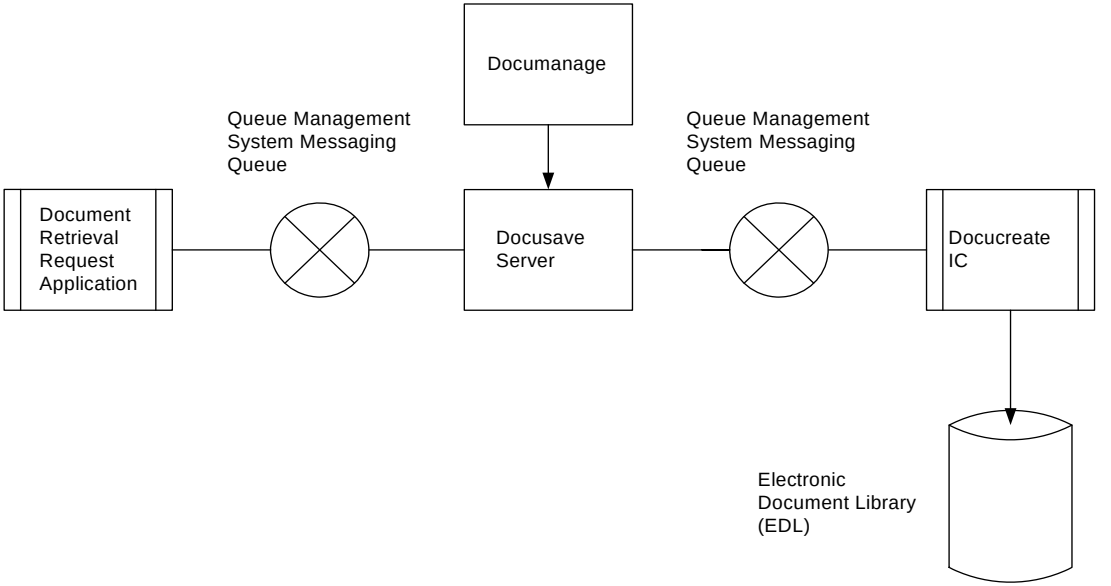
In a filing scenario, Docusave Server is often used in conjunction with Documaker or Docuflex publishing solutions. Typically the output of Documaker or Docuflex is directed to a Docusave Server input queue. Docusave Server in turn often writes its output to an archive or document management system such as Documanage.

Together, Documaker/Docuflex, Docusave Server, Documanage, and Docupresentment, with their associated tools, form the architectural foundation for a number of complete business solutions.

Docusave Server can be part of a system designed to process very high data volumes. It may be desirable to run multiple instances of Docusave Server concurrently for increased throughput. Typically, multiple publishing processes feed a single input queue from which multiple instances of Docusave Server process jobs in parallel.



In a batch retrieval scenario using DRR records, Docusave Server runs “in reverse” to extract page images from an imaging or document management system and emits them as “megafiles” into a queue for processing by Docucreate IC.



Flow of Control

Docusave Server follows the same sequence of processing steps regardless of variations in processing content and configuration. For example, the file type translation, compression choice, and destination archive or imaging system can all be changed without changing the Docusave Server program, its internal structure, or process flow.

Within the general flow, there are a number of places where Docusave Server offers processing options. Some options are controlled by a user interface (or the equivalent configuration files or input control statements). Other processing options are determined by library selection, which can be static or dynamic.

Dynamic library selection is governed by two attributes of the job being processed. A job recognition library examines a job in the input queue and returns two processing types: the Job Type and Process Type (also known as the Input type and Output type, respectively). These two types are represented with three letter codes, which trigger the naming conventions for processing libraries. For example, ‘RAS’ designates rasterization, and ‘PRT’ designates print stream format. The job and process type are used to dynamically select and load the libraries for job and document processing steps.

Summarized, the internal processing flow of Docusave Server is as follows:

- Initialize and begin monitoring an input queue for work units, called “jobs”
- Wait for a job in the input queue
- For each job in the input queue:
  - Call a “job recognition” library to determine the processing requirements, and set the corresponding Job Type and Process Type
  - Call the job pre-processor function (if any)
  - Call the job processor function which bursts the job into separate documents (transactions). If the job contains documents that need to be processed individually, the job processor will call the following document processing libraries
    - Document pre-processing library (if any)
    - Document processing library
    - Document post-processing library (if any)
  - Call the job post-processor library (if any)
  - Copy the input job to the error queue (only if an error condition is detected)
  - Remove the current job from the input queue
  - Start the next job in the input queue, or resume waiting for more

- Upon exiting, release memory, network, and system resources, close files, and return to the operating system

The normal inputs and outputs to these process are:

- Input queue (into which jobs are placed by other processes)
- Configuration settings, gathered from a User Interface in the NT version, otherwise from control card settings or configuration files
- Output to log files that show jobs and documents successfully processed, and their disposition
- Output to log files that show jobs and/or documents which failed to process, the reason for failure, and the action taken or recommended
- Output to the chosen processing destination, usually an archival imaging or document management system, but sometimes simply as files on disk
- Megafiles output to a queue, formatted for processing by Docucreate IC.
- Reprint packages output to disk, formatted for subsequent printing.
- Output and Input among various temporary files during processing

Although Docusave Server can be configured many different ways, certain configurations using the default libraries cause it to perform the specific functions of previous products:

- Accesscommander functionality can be achieved by using Docusave Server to process AFP or Metacode print streams, or stacked DCD documents. This processing includes bursting jobs into separate documents, analyzing resource requirements, ensuring required resources are accessible, performing DJDE normalization (if applicable), and

archiving to a supported imaging system. In addition to documents, printer resources can be archived using Docusave Server, fulfilling another Accesscommander function.

- Imagecreate functionality converts AFP or Metacode print streams into TIFF (or similar) image files before archiving those into an imaging system.

Other configurations are possible when used with custom libraries and provide virtually unlimited flexibility in processing.

Library Overview

NOTE: All libraries are supported by all operating systems (Microsoft Windows, AIX, Linux, and Solaris) unless otherwise specified.

When Docusave Server is started it loads and initializes the following libraries:

- DSJOBTYP – for recognizing the contents of input jobs
- CODECMGR – for managing the compression of documents and the decompression of input jobs
- DSISYSxx – for storing/retrieving documents to/from the imaging system. Each imaging system will have its own library, and the standard installation includes support for:
  - DSISYSD - Disk (local file system)
  - DSISYSDM - Documanage
  - DSISYSFL - Output File
  - DSISYSFN - FileNet: Windows, AIX
  - DSISYSIC - ImageCreate Output: Windows, AIX
  - DSISYSM - MARS/NT:Windows
  - DSISYSV - VLAM EDL: Windows
  - DSISYSMQ - Windows

Docusave Server is often used with a publishing solution such as Documaker or Docuflex. The output of Documaker or Docuflex is directed to a Docusave Server input queue. When a job is extracted from the input queue the job recognition library is called to identify

the job type. It returns a three-character string that uniquely identifies the contents of the job. Docusave Server will then use this string as a suffix to construct the names of the job pre-processing library, the job processing library, and the job post-processing library. The names will have following format:

- DSPRJxxx – pre-process job
- DSPJBxxx – process job
- DSPOJxxx – post-process job

For example, if the job-typing library returns the string "**ABC**," then Docusave Server would attempt to load the three libraries named

- DSPRJ**ABC**
- DSPJB**ABC**, and
- DSPOJ**ABC**

The job processing library is the only library that must exist. The pre/post-job/document processing libraries are optional and are not supplied with the standard installation; they are custom libraries.

The following job processing libraries are supplied with a standard installation:

- DSPJB**DCD** – for processing DCD input
- DSPJB**DRP** – for processing DRP input
- DSPJB**DRR** – for processing DRR input
- DSPJB**FPP** – for processing DMFPPP logs
- DSPJB**IMC** – for processing ImageCreate output
- DSPJB**PCL** – for processing PCL input
- DSPJB**PRT** – for processing AFP and Metacode input
- DSPJB**RSC** – for processing printer resources

A job processing library calls document processing libraries if the job contains documents that need to be processed individually. The following document processing libraries will be supplied with a standard installation:

- DSPDC**DCD** – for archiving DCD documents
- DSPDC**MRG** - for transferring normalized AFP/Metacode Documerge forms to a mainframe VLAM library.
- DSPDC**NUL** - for transferring AFP/Metacode documents to a mainframe VLAM library as-is
- DSPDC**PCL** - for archiving PCL documents
- DSPDC**PRT** – for archiving AFP/Metacode documents
- DSPDC**RAS** – for rasterizing AFP/Metacode documents

Once jobs are processed, they are output to an imaging system and Docusave Server begins processing the next job in the queue or resumes waiting for more.

General Setup

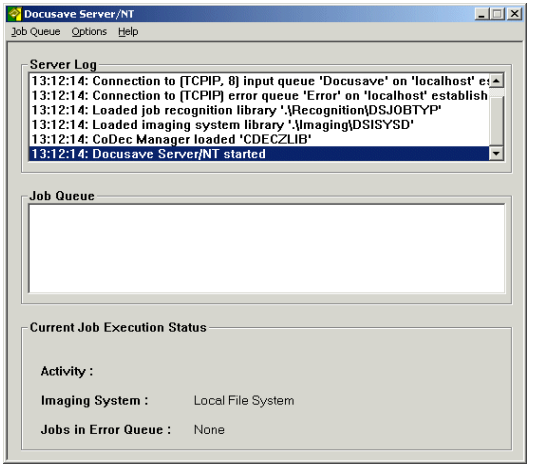
Introduction

For each platform there is general setup to perform. For Microsoft Windows, use the graphical user interface to complete setup. On the UNIX compatible operating systems (AIX, Linux, and Solaris), you must edit the Docusave Server profile to customize the entries to match your site specific information. All methods are described below.

NOTE: If your Docusave Server application is installed with Documanage, verify that Documanage Server is operating properly after general setup using the checklist in "Appendix F: Installation Verification for Documanage" on page 159.

Windows

Docusave Server consists of the main screen and several setup subscreens. On the main screen you'll find the Server Log window, the Job Queue window, and the Current Job Execution Status window.



The Server Log window displays a message for each stage of the processing. You can see the point at which the queue is started, the job servicing is started, the actual job is serviced, and the job is completed.

The log in the Server Log window is the is the main application log file which logs all Docusave Server processing steps. The log file is named DSyyyymmdd.log where *yyyy* is the year, *mm* is a two digit number representing the month, and *dd* is a two digit number presenting the day on which the information is logged. The log file is written by default to the application directory. The default log file location can be changed by entering a new directory location on the Log Directory entry on the Docusave Server Setup dialog.

NOTE: When the job is successfully completed, the Server Log window reports a Return Code of 0 (RC=0). Any other return code signifies there was a problem with storing the job.

The Job Queue window displays a list of all jobs in the queue and related information, such as the date and time the job was submitted, its source, status, and a description of the job.

Docusave Server polls its input queue every few seconds for image jobs. For an immediate check on the queue, select *Job Queue|Refresh Job Queue List.*

By default, when Docusave Server is initially started, the processing of the queue is suspended so you can always view the current jobs in the queue. This default behavior can be changed on the Docusave Server Setup Screen by clicking *Activate Input Job Queue Processing at Docusave Server Startup*.

When you're ready to restore the processing on the queue, select *Job Queue|Start Processing Jobs.* You can halt the processing on the queue by selecting *Job Queue|Stop Processing Jobs*.

The *Current Job Execution Status* window indicates the current stage of conversion and displays a message explaining the current processing stage. Docusave Server informs you of the presence of any erroneous jobs in the error queue. If all processing is completed normally, the value of the Error Queue is *None*. If not, Docusave Server displays the number of jobs in the Error Queue.

From the main screen *Job Queue* menu, you can start and stop processing jobs and refresh the job queue list.

NOTE: For information about Queue Setup see "Queue Setup" on page 21. For information about library setup see "Docusave Server Setup" on page 25.

UNIX

The Docusave Server profile (.docusave server) contains specific settings for use when processing jobs for your imaging system. You can use any editor, such as ***v*i** to customize the settings. Edit the .docusave server file with your site specific information before starting Docusave Server.

When launching Docusave Server from a script or process, you must establish the directory containing the Docusave Server application as the current working directory before starting Docusave Server.

Start the server by entering the following command. The server runs as a process until stopped.

To start the server in the...	Enter
Foreground	start_docusave server
Background	start_docusave server &

Stop the server by terminating the process. The server will stop at the next job boundary.

For a...	To terminate the server...
Foreground operation	Press CTRL+C
Background operation	Enter `kill <process #>`

Queue Setup

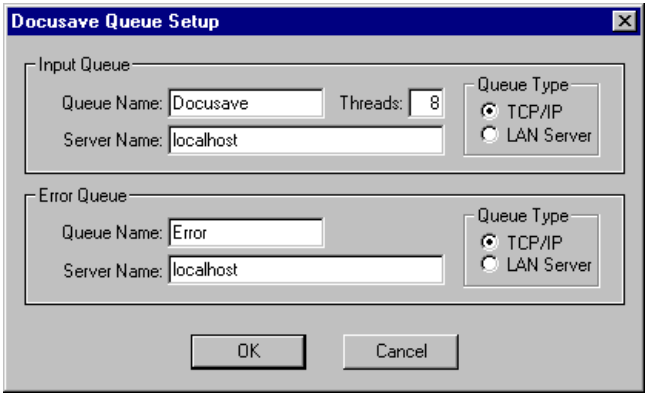
When you set up queues in Docusave Server, you instruct Docusave Server where to find the Input and Output (Error) Queues. A queue is a holding area for jobs awaiting processing.

WARNING: You can only record existing queues in this routine (i.e., queues must be previously set up). If you need to create a queue on your server, please refer to the *Docutoolbox Docucorp Queue Systems* guide.

To set up queues

Windows

1 **Go to Options|Queue Setup.**
The Docusave Server Queue Setup screen opens.



2. **Enter the name of the queue from which the input is routed in the *Input Queue Name* text box.**

   The Input Queue is the input queue for Docusave Server.

3. **Enter the server name where the input queue resides in the *Input Server Name* text box.**

   You can enter “localhost” for the Server Name if the input queue is on the same machine as Docusave Server.

4. **Enter the number of input threads in the *Threads* text box.**

   The thread count is the maximum number of jobs that are concurrently read from the queue.

5. **Select either *TCP/IP* or *LAN Server* in the *Input Queue Type* box.**

   This specifies how you communicate with the Queue daemon. TCP/IP uses a TCP/IP protocol. LAN Server uses “named pipes.”

6. **Enter the name of the queue to which the erroneous job is routed in the *Error Queue Name* text box.**

7. **Enter the server name where the error queue resides in the *Error Server* text box.**

   The Error Queue is a storage destination for any jobs that couldn’t process normally.

   You can enter “localhost” for the Server Name if the error queue is on the same machine as Docusave Server.

8. **In the *Error Queue Type* box, select either *TCP/IP* or *LAN Server*.**

   This specifies how you communicate with the Queue daemon. TCP/IP uses a TCP/IP protocol. LAN Server uses “named pipes.”

9 **Click *OK* to save your changes or *Cancel* to exit without saving.**

UNIX

Field	Explanation
QueueName	The name of the queue from which the input is routed. Default = Docusave Server
QueueType	How you communicate with the Queue daemon. TCP/IP uses a TCP/IP protocol. Default = `TCPIP`
ServerName	The server name where the input queue resides. Enter “localhost” for the Server Name if the input queue is on the same machine as Docusave Server. Default = `localhost`
JobBuffers	The maximum number of jobs that are concurrently read from the queue. Default = `8` Inputs = `0-8`
ErrorQueueName	The Error Queue is a storage destination for any jobs that couldn’t process normally. Default = `Error`
ErrorQueueType	How you communicate with the Queue daemon. TCP/IP uses a TCP/IP protocol. Default = `TCPIP`

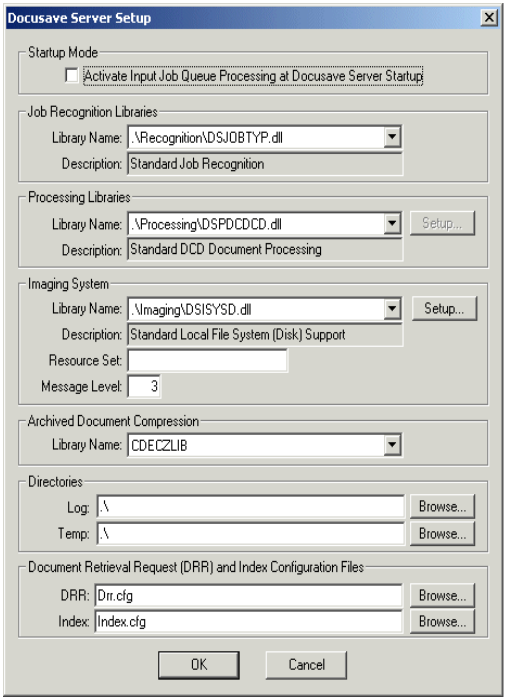
Field	Explanation
ErrorServerName	The server name where the error queue resides. You can enter “localhost” for the Server Name if the error queue is on the same machine as Docusave Server. Default = `localhost`

Docusave Server Setup

The Docusave Server Setup screen lets you define and configure the specific Docusave Server libraries. Details about each library and setup screen can be found in “Job Recognition Libraries” on page 41, “Processing Libraries” on page 43, and “Imaging Libraries” on page 79, respectively.

Windows

1 **Go to Options|Docusave Server Setup.**
The Server Setup screen opens.



2 **Select *Activate Input Job Queue Processing at Docusave Server Startup* to auto-start processing the input queue when you start the server.**

By default, when Docusave Server is initially started, the processing of the queue is suspended so you can always view the current jobs in the queue. When this option is selected, Docusave Server starts processing the input queue when the program is started.

3 **Select a Job Recognition Library from the *Job Recognition Library Name* drop down list.**

The Job Recognition Library identifies the job type when the job is pulled from the input queue. For detailed information about Job Recognition Libraries, see "Job Recognition Libraries" on page 41.

4 **Select and configure the Processing Library that will be used for your jobs from the *Processing Library Name* drop down list.**

For detailed information about Processing Libraries and to determine which libraries you need to configure, see "Processing Libraries" on page 43.

5 **Select an Imaging System Library from the *Imaging System Library Name* drop down list.**

The Imaging Libraries are used to store and retrieve documents to and from an imaging system. For detailed information about Imaging System Libraries, see "Imaging Libraries" on page 79.

- Enter the Resource Set.

  The Resource Set is the name of a collection of printer resources stored within the archive system.

  The **MARS/NT** Resource Set is the Volume Group at the MARS/NT Storage Server where the printer resources are stored. The MARS/NT volumes where printer resources are stored cannot contain leading spaces.

The **Documanage** Resource Set is the Documanage Cabinet where the printer resources are stored. Every cabinet designated as a Resource Set must be setup as described in “Setting up Documanage for Printer Resources” on page 88.

Documents in a repository often require certain printer resources for proper document display. The types of these resources vary by the kind of document being displayed (AFP, Metacode, DCD, and so on). When a document is loaded into its repository, Docusave Server can make sure all needed resources are also present in the repository. Docuview LFS and Docupresentment use these resources to properly display the document when it is viewed.

Organizations sometimes use different versions of the same printer resources (with the same names) loaded on different printers at the same time. They control how the documents look by making sure that the print job is routed to the correct printer. These resources may differ in resolution (200 dpi vs. 600 dpi) or in the actual character sets. Each collection of resources assigned to a given printer is called a “Resource Set.” Docusave Server can process groups of jobs against different Resource Sets which are concurrently stored in a single Documanage system or other repository.

A Multiple Concurrent Resource Set Cache can also be built using the Resource Set option of Docusave Server. Resource Sets apply to resources in the Docuview Resource Cache.

The **FileNet** Resource Set is the Document Class in FileNet where the printer resources are stored.

- Set the Message Level.

The value you enter determines the severity of the messages in the log file. The severity ranges from the most severe (Level One) to the least severe (Level Nine).

For example, you can set the Message Level to 1 and you'll receive messages with a severity of 1. If you set the Message Level to 9, you'll receive messages with a severity between 1-9.

For more information see "Message Level and Message Log File" on page 38.

6 **Select the *Archived Document Compression Library* to turn document compression on and enable the compression library to be updated.**

The default compression library is CDECZLIB but can be changed to any Skywire Software supported library. Select *None* to turn compression off.

NOTE: The same compression library used by Docusave Server to compress files must be used by Documanage, Docupresentment or Docuview LFS for decompressing the files for viewing.

7 **Enter the *Log* and *Temp Directories*.**

- The Log Directory holds the log file of all Docusave Server processing steps.
- The Temp Directory holds any temporary files generated by Docusave Server.

8 **Enter the *DRR* and *Index Archive Configuration Files* path.**

- The DRR file (DRR.CFG) contains the DRR configuration information.
- The Index file (INDEX.CFG) contains the index configuration information.
  See "Using DRR/Index Configuration Files" on page 32 for more information.

9 **Click *OK* to save your selections or *Cancel* to exit without saving.**

UNIX

Field	Explanation
LibJobType	The Job Recognition Library. The Job Recognition Library identifies the job type when the job is pulled from the input queue. For detailed information about Job Recognition Libraries, see "Job Recognition Libraries" on page 41. Default = `./recognition/dsjobtyp.so`
LibImgSys	The Imaging System Library. The Imaging Libraries are used to store and retrieve documents to and from an imaging system. For detailed information about Imaging System Libraries, see "Imaging Libraries" on page 79. Default = `./imaging/dsisysd.so`
ResourceSet	The Resource Set is the name of a collection of printer resources stored within the archive system. See "Enter the Resource Set." on page 26. Default = `null value`
ResourceSetEmbedded	Embeds a comment record into the AFP/Metacode documents with the given value if a value is provided. It should match the value for `ResourceSet` Default = `null value`

Field	Explanation
MsgLevel	The Message Level. The value you enter determines the severity of the messages in the log file. The value you enter determines the severity of the messages in the log file. The severity ranges from the most severe (Level One) to the least severe (Level Nine). Default = `3`
CompressArchives	Turns document compression on and enables the compression library to be updated. Default = `NO` Inputs = `Yes` or `No`
LibComp	The compression library used for compressing documents before archival. Default = `cdeczlib`
LogDir	The Log Directory holds the log file of all Docusave Server processing steps. Default = `null value`
TempDir	The Temp Directory holds any temporary files generated by Docusave Server. Default = `./`
DRRconfig	The fully qualified path to the DRR configuration (DRR.CFG) file. Default = `drr.cfg` See “Using DRR/Index Configuration Files” on page 32 for more information.

Field	Explanation
INDEXconfig	The fully qualified path to the index configuration (INDEX.CFG) file. Default = `index.cfg` See “Using DRR/Index Configuration Files” on page 32 for more information.

Using DRR/Index Configuration Files

DRP Records and **DRR Records** contain a single string that identifies the document that is to be retrieved as well as other information about the document (which is described in the chart). The **DRR Configuration File** tells Docusave Server how to break that single string of characters into the multiple fields of information contained there.

	Field Offset	Length	Type
%%%DocID%%%	0	6	C
%%%GroupName%%%	6	7	C
%%%DTN%%%	13	5	N
%%%PageSpec%%%	18	4	C

NOTE: A DRP record string only needs to contain the %%%DocID%%% field.

The first column of numbers contains the offset from the beginning of the string, expressed in bytes. The second column is the length of the field, and the third column tells Docusave Server if the character contents of the field should be converted to a numeric value.

Parameter	Description
DocID	The identification of that image
GroupName	Represents the form name that Docusave Server will use to identify the image when it's placed into the EDL
DTN	Should be the number you want to be assigned to the image in the EDL
PageSpec	The range of pages you want to retrieve for this job

The **Index Configuration File** tells Docusave Server how to interpret the single string of index information preceding an AFP/DCD/PCL/Metacode document.

Sample Documanage Index Configuration File

Here is a sample Documanage index configuration file. Update your configuration file to reflect your site's Documanage identifiers.

The first column of numbers contains the offset from the beginning of the string, expressed in bytes. The second column is the length of the field, and the third column can only be set to "C" for Documanage.

	Field Offset	Length	Type
%%%Cabinet%%%	0	8	C
%%%Class%%%	8	8	C
%%%Title%%%	16	10	C
Au_Id	26	8	C
Au_Lname	34	10	C
Au_Fname	44	10	C
Phone	54	11	C
Address	65	32	C
City	97	16	C
State	113	2	C
Zip	115	5	C

The following are Special Properties in the `index.cfg` file related to Documanage:

%%%`Class`%%% is the category
%%%`Cabinet`%%% is the cabinet name (case-sensitive)

`%%%Title%%%` is the document name

The imaging system-dependent index information embedded in the print stream is interpreted using the `index.cfg` file. The `index.cfg` file maps the input index information to the Documanage fields, which come from the user's database tables. The names in the `index.cfg` file should exactly match the Documanage Folder Property names. Keep in mind that some databases may require names to be case-sensitive. If the Folder Property name contains blanks, it should be entered with an underscore in the `index.cfg` file. For example, if the Folder Property field name is "Field Name," enter "Field_Name" in the `index.cfg` file.

NOTE: The Folder Insert Authority to the table on which the Cabinet (listed above) is built must be granted via the Documanage Administrator.

You can also set Document Level Properties. These properties are optional.

%%%Author%%%	32
%%%AddedFrom%%%	8
%%%Date%%%	19
%%%SubType%%%	15
%%%Status%%%	10
%%%Keyword1%%%	15
%%%Keyword2%%%	15
%%%UserFlag1%%%	15
%%%UserFlag2%%%	15
%%%Description%%%	240

NOTE: Do not add the optional %%%AddedOn%%% Document Level field to the Documanage index configuration file. It is automatically set by the Documanage Server.

Sample MARS/NT Index Configuration File

The following is a sample MARS/NT index configuration file. Update your configuration file to reflect your site's MARS/NT identifiers.

	Field Offset	Length	Type
%%%Class%%%	0	4	C
%%%Title%%%	8	13	C
Keys	21	13	C
Client	34	8	C
Vendor	42	8	C
Subject	50	8	C

The first column of numbers contains the offset from the beginning of the string, expressed in bytes. The second column is the length of the field, and the third column tells Docusave Server if the character contents of the field should be converted to a numeric value.

The imaging system-dependent index information embedded in the print stream is interpreted using the `index.cfg` file. The `index.cfg` file maps the input index information to the MARS/NT fields, which come from the user's Custom Resource File (CRF). The names in the `index.cfg` file should match exactly with the MARS/NT Title field names. If the Title field name contains blanks, it should be entered with an underscore in the `index.cfg` file. For example, if the Title field name is "Field Name," enter "Field_Name" in the `index.cfg` file.

The following are special required fields in the `index.cfg` file related to MARS/NT:

`%%%Class%%%` is the MARS/NT volume group
`%%%Title%%%` is the document name

Sample FileNet Index Configuration File

The following is a sample FileNet index configuration file. Update your configuration file to reflect your site's FileNet identifiers.

	Field Offset	Length	Type
%%%Class%%%	0	12	C
Application_Number	12	6	C
Document_Type	18	6	C
Policy Number	24	8	C

The first column of numbers contains the offset from the beginning of the string, expressed in bytes. The second column is the length of the field, and the third column tells Docusave Server if the character contents of the field should be converted to a numeric value.

The imaging system-dependent index information embedded in the print stream is interpreted using the index.cfg file. The index.cfg file maps the input index information to the FileNet index fields, which come from the user's database definitions. The names in the index.cfg file should match exactly with the FileNet index field names. If the FileNet field name contains blanks, it should be entered with an underscore in the index.cfg file. For example, if the FileNet index field name is "Field Name," enter "Field_Name" in the index.cfg file.

There following are Special Properties in the index.cfg file related to FileNet:

%%%Class%%% is the document class

Sample VLAM EDL Library Index Configuration File

The following is a sample VLAM EDL Library index configuration file. Update your configuration file to reflect your site's VLAM Library identifiers.

	Field Offset	Length	Type
VLAM_Chain	0	4	C
VLAM_Description	4	36	C
VLAM_DTN	40	5	C
VLAM_Effdate	45	8	C
VLAM_Library	53	8	C
VLAM_Member	61	32	C
VLAM_Revision	93	5	C

The imaging system-dependent index information embedded in the print stream is interpreted using the index.cfg file. The index.cfg file maps the input index information to the mainframe VLAM Library fields.

Message Level and Message Log File

As Docusave Server services jobs it reports messages pertaining to the status of the processing to a log file. Each imaging library determines the name of its own log file. These log files are named Disk.log, Documanage.log, FileNet.log, IMCO.log, imgsysv.log, MARS.log, and OutputFile.log.

The templates for the messages written to the log files are read from an external file. These message templates are read from a file called msg.txt.

The example file provided below contains a sample of the available system messages, which you can customize to your needs. This message file includes the message number, the severity level of the message, and the message text. The log file that Docusave Server creates collects all system messages reported during processing.

1000,9,%s %s

1001,1,RC %d from stat() on '%s'

1002,1,Document ID created: %d (%d pages)

1003,1,Index string '%s' (class '%s') did not parse properly

1004,1,Unable to begin batch in class '%s'

1005,5,Class '%s' index values follow:

1006,5,Index field '%s' is undefined in the index configuration file

1007,5,Value for '%s' is '%s'

1008,5,Value for '%s' is %d

1009,9,Resizing bitmap buffer from %d to %d bytes

1010,1,Begin Document: '%s'

1011,3,Job Begins

1012,3,Job Ends, %d document(s) in %d batch(es) committed

1013,9,Imaging system ping (%s) %s

The first column contains the message number and the second column contains the severity level. The severity ranges from the most severe (Level One) to the least severe (Level Nine). The third column contains the message text.

When you are setting the message level, you can set the types of messages you want Docusave Server to send to the log file, based on the severity level of the message itself. The profile contains a line called **MsgLevel**=, and the value you enter determines the severity of the messages you want to be in your log file. For example, you can set the **MsgLevel=1** and you will receive severity Level One messages in your log report. If you set the message level to 9, you'll receive severity level 1-9 messages in your log report. The Message Level is set on the Docusave Server Setup screen.

Job Recognition Libraries

Introduction

The Job Recognition Library identifies the job type when the job is pulled from the input queue. The library returns a 3-character string that uniquely identifies the contents of the job, which Docusave Server uses as a suffix to construct the name of the job processing libraries.

Depending on the job type coming in to the system, the job recognition library selects the appropriate job and document processing library to process the job. For more information about Job Processing Libraries see “Processing Libraries” on page 43.

The following profile entry is the path to the library:

```
LibJobType
```

DSJOBTYP Library

DSJOBTYP recognizes all job types:

- AFP and Metacode Print Streams
- DMFPPP Log Files
- Docucorp Compound Documents (DCDs)
- Document RePrint Records (DRPs)
- Document Retrieval Requests (DRRs)
- ImageCreate Output
- Printer Control Language (PCL)
- Printer Resources

Processing Libraries

Introduction

The job recognition library identifies the type of job in the queue, and determines the job processing and document processing libraries to be used. The mechanism for automatically selecting the correct processing libraries is a *library naming convention*.

The library naming convention constructs processing library names from the job and document types (determined by the recognition step). If a pre-, post-, or processing library is present, and its name matches the naming convention for the current job, it will be invoked automatically at the appropriate time.

Job and document processing actions are implemented through separate libraries to allow a high degree of flexibility in controlling how processing occurs, and to provide custom behavior by using substitute or optional libraries.

Job processing refers to operations that happen once per job, where a job is a single entry in the input queue.

Document processing refers to operations that happen for each document in a job – potentially many times during one job. (A job typically contains many documents.)

Job processing always uses at least one library. The job processing library, may in turn, use a document processing library (for a total of

2 libraries). However, both job and document processing can also include optional pre-processing and post-processing libraries to bring the total number of processing libraries to 6. (Often pre- and post- processing libraries are not needed, and may not even be installed.)

Not surprisingly, the pre-processor library runs immediately before the corresponding job or document processor, and the post-processor runs immediately after.

The only purpose for using pre- and post- processing libraries is to modify (or supplement) the behavior of the main processing library. This makes it possible to use the standard processing libraries in non-standard situations.

For example, if you wish to validate index data in a database lookup before the standard document processor converts and files it, you might write a document pre-process library to validate the index data.

If you wanted to modify the result code assigned under a certain error condition, you might write a document post-processor library to test for that condition and modify the result code.

A custom job post-processor library could be used to provide additional notification (e.g., send an email or advance a workflow) when a job completes.

Pre- and Post- processing libraries can be simple or complex, and are entirely optional.

In addition to pre- and post- processing, custom processing libraries can be used to create new functions and operations (e.g., convert print images to a different output type such as PDF). Custom processing libraries may or may not use pre- or post- processing libraries as well.

Library Configuration

Depending on the type of input you have, you will need to configure the correct Job Processing and Document Processing Libraries. The table below identifies which libraries need to be configured, depending on the type of input.

If your input type is:	Set up these libraries:		Notes
	Job Processing	Document Processing	
DCD	DSPJBDCD	DSPDCDCD	No configuration required
ImageCreate Output	DSPJBIMC		No configuration required
AFP and library suffix is set to PRT	DSPJBPRT	DSPDCPRT	For Print Stream Document Archiving
AFP and library suffix is set to RAS		DSPDCRAS	For Print Stream Document Rasterization
DRP	DSPJBDRP		
DRR	DSPJBDRR		
DMFPPP	DSPJBFPP		
PCL	DSPJBPCL	DSPDCPCL	
Metacode and library suffix is set to PRT	DSPJBPRT	DSPDCPRT	For Print Stream Document Archiving
Metacode and library suffix is set to RAS		DSPDCRAS	For Print Stream Document Rasterization
Printer Resources	DSPJBRSC		For printer resource archiving.

DSPJBDCD Library

This job processing library processes DCD documents and breaks the input file into individual DCD documents. It then calls the DSPDCDCD document processing library to archive each document. The archived DCD includes the DCD document content and any (optional) one data set (one-transaction) Docuflex Data File (DDF) which immediately follows the DCD document in the input file. A one data set (one-transaction) DDF is the data used by Docuflex composition to create the DCD document.

A message is written to the Docusave Server log which lists the size in bytes of the DCD document and trailing data. The document data starts with the <DCD …> tag and ends with the</DCD> tag. The size reported as document data includes these tags. The trailing data is between the </DCD> and the next <DCD …> tags. The size reported as trailing data does not include these tags.

No configuration is required.

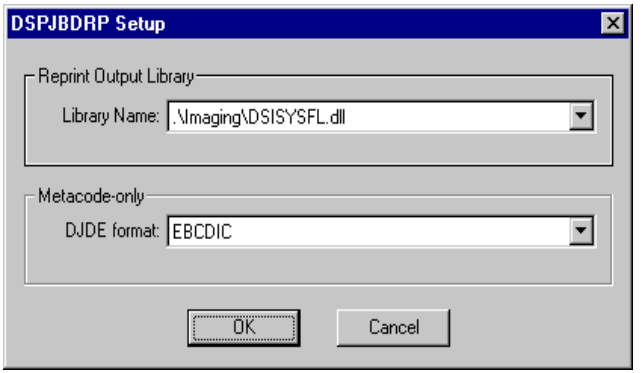
DSPJBDRP Library

This job processing library processes DRP (Document RePrint) records. A DRP record specifies a compressed AFP or Metacode print stream in the document management system which Docusave Server retrieves and bundles with its dependant printer resources for subsequent printing.

A DRP record contains a single string that identifies the compressed AFP or Metacode print steam that is to be retrieved. The DRR Configuration File tells Docusave Server how to break that single string of characters into the multiple fields of information contained there. See “Using DRR/Index Configuration Files” in the Docusave Server Installation Guide for more information.

Windows

The output of this library called a “reprint package” is a print stream which has the printer resources embedded. The reprint package can be printed to an IBM or Xerox printer using a third-party application.



- The Reprint Output Library Name is the Imaging System Library where print streams constructed for reprinting are sent. The Output File Imaging System (DSISYSFL) is specifically designed for use with the DSPJBDRP job processing library.

- The DJDE format tells Docusave Server to encode the initial DJDE packet in either EBCDIC or ASCII. The printer setup will dictate which one is appropriate for each customer site.

UNIX

Field	Explanation
DRPoutputLibrary	The Reprint Output Library Name is the Imaging System Library where print streams constructed for reprinting are sent. The Output File Imaging System (DSISYSFL) is specifically designed for use with the DSPJBDRP job processing library. Default = dsisysfl
DRPdjdeFormat	The DJDE format tells Docusave Server to encode the initial DJDE packet in either EBCDIC or ASCII. The printer setup will dictate which one is appropriate for each customer site. Default = EBCDIC

Printing Reprint Packages

A reprint package for an compressed AFP print stream includes an inline resource group which can contain any number of printer resources. Those printer resources will be used only for the duration of printing the current print stream.

A reprint package for an compressed metacode print stream contains the printer resources appropriate for the document, followed by the document

itself and then the most current version of the printer resources in the document management system. The last version contained in the reprint package is the version that will be used in subsequent print jobs.

If the most recent version of a printer resource is the correct one for the document it will not be included in the reprint package.

Printer Resources

When using the DSPJBDRP job processing library, it is critical that the document management system contain the most recent version of all printer resources on the printer.

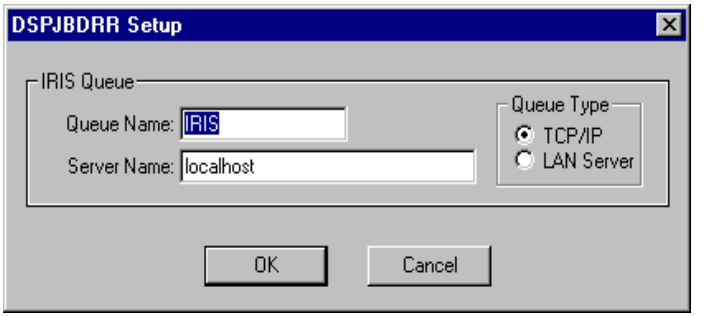
To make sure the document management system contains the most recent versions, always archive printer resources using Docusave Server whenever new or modified resources are installed on printers. This makes sure the document management system accurately reflects the state of the printers.

Using the DSPJBDRP job processing library when printer resources are not current in the document management system can cause incorrect resources to be installed on the printer, which in turn could cause incorrect output or failure to print.

DSPJBDRR Library

This job processing library processes DRRs (Document Retrieval Requests). DRRs specify TIFF documents in the archive which Docusave Server retrieves and bundles into Megafiles for subsequent Docucreate IC (IRIS) processing. If there are back to back DRRs that reference the same group name, Docusave Server concatenates them into the same form.

Windows



1. **Enter the name of the queue from which IRIS information is routed in the *IRIS Queue Name* text box.**

2. **Enter the server name where the IRIS queue resides in the *IRIS Server Name* text box.**

   You can enter “localhost” for the Server Name if the IRIS queue is on the same machine as Docusave Server.

3. **Select either *TCP/IP* or *LAN Server* in the *IRIS Queue Type* text box.**

   This specifies how you communicate with the Queue daemon. TCP/IP uses a TCP/IP protocol. LAN Server uses “named pipes.”

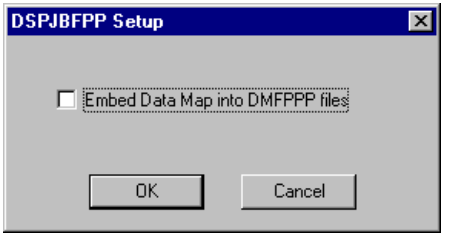
UNIX

Field	Explanation
IrisQueueName	The name of the queue from which IRIS information is routed. Default = `IRIS`
IrisQueueType	How you communicate with the Queue daemon. Default = `TCPIP`
IrisServerName	The server name where the IRIS queue resides. Default = `localhost`

DSPJBFPP Library

This job processing library processes DMFPPP log files. DMFPPP log files are generated by Documaker FP and record the forms and variable data used to generate the final print stream. Other Skywire Software products use the DMFPPP logs to generate XML.

Windows



Select whether or not you want to *Embed Data Map into DMFPPP files.* A Data Map embeds the index.cfg information into the documents as specially formatted comment records.

This setting is independent of the data map setting for AFP and Metacode.

UNIX

Field	Explanation
FPPIncludeDataMap	Embeds the index.cfg information into the documents as specially formatted comment records. This setting is independent of the data map setting for AFP and Metacode. Default = NO Inputs = YES or NO

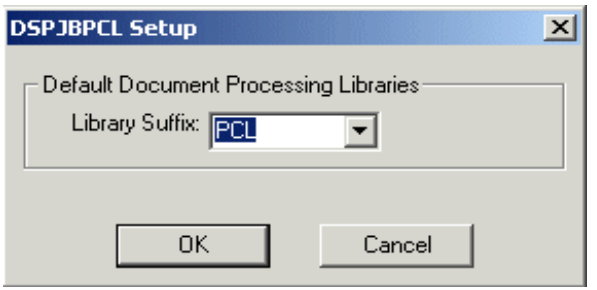
DSPJBIMC Library

This job processing library processes the output of ImageCreate (TIFF images, for example). Originally, ImageCreate had no direct connection to the imaging system and simply wrote its output to a queue for subsequent processing by AccessCommander. This library supplies support for backward-compatibility. No additional configuration is required.

DSPJBPCL Library

This job processing library processes PCL jobs. In this release it assumes that each job contains exactly one document, but the job and document processing has been separated into two processing libraries to allow for multiple documents in future jobs. In this release the document processor DSPDCPCL should be used by setting the Library Suffix = “PCL” on the DSPJBPCL Setup dialog.

Windows

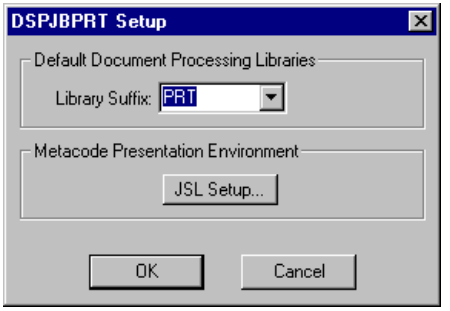


NOTE: All document processing libraries in your system display in the Library Suffix drop down list.

DSPJBPRT Library

This job processing library processes AFP and Metacode print streams. It breaks the print stream into individual documents and calls document processing libraries to archive them. The *Library Suffix* (see the setup dialog below) determines which document processing libraries are used. The document processing libraries with the “PRT” suffix will archive the documents in their native format, and the libraries with the “RAS” suffix will rasterize the documents into an image format before archiving them.

Windows



NOTE: All document processing libraries in your system display in the Library Suffix drop down list.

UNIX

Field	Explanation
DefaultPRTsuffix	The document processing libraries with the “PRT” suffix will archive the documents in their native format, and the libraries with the “RAS” suffix will rasterize the documents into an image format before archiving them. Default = PRT Inputs = PRT or RAS

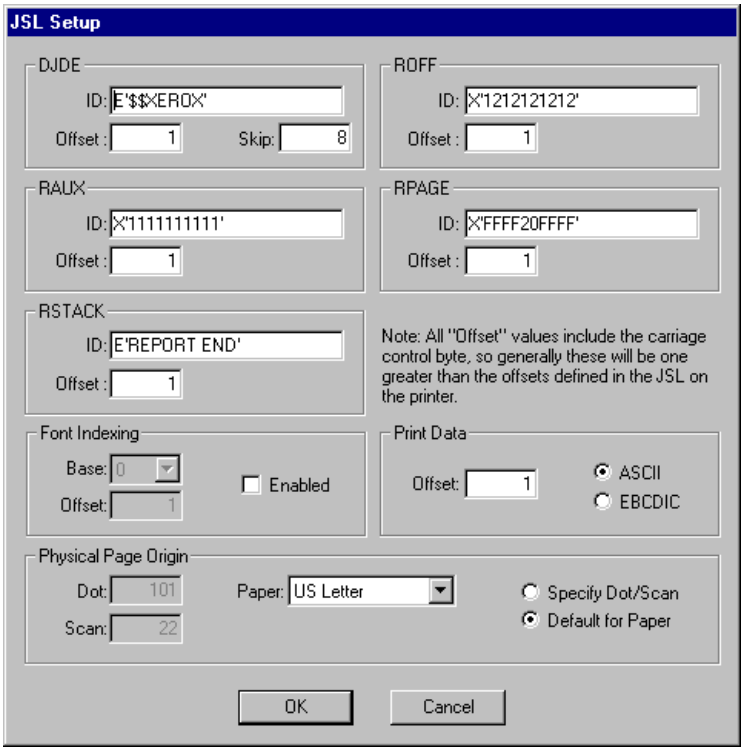
Metacode Presentation Environment (JSL Setup)

Docusave Server can process Metacode print streams as input, but needs additional setup information to replicate the print environment on the Xerox printer. Most of this setup information simply allows Docusave Server to distinguish common print records (containing text and formatting codes) from the special records containing directives to the printer (like DJDE, RSTACK, RPAGE, ROFF, and RAUX records).

The information required for this dialog can be gleaned from the JSL (Job Source Library) on the printer which would normally print the input documents. Differences between the JSL specification and this dialog are noted below by each parameter description when appropriate.

In addition, Xerox explains these JSL statements in the Dynamic Job Descriptor Entries chapter of the *Xerox Laser Printing Systems PDL/ DJDE Referenc*e.

Windows



NOTE: You should specify the value of each control card identifier in one of three values — (E)BCDIC, (A)SCII, or He(X)adecimal. The value you enter should be enclosed in apostrophes and preceded by the letter representing the data format. For example, E'REPORT END' is the EBCDIC value of the RStack ID in the screen above.

1 **Enter the appropriate DJDE information.**

- Enter the *DJDE ID*. This is the character string that uniquely identifies the DJDE record. Docusave Server interprets the skip value

to be the number of bytes from the beginning of the DJDE identifier where we expect to the find the DJDE parameters.

- Enter *Offset* value. This is the integer value representing the offset from the beginning of the string expressed in bytes. **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer.
- Enter the *Skip* value. This is the integer representing the starting position of DJDEs (for example, font references) in the Metacode print stream.

2 **Enter the ROFF information.**

- Enter the *ID* character string that activates page offsets for collating document sections in the printer output bin.
- Enter the *Offset* value. This is the integer value representing the offset from the beginning of the string expressed in bytes. **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer.

3 **Enter the RAUX information.**

- Enter the *ID* character string that indicates which auxiliary paper tray to use.
- Enter the *Offset* value. This is the integer value representing the offset from the beginning of the string expressed in bytes. **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer.

4 **Enter the RPAGE information.**

- Enter the *ID* character string used to request that a new image begins duplex printing on a right-hand page.
- Enter the *Offset* value. This is the integer value representing the offset from the beginning of the string expressed in bytes. **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer.

5 **Enter the RSTACK information.**

- Enter the *ID* character string that indicates the end of the print report.
- Enter the *Offset* value. This is the integer value representing the offset from the beginning of the string expressed in bytes. **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer.

6 **Enter the Font Indexing information.**

Font indexing notifies Docusave Server which column in the print stream contains the index values for the font information.

- *Enable* Font Indexing if necessary.
- Select the *Base* value.
  If you select 0, the first font in the font list is “Font0”
  If you select 1, the first font in the font list is “Font1”
- Enter the *Offset* value (from the beginning of each record) where the font index byte is located (e.g., entering a value of 2 skips two columns and proceeds to the third column). **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer.

7 **Enter Print Data information.**

This notifies Docusave Server where the print data begins.

- Enter the *Offset* value to the beginning of the print data (e.g., entering a value of 2 directs Docusave Server to begin at the first column, skip two columns, and proceed to the third column). **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer.
- Select whether the print data is ASCII or EBCDIC.

8 **Enter the Physical Page Origin information.**

- Select whether your will *Specify Dot/Scan* or *Default for Paper.*

- Enter the *Dot* origin for your paper stock if you selected *Dot/Scan*. This applies to non-standard paper size.
- Enter the *Scan* origin for your paper stock if you selected *Dot/Scan*. This applies to non-standard paper size.
- Select the *Paper* size (US Letter, US Legal, A4) you are using if you selected *Default for Paper.*

UNIX

Field	Explanation
DJDE	The character string that uniquely identifies the DJDE record. Default = *E'$$XER0X'*
DJDEoffset	The integer value representing the offset from the beginning of the string expressed in bytes. **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer. Default = 1
DJDEskip	The integer representing the starting position of DJDEs (for example, font references) in the Metacode print stream. Default = 8
ROFF	The *ID* character string that activates page offsets for collating document sections in the printer output bin. Default = *X'1212121212121212'*
ROFFoffset	The integer value representing the offset from the beginning of the string expressed in bytes. **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer. Default = 1

Field	Explanation
RAUX	The *ID* character string that indicates which auxiliary paper tray to use. Default = *X'1111111111111111'*
RAUXoffset	The integer value representing the offset from the beginning of the string expressed in bytes. **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer. Default = 1
RPAGE	The *ID* character string used to request that a new image begins duplex printing on a right-hand page. Default = *X'01FFFF20FFFF'*
RPAGEoffset	The integer value representing the offset from the beginning of the string expressed in bytes. **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer. Default = 1
RSTACK	The *ID* character string that indicates the end of the print report. Default = *E'REPORT END'*
RSTACKoffset	The integer value representing the offset from the beginning of the string expressed in bytes. **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer. Default = 1

Field	Explanation
FontIndexUsed	Notifies Docusave Server which column in the print stream contains the index values for the font information. Default = `NO` Inputs = `YES` or `NO`
FontIndexBase	0, the first font in the font list is “Font0” 1, the first font in the font list is “Font1” Default = `0`
FontIndexOffset	The *Offset* value (from the beginning of each record) where the font index byte is located (e.g., entering a value of 2 skips two columns and proceeds to the third column). **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer. Default = `1`
DataOffset	The Offset value to the beginning of the print data (e.g., entering a value of 2 directs Docusave Server to begin at the first column, skip two columns, and proceed to the third column). **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer. Default = `1`
CharacterEncoding	The print data is ASCII or EBCDIC. Default = `ASCII` Inputs = `ASCII` or `EBCDIC`
XeroxMinDot	The Dot origin for your paper stock. This applies to non-standard paper size. Default = `99`

Field	Explanation
XeroxMinScan	The Scan origin for your paper stock. This applies to non-standard paper size. Default = 23

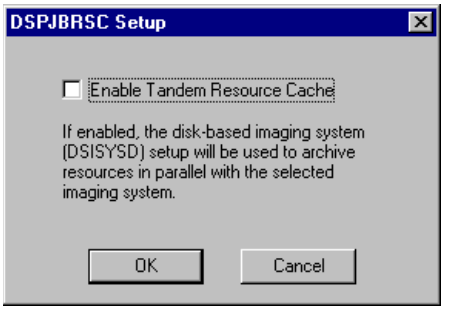
DSPJBRSC Library

Use this job processing library to archive printer resources. You must archive the printer resources before you can archive print streams.

This library processes printer resources which have been submitted to the input queue using the *addres* utility. (For more information about the Addres Utility see “Appendix D: Utility and Batch File” on page 141.) AFP and Metacode print streams cannot be archived using the DSPDCPRT processing library until all dependent printer resources have been archived first. This library emits the resources either to a selected imaging system, or simultaneously to both the imaging system and to a disk cache used by Docuview LFS (see below).

By selecting *Enable Tandem Resource Cache* you can archive printer resources to both the selected imaging system and to disk. Printer Resources are then archived to disk in a private structure which allows multiple versions of printer resource to coexist and be automatically accessed by Docuview LFS with the Docuview Resource Cache option enabled. Printer resources need to be archived only once.

Windows



UNIX

Field	Explanation
TandemResourceCache	Archives printer resources to both the selected imaging system and to disk. Default = NO Inputs = YES or NO

DSPRJPRT Library

This job pre-processing library is intended to help verify the setup for Xerox metacode print stream jobs. This library scans metacode print streams for DJDEs according to the current Docusave Server setup and returns an error if at least one DJDE record is not detected or if no fonts are referenced on any DJDE record. If no DJDE records are found, an additional pass through the print stream is made to suggest appropriate DJDE settings.

Other than the DJDE settings for the normal print stream job processor, no profile settings specific to this job pre-processor are used.

DSPDCDCD Library

This document processing library archives each DCD document. No configuration is required.

DSPDCMRG Library

This document processing library is called by DSPJBPRT to transfer normalized AFP and Metacode Documerge forms to a mainframe VLAM library. This library performs no compression or resource checking, but it removes the leading Docusave Server index comment records before the forms are written to the VLAM library.

DSPDCNUL Library

This document processing library is called by DSPJBPRT to transfer AFP/ Metacode documents to a mainframe VLAM Library “as-is”, without any modification. This library does no processing, compression or resource checking. No configuration is required.

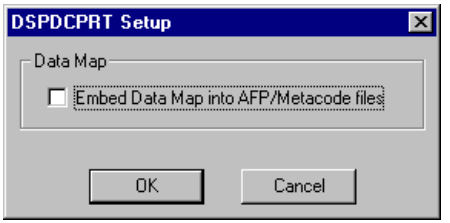
DSPDCPCL Library

This document processing library is called by DSPJBPCL to archive each PCL document. The archived PCL will be identical to the PCL in the input job; no modifications to the input PCL are made. No configuration is required.

DSPDCPRT Library

This document processing library is called by DSPJBPRT to archive each AFP/Metacode document.

Windows



Select whether or not you want to *Embed Data Map into AFP/Metacode files.* A data map embeds the index.cfg information into the documents as specially formatted comment records.

If a print stream file contains index information and an associated data map, Docuview LFS can present this information in the *Document Index Information* screen.

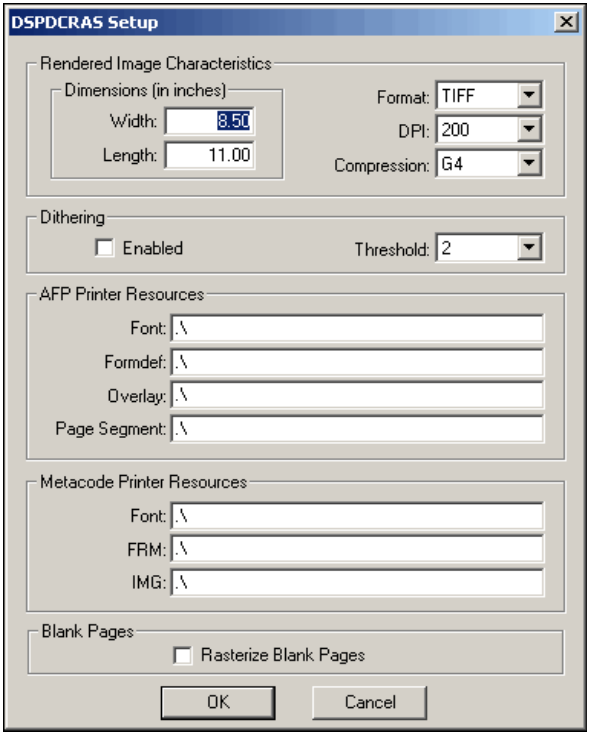
UNIX

Field	Explanation
IncludeDataMap	Embeds the index.cfg information into the documents as specially formatted comment records. Default = `NO` Inputs = `YES` or `NO`

DSPDCRAS Library

This document processing library is called by DSPJBPRT to rasterize and archive each AFP/Metacode document.

Windows



1 **Enter the *Rendered Image Characteristics*.**
The Rendered Image Characteristics describe the general format of each image generated by this library. These common characteristics include the width and height of the images (in inches), the DPI, the compression format used, and the image format itself. Several image formats can be

generated, but a format should be chosen that is compatible with the viewer that will be used to display the images.

- Enter the image *Width* and *Length* in inches.
- Select the image *Format* from the drop down list. The available choices are FNBI, MO:DCA, MP_TIFF, TIFF.
- Select the image *DPI.* This specifies the image resolution (dots per inch density). Valid choices are 300, 240, 200, 150, 120, 100.
- Select the *Compression.* This is the compression algorithm options for the output image. Valid choices are G4 (CCITT Group 4 bit compression) G3 (CCITT Group 3 bit compression) and No (uncompressed images).

2 **Enable *Dithering* if you want Docusave Server to apply a dither pattern to the image (s) it creates. Select the threshold from the drop down list, if necessary.**

Dithering affects the quality of pixel images during scaling. Lower dither numbers result in darker, thicker characters. Higher numbers result in lighter, thinner characters.

3 **Enter the *AFP Printer Resources* location.**

Set the paths for the designated folders where a single copy of the AFP Printer Resources are located. An example of the format expected by this library is:

D:\AFPFONTS\C0UA108.382
C0UA110.382
C0UA112.382

NOTE: All printer resources of a particular type (e.g., all AFP page segments) must reside in the same directory, although more than one type may reside in a single directory.

- **Font.** Set the path for the designated folder where a single copy of the AFP Font Printer Resources is located.
- **Formdef.** Set the path for the designated folder where a single copy of the Form Definitions is located.
- **Overlay.** Set the path for the designated folder where a single copy of the AFP Overlay Printer Resources is located.
- **Page Segment.** Set the path for the designated folder where a single copy of the AFP PSEG Printer Resources is located.

4 **Enter the *Metacode Printer Resources* location.**

Set the paths for the designated folders where a single copy of the Metacode Printer Resources are located. An example of the format expected by this library is:

D:\XERFONTS\UA108E.FNT
UA110E.FNT
UA112E.FNT

NOTE: All printer resources of a particular type (e.g., all Metacode IMGs) must reside in the same directory, although more than one type may reside in a single directory.

- **Font.** Set the path for the designated folder where a single copy of the Metacode Font Printer Resources is located.
- FRM. Set the path for the designated folder where a single copy of the Metacode FRMs is located.
- IMG. Set the path for the designated folder where a single copy of the Metacode IMGs is located.

5 **Select *Rasterize Blank Pages* if you want to include blank pages in the rasterized output.**

UNIX

Field	Explanation
PageWide	The image Width in inches. Default = `8.50`
PageHigh	The image Length in inches. Default = `11.00`
TargetFormat	The image Format. Default = `TIFF` Inputs = `FNBI, MO:DCA, MP_TIFF, TIFF`
DPI_OUT	Specifies the image resolution (dots per inch density). Default = `200` Inputs = `300, 240, 200, 150, 120, 100`
COMPRESS	The compression algorithm options for the output image. Valid choices are G4 (CCITT Group 4 bit compression) G3 (CCITT Group 3 bit compression) and No (uncompressed images). Default = `G4` Inputs = `G4, G3, NONE`
Dither	Sets dither option when scaling. Default = `NO` Inputs = `YES` or `NO`

Field	Explanation
DitherThreshold	Applies a dither pattern to the images it creates. Dithering affects the quality of pixel images during scaling. Lower dither numbers result in darker, thicker characters. Higher numbers result in lighter, thinner characters Default = 2
AfontDir	UNIX: The path for the designated directory where a single copy of the AFP Printer Resources is located. Sample value = `./prtResources/` **Note:** You must include a forward slash (/) at the end of the directory entry. Default = Null String
FormdefDir	UNIX: The path for the designated directory where a single copy of the Form Definitions is located. Sample value = ./prtResources/ **Note**: You must include a forward slash (/) at the end of the directory entry. Default = Null String
OverlayDir	UNIX: The path for the designated directory where a single copy of the AFP Overlay Printer Resources is located. Sample value = ./prtResources/ **Note**: You must include a forward slash (/) at the end of the directory entry. Default = Null String

Field	Explanation
PsegDir	UNIX: The path for the designated directory where a single copy of the AFP PSEG Printer Resources is located. Sample value = ./prtResources/ **Note**: You must include a forward slash (/) at the end of the directory entry. Default = Null String
ShowBlanks	Controls whether or not blank pages are included in the rasterized output. Default = NO
MfontDir	UNIX: The path for the designated directory where a single copy of the Metacode Printer Resources is located. Sample value = ./prtResources/ **Note**: You must include a forward slash (/) at the end of the directory entry. Default = Null String
FrmDir	UNIX: The path for the designated directory where a single copy of the Metacode FRMs is located. Sample value = ./prtResources/ **Note**: You must include a forward slash (/) at the end of the directory entry. Default = Null String

Field	Explanation
ImgDir	UNIX: The path for the designated directory where a single copy of the Metacode IMGs is located. Sample value = ./prtResources/ **Note**: You must include a forward slash (/) at the end of the directory entry. Default = Null String

Imaging Libraries

Introduction

Imaging libraries are used to store and retrieve documents to and from an imaging system. Each imaging system has its own library.

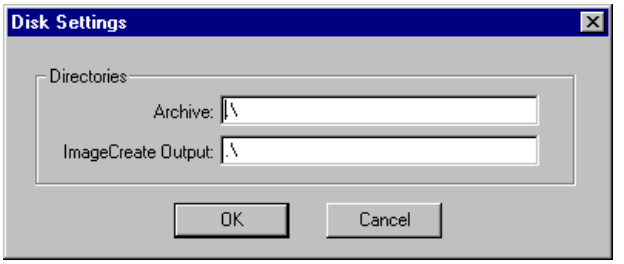
The following profile entry is the path to the library:

```
LibImgSys
```

DSISYSD Library

This library, used for disk storage, is primarily for debugging and creating a resource cache for Docuview LFS.

Windows



Enter the drive letter and path name of the directory that will hold your Docusave Server printer resources and print streams in the *Archive* text box.

Enter the drive letter and path name of the directory that will hold the ImageCreate jobs in the *ImageCreate Output* text box.

As Docusave Server services your imaging jobs, it reports system messages pertaining to the status of the processing to a log file named Disk.log.

UNIX

Field	Explanation
ArchiveDir	The path name of the directory that will hold your Docusave Server printer resources and print streams. Default = `null value`

Field	Explanation
OutDir	The path name of the directory that will hold the ImageCreate jobs. Default = `null value (defaults to TempDir which defaults to null value)`

DSISYSDM Library

This library works with Documanage as its archive component. When using Documanage as the archiving component, you must set up Documanage to use printer resources.

At start up, dsisysdm tries to establish a connection to the Documanage Server as specified in the configuration file. If it is unable to establish the connection, it continues to retry until it establishes the connection.

Once dsisysdm establishes the connection to the Documanage Server, it pings the connection on a regular basis when it is idle. If a ping determines that the connection has failed, dsisysdm attempts to re-establish it. Again, reconnection attempts continue until the connection is re-established.

If the connection to the Documanage Server fails while the Docusave Server is processing a job, the job fails and the failed connection keeps Docusave Server's rollback mechanism from undoing the job. Like any other failed job, Docusave Server puts the interrupted job in the error queue. After dsisysdm finishes putting the failed job in the error queue, it tries to re-establish a connection to Documanage.

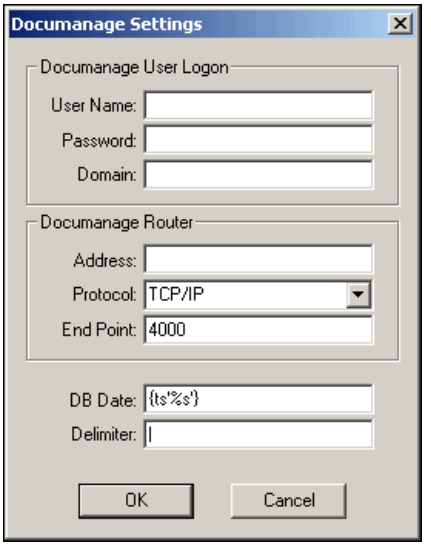
NOTE: While dsisysdm is not connected to Documanage, Docusave Server does not accept jobs from the input queue.

WARNING: You must be running Documanage v6.4 Service Release 3 or greater to use the DSISYSDM library.

Documanage Settings

In UNIX, you specify the Documanage settings in the Docusave.ini file. For information about the fields in this configuration file, see "Docusave.ini File" on page 86.

In Windows, you can specify the Documanage settings in the Docusave.ini file or you can use the Documanage Settings screen to provide most of the information specified in that configuration file.



NOTE: To specify whether you want to include blanks in Documanage folder queries, you must use the QueryWithBlankFields field in the configuration file. See "Docusave.ini File" on page 86 for more information.

Make the following entries on the Documanage Settings screen:

Field	Description
User Name	Enter the Documanage user name.
Password	Enter the Documanage user password.
Domain	Enter the Documanage Domain. The domain should be the Windows workgroup of which the Documanage user is a part.
Address	Enter the IP address or the name for the Documanage router. You can enter any name your network can resolve in the *Address* box. On a Windows network, this may include a NETBIOS computer name and/or a standard Domain Name Services (DNS) name. Ask your network administrator what name resolution services are available on your network.
Protocol	Select the default protocol used by Documanage from the list. The default is TCP/IP. Other options include NetBIOS over TCP, NetBIOS over IPX, NETBEUI, Pipes, and SPX.
End Point	Enter the TCP/IP port number used to reach Documanage. Before you change the port number from the default, talk with your network administrator.

DB Date	Enter a format string to describe how to format date and time data to match your database system. The format expected by a database system can vary according to setup options and even machine settings (the “Regional Options” control panel, for example). By modifying the DB Date text you can transform date and time data to a format recognized by the database. For more information about date formats, see “Date Format” on page 89.
Delimiter	Enter a character that Docusave Server will use to retrieve the document information from the DRR record. In Documanage when you specify a document to retrieve on a DRR record, you specify a document by providing a description of the cabinet, folder, and document in the DocID. Those three descriptions are delimited by the delimiter character entered in the *Delimiter* field. For example, if you use “|” as your delimiter character, when you specify a document to retrieve on a DRR record, you identify the cabinet, folder, and document, separating each value by the delimiter: Authors|au_id=’5763’|DocId=1473.

The dsisysdm imaging system library logs additional messages about its communication with Documange in the “Documanage.log” file.

Docusave.ini File

This table shows the fields in the Docusave.ini file you use to specify the Documanage settings:

Field	Explanation
DMgLogonID	The Documanage user ID. Default = EZPOWER
DMgLogonPW	The Documanage user password. Default = EZPOWER
DMgDomain	The Documanage Domain. The domain should be the Windows workgroup of which the Documanage user is part. Default = DOCUCORP
DMgRouterIP	The TCP/IP address for the Documanage router. Default = 127.0.0.1
DMgRouterProtocol	The protocol used by Documanage. Default = ncacn_ip_tcp
DMgRouterEndpoint	The TCP/IP port number used to reach Documanage. Default = 4000
DateFormat	A format string which describes how to format date and time data to match your database system. The format expected by a database system can vary according to setup options and even machine settings (e.g., the “Regional Options” control panel). By modifying the DB Date text you can transform date and time data to a format recognized by the database. Default = {ts'%s'}

Field	Explanation
QueryWithBlankFields Note: You can only set this field via the Docusave.ini file.	Enter Yes to allow blanks in Documanage folder queries. The INI value is in the Docusave section of the Docusave.ini file. If a value in the index record is blank and that field is one of the keys to a compound key to the database, include it in folder queries. For example, suppose you have a Documanage folder and DB table set up as a compound key. If one of the keys is blank from the index record, you should include it in the folder query. And if these index records are found: Document Key1 Key2 ======== ===== ===== 1 12345 ABC 2 45678 DEF 3 12345 If you set this field to Yes, document 3 is included in folder 12345 and document 1 is added to folder 12345ABC. If this field is set to No, document 3 is included with document 1 in folder 12345ABC. Default = No.
URLdelimiter	A delimiter character that Docusave Server uses to retrieve document information from the DRR record. In Documanage, when you specify a document to retrieve on a DRR record, you specify a document by providing a description of the cabinet, folder, and document in the DocID. Those three descriptions are delimited by the delimiter character. Default = |

Setting up Documanage for Printer Resources

WARNING: The following steps must be completed for Docusave Server to work with Documanage.

Follow the installation procedures provided by the documentation included with Documanage. In addition, check that the following installation considerations specific to using Docusave Server with Documanage have been performed:

- A Documanage cabinet called “Printer Resources” must exist (notice the embedded blank) for the exclusive use of storing printer resources. For convenience, the Label Formula should be the ResourceDescr value (see below for details on ResourceDescr value).

- The table on which this cabinet is built must contain at least two indexes: ResourceType (a single character which is the primary key), and ResourceDescr (32 byte character field). The Documanage Folder Property names must match these index names exactly.

  The Folder Insert Authority to this table must be granted via the Documanage Administrator.

- Name the table “RSRC” to allow for future compatibility.

- A Documanage document type of “RSRC” with Labels Keyword 1 = ResourceKey must be created via the Documanage Administrator.

- Seven folders should be created in the Printer Resources cabinet with these values for ResourceType and ResourceDescr:

ResourceType	ResourceDescr
‘A’	“AFP Fonts”
‘D’	“AFP FORMDEFs”
‘O’	“AFP Overlays”
‘P’	“AFP Page Segments”
‘M’	“Metacode Fonts”
‘F’	“Metacode FRMs”
‘I’	“Metacode IMGs”

Date Format

All supported databases accept the ODBC standard format. Additionally, “database native” formats and “database conversion functions” may be available, depending on what database is used.

The table below provides a few examples to illustrate how the DB Date can be used:

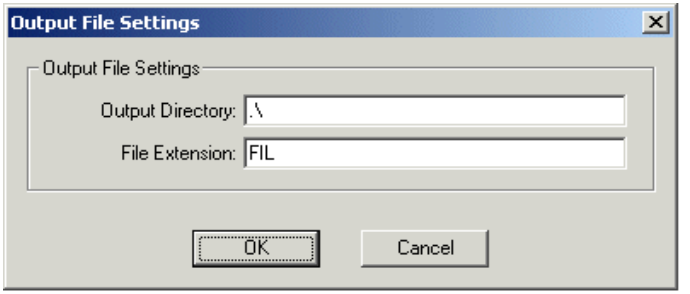
NOTE: The contents of the DB Date field are interpreted by Docusave Server, and possibly by the ODBC database driver, and then passed to the database wherever an incoming field of type date appears. The incoming data string for the field replaces the special characters *‘%s’* wherever they appear in the DB Date string. All other characters are passed literally as they appear in the DB Date string.

Sample Input Data	Database Format	DB Date	Comments
2001-04-12	yyyy-mm-dd hh:mi:ss	{ts'%s'}	ODBC standard date
2001-04-12 10:24:32	yyyy-mm-dd hh:mi:ss	{ts'%s'}	ODBC standard date and time
08/10/01 10:24:32	Database native	'%s'	Data in database native format, varies with database settings
'08/14/01'	Database native	%s	Database native, with single quotes in input
23/10/01 15:32:19	DB Date function	TO_DATE ('%s', 'DD/MM/YY HH24:MI:SS')	Database conversion function (example is Oracle)

For more information about Documanage, please consult your Documanage product documentation.

DSISYSFL Library

This library writes the output generated by the DSPJBDRP job processing library to a file in the specified output directory. The output files called “reprint packages” are named with a random 10-digit number and the specified extension. A reprint package is a print stream which has the printer resources embedded. The reprint package can be printed to an IBM or Xerox printer using a third party application. The file extension is provided in case the third party application only looks for files with a particular extension.



As Docusave Server services your imaging jobs, it reports system messages pertaining to the status of the processing to a log file named OutputFile.log.

UNIX

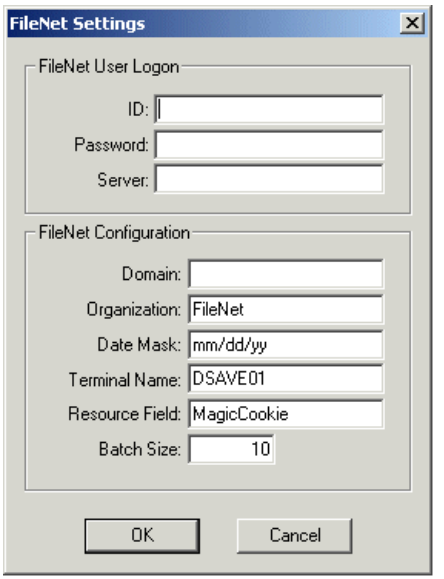
Field	Explanation
OutputFileDir	The path name of the directory that holds the output reprint packages. Default = ./
OutputFileExt	The file extension for the reprint packages. Default=FIL

DSISYSFN Library

This library is designed to function with FileNet as its archive component.

WARNING: For Docusave Server/NT you must install Panagon Image Services Toolkit (Windows Server) release number 3.6 to use the DSISYSFN library.

Windows



- Enter the FileNet User ID in the *ID* field.
- Enter the FileNet Password in the *Password* field.
- Enter the name of the machine running the FileNet Image Management System (IMS) Server to which Docusave Server will logon in the *Server* field.

- Enter the Domain name that has been established in the FileNet IMS in the *Domain* field. This field is case sensitive.
- Enter the Organization that has been established in the FileNet IMS in the *Organization* field. This field is case sensitive.
- Enter the format of the mask Docusave Server will use to convert character strings to FileNet Date fields in the *Date Mask* field. The default is “mm/dd/yy” and reverts to a FileNet default in case of an invalid mask.
- Enter the Terminal Name in the *Terminal Name* field. The terminal name is a unique string that identifies the Docusave Server session to FileNet and differentiates it from all other applications connected to FileNet.
- Enter the Resource Field in the *Resource Field* field. The resource field is the name of the index field (defined as integer) where Docusave Server places a “magic cookie” whenever it stores a printer resource. This allows Docusave Server Sever to distinguish between printer resources and other non-resources that may reside in the document class reserved for printer resources. **NOTE:** The resource field is used exclusively with print stream archival. If you are rasterizing print streams, leave this field blank.
- Enter the maximum number of documents you want Docusave Server to place into a single batch in the *Batch Size* field. For example, a value of 10 means you want to include 10 documents per batch.

As Docusave Server services your imaging jobs, it reports system messages pertaining to the status of the processing to a log file named FileNet.log.

NOTE: The format for the DocID field of a DRR record with FileNet is the numeric document identifier defined by FileNet. For example, 1234567890.

The following entries can be added to your profile to enable and disable Fast Batch Committal and TIFF Recoding.

Field	Explanation
FNfastBatchCommittal	An alternate method of archiving documents into FileNet. Caution: This not a safe way to archive documents since these functions do not perform validation testing of the document contents or their index information. Default = NO
FNfastBatchAllocation	Optionally specifies the number of bytes of memory to allocate to each batch of documents during Fast Batch Committal. Default = 5120000
RecodeTIFFs	Controls use of the FileNet FFI_image_decode and FFI_image_encode on TIFF files returned by DRR jobs. By default, Docusave Server makes calls to decode TIFF images retrieved via Document Retrieval Request Processing and calls to re-encode them using the FileNet WAL Library. Recoding verifies the image is formed correctly when is retrieved from FileNet; for example, it makes sure the bitmap width is a proper multiple of pixels. Errors can result from this process which will cause the DRR to fail if the TIFF images were originally imported into FileNet by a non-conforming application. This option lets you disable the automatic TIFF recoding and continue to retrieve the non-conforming images. Set this option to “N” if you encounter a message such as the following written to the FileNet log: “(01021): Non-zero WAL return code from FFI_image_encode: <203,0,1104> SC: ucb line width not a multiple of 8." Alerted by the original error, customers should then determine that the retrieved images are still usable for their purposes. Default = Y

AIX

Field	Explanation
FNLogonID	The FileNet User ID. Default = `SysAdmin`
FNLogonPW	The FileNet Password. Default = `null value`
FNserver	The FileNet Server to which Docusave Server will logon. Default = `FNServer`
FNimsDomain	The Domain name established in the FileNet Image Management System (IMS). Default = `ISI`
FNimsOrganization	The Organization established in the FileNet IMS. Default = `FileNet`
FNdateMask	The format of the mask Docusave Server uses to convert character strings to FileNet Date fields. The default is “mm/dd/yy” and reverts to a FileNet default in case of an invalid mask. Default = `mm/dd/yy`
FNterminalName	The Terminal Name. The terminal name is a unique string that identifies the Docusave Server session to FileNet and differentiates it from all other applications connected to FileNet. Default = `IMGACC01`

Field	Explanation
FNresourceField	The resource field is the name of the index field where Docusave Server places a “magic cookie” whenever it stores a printer resource. This allows Docusave Server Sever to distinguish between printer resources and other non-resources that may reside in the document class reserved for printer resources. Default = null value
FNbatchSize	The maximum number of documents you want Docusave Server to place into a single batch in the *Batch Size* field. For example, a value of 10 means you want to include 10 documents per batch. Default = `10`

Setting up FileNet for Printer Resources

WARNING: The following steps must be completed for Docusave Server to work with FileNet.

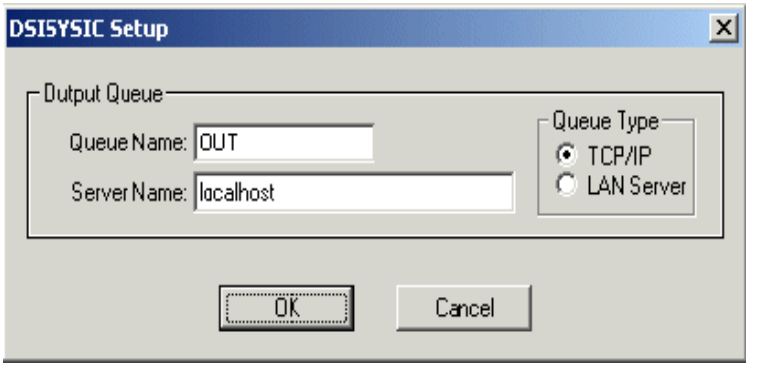
Follow the installation procedures provided by the documentation included with FileNet. In addition, check that the following installation considerations specific to using Docusave Server with FileNet have been performed:

- A FileNet class called “ISIresources” must exist for the exclusive use of storing printer resources.
- Within this class, an index field called “ResourceID” must be defined as a “string” field with a minimum length of 64 bytes.

DSISYSIC Library

This library writes the output generated by Docusave Server to a queue in a format consistent with the format of ImageCreate output. (This job can later be read using the IMCO functions in DAPI.)

The documents output to this imaging library must be rasterized images generated by the document processor DSPCDRAS. In this scenario, the Library Suffix must be set to “RAS” on the DSPJBPRT setup dialog.



6 **Enter the name of the queue from which the output is routed in the *Output Queue Name* text box.**

The output queue is the queue to which DSISYSIC will write its output for subsequent processing by another application using the IMCO DAPI functions.

7 **Enter the server name where the output queue resides in the *Output Server Name* text box.**

You can enter “localhost” for the Server Name if the output queue is on the same machine as Docusave Server.

8 **Select either TCP/IP or LAN Server in the *Output Queue Type* box.**

This specifies how you communicate with the Queue daemon. TCP/IP uses a TCP/IP protocol. LAN Server uses “named pipes.”

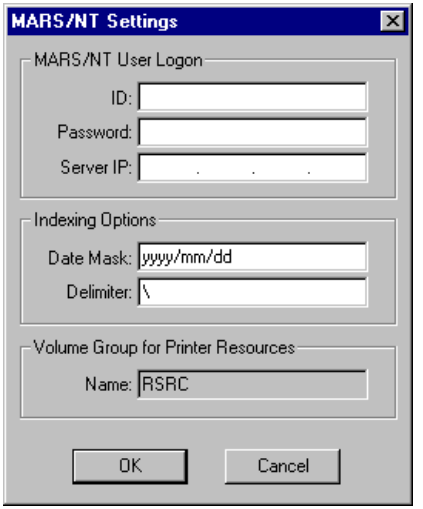
As Docusave Server services your imaging jobs, it reports system messages pertaining to the status of the processing to a log file named IMCO.log.

AIX

Field	Explanation
ImcoQueueName	The name of the output queue to which DSISYSIC will write its output for subsequent processing by another application using the IMCO DAPI functions. Default = OUT
ImcoQueueType	How you communicate with the Queue daemon. Default = TCPIP
ImcoServerName	The server name where the output queue resides. Default = localhost

DSISYSM Library

This library is designed to function with MARS/NT as its archive component.



- Enter the user ID necessary to log in to MARS/NT in the *ID* field.
- Enter the user password necessary to log in to MARS/NT in the *Password* field.
- Enter the *Server IP.* The server IP enforces a valid IP address to identify the MARS/NT Server.
- Enter the format of the mask Docusave Server will use to convert character strings to MARS/NT DATE fields in the *Date Mask* field. The default is “yyyy/mm/dd” and reverts to today’s date in case of an invalid mask.

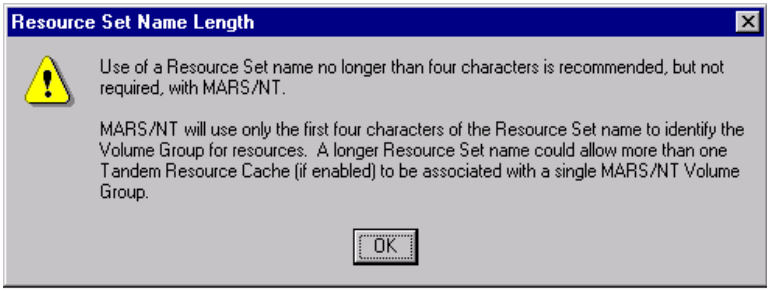
- Enter a character that Docusave Server will use to separate the different values of an index field in the data base in the *Delimiter* field.

  For example, if your delimiter character is "|" and you are populating a multi-value field that contains colors, the value passed in the index string might look like this: Red|White|Blue.

NOTE: The format for the DocID field of a DRR record with MARS/NT is IRN (Image Reference Number). For example, C#1234567890L#1234567890R#00001P#00001.

- The Volume Group for Printer Resources Name field displays the first four characters of the Resource Set field (from the main setup screen). This field is read only.

  If the resource set name is longer than four characters, you see this message:



As Docusave Server services your imaging jobs, it reports system messages pertaining to the status of the processing to a log file named MARS.log.

For more information about the MARS/NT system, please consult your MARS/NT product documentation.

DSISYSU Library

This library echoes the call sequence to the log, which can be used as the skeleton code for a custom implementation of this library. This library does not archive anything. It is not platform specific.

DSISYSV Library

This library is designed to function with a mainframe VLAM Library as its archive component. This library is provided specifically to allow Docusave Server to upload AFP and Metacode documents, without modification, to a mainframe VLAM Library. In this scenario the document processor DSPDCNUL should be used by setting the Library Suffix = “NUL” on the DSPJBPRT Setup dialog.

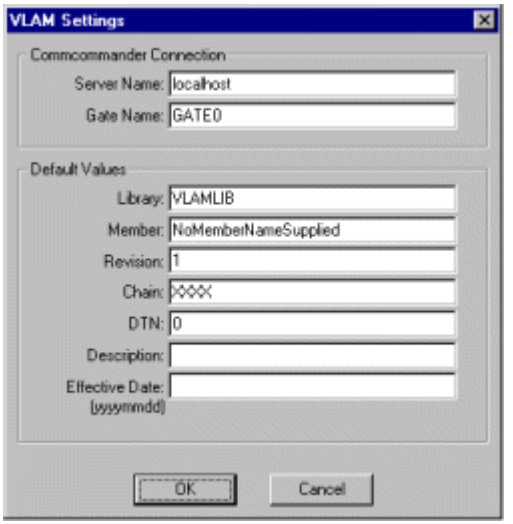
Software prerequisites

Commcommander 2.0.2 and Virtual Library Access Method (VLAM) 3.1 or higher are a prerequisites for this library.

On Windows you must add the Commcommander installation directory to the Windows System Path to make it known to Docusave Server.

On AIX you must copy the Commcommander components named ccmcoma.so, ccmcscpa.so, ccmcsska.so and isimsgwa.so from your Commcommander installation to the directory named “docusave server” in your Docusave Server installation.

Windows



- Enter the logical name of the LAN server upon which the Commcommander modules have been installed in the *Server Name* field.

- Enter the logical name associated with the Commcommander Server running on the server machine in the *Gate Name* field. The name labels a software gateway through which your PC can communicate with the Commcommander components on the mainframe. The Gate name is assigned to Commcommander Server when it is executed on the NT server machine.

The contents of a VLAM library is organized into members. The combination of member name and revision level uniquely identifies each member. To make life a bit easier, each member has a description field.

A Document Type Number (DTN) is used to associate the member's data with rules for printing and document assembly. Documerge uses Effective Date

fields as selection criteria when choosing among dated versions of a given member.

The other parts of a VLAM library member are called chains. Chains are the data components.

- Enter the name of the VLAM Library in the *Library* field.
- Enter the Member Name in the *Member* field. Do not use Member Names which contain an @ or $ symbol.
- Enter the Revision Level in the *Revision* field.
- Enter the Chain Name in the *Chain* field.
- Enter the Document Type Number (DTN) in the *DTN* field.
- Enter the Description in the *Description* field.
- Enter the Effective Date in yyyymmdd format in the *Effective Date* field.

You can enter values in the Default Values section of the VLAM Settings dialog or you can over ride theses values by adding the VLAM fields to your index configuration file.

The maximum sizes (imposed by VLAM) are:

VLAM_Chain	4	
VLAM_Description	36	
VLAM_DTN	5	(0-99999)
VLAM_Effdate	8	(yyyymmdd)

VLAM_Library	8	
VLAM_Member	32	
VLAM_Revision	5	(0-32767)

As Docusave Server services your imaging jobs, it reports system messages pertaining to the status of the processing to a log file named imgsysv.log.

Field	Explanation
VLAMdefaultLibrary	The name of the VLAM library. Default = VLAMLIB
VLAMdefaultMember	The Member Name. Do not use Member Names which contain an @ or $ symbol. Default = NoMemberNameSupplied
VLAMdefaultRevision	The revision level. Default = 1
VLAMdefaultChain	The Chain Name. Default = XXXX
VLAMdefaultDTN	The Document Type Number (DTN). Default = 0
VLAMdefaultDescription	The description. Default = “”
VLAMdefaultEffdate	The effective date in yyyymmdd format. Default = “”

DSISYSMQ Library

The WebSphere MQ Imaging Library (DSISYSMQ.dll) is used with Docusave Server to insert documents (files) into a WebSphere MQ (WSMQ) queue for processing by a WSMQ server application.

The library packages the document content along with its associated indexing metadata into a WSMQ message and places it on the configured queue. The format of the message is defined by one of the template files in the Docusave Server directory. The template is chosen based on information in each document. The template is an XML file with tags that correspond to the desired index fields. Thus, the messages placed in the WSMQ queue are XML documents, such as (Simple Object Access Protocol) SOAP requests. The binary TIFF file data is encoded with Base64 encoding to place the binary data in the appropriate tag in the XML document.

The input documents to the DSISYSMQ imaging library are expected to be multi-page TIFF files, as produced by the *P*rint-stream *D*o*C*ument *RAS*terization Processing Library (DSPDCRAS).

NOTE: DSPDCRAS requires printer resource configuration and the presence of formatted index information records as generated by Documaker or Documerge. Please refer to "DSPDCRAS Library" on page 72 for complete information on Docusave Server and the DSPDCRAS library.

For maximum flexibility, the XML message format is supplied from external template files which are copied and edited to create each submitted XML message. No XML processing library is used to create the messages within the DSISYSMQ library, so XML support for this purpose on the client is not required.

You must supply the document type name for each document being processed to Docusave as part of the document indexing data. The name is encoded in the USER.DATA indexing record and found in the reserved field name of "documentType." Here are some example document type names:

```
BankingCreditApplicationDocument
BankingMortgageDocument
BankingPersonalDocument
```

The DSISYSMQ library, for each incoming document, looks for a saved template file by the name:

```
tmpl<documentType>.xml
```

For example, *BankingPersonalDocument* resolves to a template file of t*mplBankingPersonalDocument.xml*. The library processes the template and expects to find an <ObjectContent> tag with a CDATA[] part to hold the base-64 data like this:

<objectContent><![CDATA[*data is placed here*]]>**</objectContent>**

DSISYSMQ encodes and inserts the TIFF data in the base-64 format at this tag. The base-64 data is broken into a multi-line CDATA XML part in lines of 76 base-64 characters, with the last line being 76 or fewer characters.

DSISYSMQ also scans the XML document for tags with names identical to fields specified in the INDEX.CFG file and sets the data values for those tags to the values in the USER.DATA record provided to Docusave in the original print stream document. Tags with names that do not appear in the INDEX.CFG file are unaffected. That is, the template file can supply full tag names and values that are not supplied in the INDEX.CFG data. These values are then constant for all documents using that template. If an INDEX.CFG field is not found as a tag name in a given template, it is ignored. It is not an error to supply extra indexing fields that are not in every template file.

Document attribute tags in the template can be in these forms:

```
<Namespace:Name></Namespace:Name>
<Namespace:Name />
<Namespace:Name xsi:nil="true"/>
```

In these examples, the *Name* portion of the tag is matched against the INDEX.CFG field list. If a non-empty value, such as *abc*, for a document attribute is given in the index data, its tag is modified and written in the output XML file as shown here:

```
<Namespace:Name>abc</Namespace:Name>
```

Otherwise it is left as it is in the template.

You can add document types and fields per document type by creating new template XML files and changing the existing template XML files and INDEX.CFG files respectively, in addition to changing the input process generating the print stream files.

Defining the WebSphere MQ Client Channel

DSISYSMQ.DLL connects to a WSMQ server via a defined client-connection channel. This channel can be defined different ways depending on the capabilities required. A basic client channel can be defined using the MQSERVER environment variable. The channel name must match a defined WSMQ Server Connection Channel. The DSISYSMQ library uses this channel if it is defined. In the Docusave environment, define:

MQSERVER=*serverChannel***/tcp/***WSMQhost***(PortNumber)**

Parameter	Explanation
serverChannel	The name of the server channel defined on the WSMQ server.

Parameter	Explanation
tcp	The supported communication transport protocol for DSISYSMQ.
WSMQhost	The host name or IP address of the WSMQ server computer.
PortNumber	(Optional) For TCP/IP, WebSphere MQ assumes the channel is connected to port 1414. If you have assigned another port number to TCP/IP for WebSphere MQ, you must supply that port number wrapped in parentheses with the MQSERVER environment variable.

To define a channel that uses advanced options, including SSL authentication and communication, you cannot use the MQSERVER definition method. To supply advanced channel definition options, you must create a client channel definition table (CCDT) file, make it accessible to Docusave, and configure the location and name of the file in environment variables:

MQCHLLIB=*directory_path*

MQCHLTAB=*file name*

Variable	Explanation
directory_path	The path to the directory that contains the CCDT file.
file name	The name of the CCDT file, by default AMQCLCHL.TAB.

You can create a client channel definition table on the WSMQ server using the WSMQ Explorer interface. This option lets you define multiple alternative channels to a single WSMQ queue manager. WSMQ client code will try each channel in turn until one connect or all fail. See the IBM WSMQ documentation for more information on the available configuration options, including configuring channels which use SSL authentication and encryption. If the table is created on the WSMQ server installed in the default location, the table file will be in the following directory:

*WSMQ-data-file-directory***IBM\WebSphere MQ\Qmgrs****queue-manager-name***@ipcc**

Configuring Docusave

Since the DSISYSMQ.DLL is expecting to release multi-page TIFF files only, set up Docusave Server as follows:

1. **Go to the Docusave Server Setup screen (Options | Docusave Server Setup).**
2. **Go to the Processing Library Name drop down list and select .\Processing\DSPJBPPRT.dll. Click the Setup button. Set the Library Suffix to RAS.**
3. **Go to the Processing Library Name drop down list and select .\Processing\DSPDCRAS.dll. Click the Setup button. Set the Output Format to MP_TIFF.**

As Docusave Server services your imaging jobs, it reports system messages pertaining to the status of the processing via a log file named *DSyyyymmdd.log*, where *yyyy* is the year, *mm* the month, and *dd* is the day the log file was created.

Refer to "Docusave Server Setup" on page 25 for more information on setting up Docusave Server.

Configuring the DSISYSMQ Library

Follow these steps to configure the DSYSYSMQ library:

1. **From the Docusave Server Setup screen (Options | Docusave Server Setup), select the .\Imaging\DSISYSMQ.DLL from the Imaging System Library Name drop down list.**

2. **Click the Setup button.**

3. **Enter the queue name into the Target Queue field. Enter the queue manager name into the Queue Manager field. Click Ok to save the values into the DOCUSAVE.INI file.**

You can also enter these values directly into the DOCUSAVE.INI file. Open your DOCUSAVE.INI file and look for a section named [MQ]. If there is no [MQ] section, scroll to the end of the file and add a section by that name with these entries:

```
[MQ]
QueueManager=<Queue Manager Name>
TargetQueue=<Queue Name>
```

Here is an example:

```
[MQ]
QueueManager=QM_Example
TargetQueue=default
```

This additional setting is available in the [MQ] section of the DOCUSAVE.INI file:

```
Debug=1
```

Only include this setting to create a log file, called IMGSYSMQ.LOG, that tracks activity of the DSISYSMQ library if recommended by Skywire Software Support.

Sample WSMQ Index Configuration File

The imaging system-dependent index information embedded in the print stream is interpreted using the index configuration (INDEX.CFG) file. The INDEX.CFG file maps the input index information to the destination system index fields, which you define.

Here is a sample INDEX.CFG file. Update your configuration file to reflect your site's index fields.

Field Name	Field Offset	Length	Type
formNumber	0	15	C
documentTitle	15	16	C
imageMethodCode	31	16	C
receivedDateTime	47	16	C
documentStatus	63	16	C
recordsManagementStatus	79	16	C
recordsManagementId	95	16	C
recordsManagementTitle	111	16	C
clientId	127	16	C
documentDateTime	143	16	C
captureCentreCode	159	16	C
externalReferenceNumber	175	16	C
imprintId	191	16	C
arrangementId	207	16	C
applicationId	223	16	C

Field Name	Field Offset	Length	Type
productId	239	16	C
computerCentreCode	255	16	C
documentType	271	40	C

The Field Offset column contains the offset from the beginning of the string, expressed in bytes. The Length column contains the length of the field, and the Type column must be C for DSISYSMQ indicating Character data interpretation of the field.

Sample XML template

Each document is placed in the CDATA section of a template XML file and the associated indexing data is merged into the surrounding XML tags.

Here is a section of a sample XML file showing tags available to hold the data. In this example, the tags are nested within the following tag:

```
AcmeBankingPersonalDocumentMetaData
```

This is not required. The DSISYSMQ parser will find the tags, if they are in the appropriate form, wherever they are in an XML document.

```
<objectContent><![CDATA[]></objectContent>

<AcmeBankingPersonalDocumentMetaData xsi:type="ns-
1449102670:AcmeBankingPersonalDocument" xmlns:ns-
1449102670="http://attributes.message.yjs0.acme.com">

   <ns-1449102670:formNumber></ns-1449102670:formNumber>

   <ns-1449102670:documentTitle></ns-1449102670:documentTitle>

   <ns-1449102670:imageMethodCode></ns-
1449102670:imageMethodCode>

   <ns-1449102670:receivedDateTime/>
```

```
   <ns-1449102670:documentStatus></ns-
1449102670:documentStatus>
   <ns-1449102670:recordsManagementStatus xsi:nil="true"/>
   <ns-1449102670:recordsManagementId xsi:nil="true"/>
   <ns-1449102670:recordsManagementTitle xsi:nil="true"/>
   <ns-1449102670:clientId></ns-1449102670:clientId>
   <ns-1449102670:documentDateTime></ns-
1449102670:documentDateTime>
   <ns-1449102670:captureCentreCode xsi:nil="true"/>
   <ns-1449102670:externalReferenceNumber xsi:nil="true"/>
   <ns-1449102670:imprintId xsi:nil="true"/>
   <ns-1449102670:arrangementId xsi:nil="true"/>
   <ns-1449102670:applicationId></ns-1449102670:applicationId>
   <ns-1449102670:productId></ns-1449102670:productId>
   <ns-1449102670:computerCentreCode></ns-
1449102670:computerCentreCode>
```

Appendix A: Docusave Server UNIX Profile Entries

This section lists the .docusave server profile entries, description, and sample values as they appear in the default profile shipped with Docusave Server on AIX, Linux and Solaris.

Field	Explanation	Sample Value
[Docusave Server]		
QueueName	The name of the queue from which the input is routed	Docusave Server
QueueType	How you communicate with the Queue daemon. TCP/IP uses a TCP/IP protocol.	TCPIP
ServerName	The server name where the input queue resides. Enter “localhost” for the Server Name if the input queue is on the same machine as Docusave Server.	localhost
JobBuffers	The maximum number of jobs that are concurrently read from the queue.	8
ErrorQueueName	The Error Queue is a storage destination for any jobs that couldn’t process normally. You can enter “localhost” for the Server Name if the error queue is on the same machine as Docusave Server.	Error
ErrorQueueType	The *Error Queue Type* box, select either *TCP/IP* or *LAN Server.* This specifies how you communicate with the Queue daemon. TCP/IP uses a TCP/IP protocol.	TCPIP
ErrorServerName	Server name where the error queue resides.	localhost

Field	Explanation	Sample Value
IrisQueueName	The name of the queue from which IRIS information is routed.	IRIS
IrisQueueType	The server name where the IRIS queue resides.	TCPIP
IrisServerName	How you communicate with the Queue daemon	localhost
LibJobType	The Job Recognition Library. The Job Recognition Library identifies the job type when the job is pulled from the input queue. For detailed information about Job Recognition Libraries, see “Job Recognition Libraries” on page 41.	./recognition/dsjobtyp.so
LibImgSys	The Imaging System Library. The Imaging Libraries are used to store and retrieve documents to and from an imaging system. For detailed information about Imaging System Libraries, see “Imaging Libraries” on page 79.	./imaging/dsisysu.so
DefaultPRTsuffix	The document processing libraries with the “PRT” suffix will archive the documents in their native format and the libraries with the “RAS” suffix will rasterize the documents into an image format before archiving them.	RAS
ResourceSet	The Resource Set is the name of a collection of printer resources stored within the archive system.	RSRC
ResourceSetEmbedded	Embeds a comment record into the AFP/Metacode documents with the given value if a value is provided. It should match the value for `ResourceSet`	RSRC
IncludeDataMap	A data map embeds the index.cfg information into the documents as specially formatted comment records.	NO
OutDir	The path name of the directory that will hold the Imagecreate jobs.	./
ArchiveDir	The path name of the directory that will hold your Docusave Server printer resources and print streams.	./

Field	Explanation	Sample Value
CompressArchives	Turns document compression on and enables the compression library to be updated.	YES
PageWide	The image Width in inches.	8.50
PageHigh	The image Length in inches.	11.00
Dither	Sets dither option when scaling.	YES
DitherThreshold	Applies a dither pattern to the image (s) it creates. Dithering affects the quality of pixel images during scaling. Lower dither numbers result in darker, thicker characters. Higher numbers result in lighter, thinner characters	2
TargetFormat	The image Format.	TIFF
DPI_OUT	Specifies the image resolution (dots per inch density).	200
COMPRESS	The compression algorithm options for the output image. Valid choices are G4 (CCITT Group 4 bit compression) G3 (CCITT Group 3 bit compression) and No (uncompressed images).	G4
MfontDir	The paths for the designated folders where a single copy of the Metacode Printer Resources is located. **Note:** You must include a forward slash (/) at the end of the directory entry.	./prtResources/
ImgDir	The path for the designated folder where a single copy of the Metacode IMGs is located. **Note:** You must include a forward slash (/) at the end of the directory entry.	./prtResources/
FrmDir	The path for the designated folder where a single copy of the Metacode FRMs is located. **Note:** You must include a forward slash (/) at the end of the directory entry.	./prtResources/

Field	Explanation	Sample Value
AfontDir	The path for the designated folders where a single copy of the AFP Printer Resources is located. **Note:** You must include a forward slash (/) at the end of the directory entry.	./prtResources/
PsegDir	The path for the designated folder where a single copy of the AFP PSEG Printer Resources is located. **Note:** You must include a forward slash (/) at the end of the directory entry.	./prtResources/
OverlayDir	The path for the designated folder where a single copy of the AFP Overlay Printer Resources is located. **Note:** You must include a forward slash (/) at the end of the directory entry.	./prtResources/
FormdefDir	The path for the designated folder where a single copy of the Form Definitions is located. **Note:** You must include a forward slash (/) at the end of the directory entry.	./prtResources/
XeroxDefaultJDL	Specifies the name of the JDL that will be used until another is located in the print stream.	XJDL
XeroxDefaultJDE	Specifies the name of the JDE that will be used until another is located in the print stream.	XJDE
XeroxDefaultPMODE	Specifies the default coordinate system for Xerox Graphics and Images. You can choose PORT or LAND. The default is LAND	PORT
XeroxMinDot	The Dot origin for your paper stock. This applies to non-standard paper size.	101
XeroxMinScan	The Scan origin for your paper stock. This applies to non-standard paper size.	22

Field	Explanation	Sample Value
DJDE	The character string that uniquely identifies the DJDE record.	E'$$XEROX'
DJDEoffset	The integer value representing the offset from the beginning of the string expressed in bytes. **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer.	1
DJDEskip	The integer representing the starting position of DJDEs (for example, font references) in the Metacode print stream.	8
RAUX	The *ID* character string that indicates which auxiliary paper tray to use.	X'1111111111'
RAUXoffset	The integer value representing the offset from the beginning of the string expressed in bytes. **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer.	1
ROFF	The *ID* character string that activates page offsets for collating document sections in the printer output bin	X'1212121212'
ROFFoffset	The integer value representing the offset from the beginning of the string expressed in bytes. **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer.	1
RPAGE	The *ID* character string used to request that a new image begins duplex printing on a right-hand page.	X'FFFF20FFFF'
RPAGEoffset	The integer value representing the offset from the beginning of the string expressed in bytes. **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer.	1

Field	Explanation	Sample Value
RSTACK	The *ID* character string that indicates the end of the print report.	E'REPORT END'
RSTACKoffset	The integer value representing the offset from the beginning of the string expressed in bytes. **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer.	1
DataOffset	The Offset value to the beginning of the print data (e.g., entering a value of 2 directs Docusave Server to begin at the first column, skip two columns, and proceed to the third column). **NOTE:** All offset values include the carriage control byte, so these will generally be one greater than the offsets defined in the JSL on the printer.	1
DRRconfig	The fully qualified path to the DRR configuration (DRR.CFG) file. See “Using DRR/Index Configuration Files” on page 32 for more information.	drr.cfg
INDEXconfig	The fully qualified path to the index configuration (INDEX.CFG) file. See “Using DRR/Index Configuration Files” on page 32 for more information.	index.cfg
MsgLevel	The Message Level. The value you enter determines the severity of the messages in the log file. The value you enter determines the severity of the messages in the log file. The severity ranges from the most severe (Level One) to the least severe (Level Nine).	9

Additional profile entries include:

Field	Explanation	Default Entry
[Docusave Server]		
LibComp	The fully qualified path of the compression library used for compressing documents before archival.	cdeczlib
LogDir	The Log Directory holds the log file of all Docusave Server processing steps.	Null value
TempDir	The Temp Directory holds any temporary files generated by Docusave Server.	./

Appendix A: Docusave Server UNIX Profile Entries

Appendix B: Print Stream Profile Entries

Introduction

Docusave Server and its libraries read the contents of the Docusave Server profile to determine many aspects of their behavior. This profile is maintained by various setup dialogs under Windows, or it can be manually edited on platforms with no graphical user interface.

The profile can be modified to fit the needs of each customer site. For example, if you are using Metacode print streams and certain character strings vary from the default profile, you can change the default profile to fit your needs.

Docusave Server Profile

NOTE: These profile entries are used by multiple Skywire Software products.

The Docusave Server profile recognizes the following settings and are listed with their default values. To the right of each entry are codes defining which applications use each entry.

- DS - Docusave Server
- DV - Docuview LFS

```
[Docusave Server]
TargetFormat=TIFF                    (DS)
DPI_OUT=0                          (DS,DV)
COMPRESS=G4                        (DS,DV)
LeftMargin=0                       (DS,DV)
TopMargin=0                        (DS,DV)
AfontDir=                          (DS,DV)
FrmDir=                            (DS,DV)
ImgDir=                            (DS,DV)
MfontDir=                          (DS,DV)
OverlayDir=                        (DS,DV)
VfontDir=                          (DS,DV)
FrmBuffers=3                       (DS,DV)
ImgBuffers=3                       (DS,DV)
```

```
OverlayBuffers=3                (DS,DV)
FormdefDir=                     (DS,DV)
DefaultFormdef=                 (DS,DV)
PsegDir=.\                      (DS,DV)
DropShading=NO                  (DS,DV)
PageWide=8.50                   (DS,DV)
PageHigh=11.00                  (DS,DV)
LPP=60                          (DS,DV)
DJDE=E'$$XEROX'                 (DS,DV)
DJDEoffset=1                    (DS,DV)
DJDEskip=8                      (DS,DV)
RAUX=X'1111111111111111'        (DS,DV)
RPAGE=X'01FFFF20FFFF'           (DS,DV)
ROFF=X'1212121212121212'        (DS,DV)
RSTACK=E'REPORT END'            (DS,DV)
RAUXoffset=1                    (DS,DV)
RPAGEoffset=1                   (DS,DV)
ROFFoffset=1                    (DS,DV)
RSTACKoffset=1                  (DS,DV)
BFORMdisabled=NO                (DS,DV)
FORMSdisabled=NO                (DS,DV)
SIDEdisabled=NO                 (DS,DV)
XeroxMinScan=23                 (DS,DV)
XeroxMinDot=99                  (DS,DV)
XeroxHighlight=252              (DS,DV)
```

```
XeroxDefaultJDL=DPLJDL          (DS,DV)
XeroxDefaultJDE=ONLINE          (DS,DV)
XeroxDefaultPMODE=LAND          (DS,DV)
CCOffset=0                      (DS,DV)
DataOffset=1                    (DS,DV)
LastDataOffset=32760            (DS,DV)
FontIndexOffset=1               (DS,DV)
FontIndexBits=4                 (DS,DV)
FontIndexUsed=NO                (DS,DV)
FontIndedBase=0                 (DS,DV)
CharacterEncoding=ASCII         (DS,DV)
ExternalCCtable=NO              (DS,DV)
IncludeProcessCard=YES             (DS)
INDEXconfig=index.cfg              (DS)
ResourceSet=                       (DS)

[JDL:DPLJDL - JDE:ONLINE]
PageWide=                       (DS,DV)
PageHigh=                       (DS,DV)
XeroxMinDot=                    (DS,DV)
XeroxMinScan=                   (DS,DV)
LeftMargin=                     (DS,DV)
TopMargin=                      (DS,DV)
FORMS=                          (DS,DV)
```

```
FORMAT=                          (DS,DV)
FontIndexBase=                   (DS,DV)
FontIndexBits=                   (DS,DV)

[PDE:PDE000]
FONTScount=0                     (DS,DV)
FONTS000=FNT000                  (DS,DV)
PMODE=                           (DS,DV)
BEGINcount=0                     (DS,DV)
BEGIN00=(0.18IN, 0.66IN)         (DS,DV)

[CC Table]
CCentry5F=0,0,0,0,0              (DS,DV)
```

Profile Sections

The main section of the profile is named [Docusave Server]. Entries in this section are the initial default values, but may be overridden by entries in other sections. In the Docusave Server profile, there can be blank lines between each section, but not between entries within the sections.

The [CC Table] section may contain up to 256 entries to define the behavior of all possible 256 carriage control bytes. All characters have a default definition which is appropriate for most users, but those default definitions can be overridden here.

Each entry is in the format:

CCentryXX=nba,nbe,naa,nae,p

where:

XX – are hex digits 00-FF

nba – is the number of print lines to advance ***before*** displaying the data

nbe – can be 0 or 1 (false or true) and indicates whether to advance to a new page ***before*** displaying the data

naa – is the number of print lines to advance ***after*** displaying the data

nae – can be 0 or 1 (false or true) and indicates whether to advance to a new page ***after*** displaying the data

p – can be 0 or 1 (false or true) and indicates whether the data should be displayed

The remaining sections apply only to Metacode.

JDL/JDE section

When a new JDL/JDE pair is referenced in a DJDE packet in a Metacode print stream, a section name is constructed in the following way:

[JDL:jdlname – JDE:jdename]

If this section exists, the entries contained in this section will interpreted and honored immediately. The following entries are recognized here:

- PageWide
- PageHigh
- XeroxMinDot
- XeroxMinScan
- LeftMargin
- TopMargin
- FORMS
- FORMAT
- FontIndexBase
- FontIndexBits

PDE section

A FORMAT DJDE parameter may specify a font list in RAD50 format or it may specify a PDE name. In the latter case a section name is constructed in the following way:

[PDE:pdename]

If this section exists, the entries contained in this section will be interpreted and honored immediately. The following entries are recognized here:

- FONTScount

- FONTS*nnn*
- PMODE
- BEGINcount
- BEGIN*nn*

Profile Description

The entry descriptions listed below are broken into three categories: entries that apply to Metacode and AFP, entries that apply to only Metacode, and entries that apply only to AFP.

Metacode and AFP Entries

TargetFormat – determines the type of images produced. Possible values:

- TIFF
- MP_TIFF(Multi-page TIFF)
- FNBI(FileNet Banded Image)
- MO:DCA(IBM Mixed Object: Document Content Architecture)

DPI_OUT – specifies the resolution (in dots per inch) of the rendered images. Possible values:

- For AFP:
  - 240
  - 200
  - 120
  - 100
- For Metacode:
  - 300
  - 200
  - 150
  - 100
- 0 implies 240 for AFP and 300 for Metacode

COMPRESS – determines the compression algorithm of the rendered images. Possible values:

- "G4" – CCITT Group-4 (2-dimensional) compression
- "G3" – CCITT Group-3 (1-dimensional) compression
- "NONE" – no compression

LeftMargin – shifts the entire page image this many pixels to the right

TopMargin – shifts the entire page image this many pixels downward

VfontDir - specifies the path where (obsolete) Vfonts are stored

PageWide – specifies the width, in inches, of the images produced

PageHigh – specifies the height, in inches, of the images produced

LPP – specifies the maximum number of lines rendered per page before automatically advancing to the next page

CCOffset – specifies the number of bytes from the beginning of each record where the carriage control byte is located.

DataOffset – specified the number of bytes from the beginning of each record where the first byte of print data is located.

LastDataOffset – specifies the number of bytes from the beginning of each record where the last byte of print data is located.

FontIndexOffset – specifies the number of bytes from the beginning of each record where the font index byte is located. This is only used if FontIndexUsed=YES.

FontIndexBits – specifies the number of low-order bits in the font index byte actually used to determine the font index value.

FontIndexUsed – “YES” or “NO”. “YES” indicates that a particular byte (see FontIndexOffset) of each record should be interpreted as an index into the current font list for displaying the print data on that record.

FontIndexBase – 0 or 1. Specifies the index value associated with the first font in the font list.

CharacterEncoding – “ASCII” or “EBCDIC”. Specifies the method used to interpret each code point in the print stream.

IncludeProcessCard – “YES” or “NO”. If “YES” a special header record is prepended to each document retrieved from the input print stream. This header record contains information that reflects the current user settings, most notably the DJDE settings, that help make the document self-describing.

INDEXconfig – the fully-qualified path of the index configuration file used to include data maps into the individual documents.

ResourceSet – the name of the Resource Set that will be embedded into the document as a special comment record.

Metacode Only Entries

NOTE: Docuview has its own method for locating printer resources. The following directories will be used to load printer resources as a last option only if Docuview is unable to locate them otherwise.

FrmDir – specifies the path where Xerox FRMs are stored

IMGDir – specifies the path where Xerox IMGs are stored

MfontDir – specifies the path where Xerox fonts are stored

FrmBuffers – specifies the maximum number of Xerox FRMs that will be buffered in memory

ImgBuffers – specifies the maximum number of Xerox IMGs that will be buffered in memory

DJDE – the character string used by Metacode printers to identify DJDE records. The value may be prefaced with the letter ‘E’, ‘A’, or ‘X’ to indicate that the string should be translated as EBCDIC, ASCII, or hexadecimal (respectively).

DJDEoffset – specifies the number of bytes from the beginning of the record at which the DJDE identifier will be located. *Note: this value includes the carriage control byte, so generally this value will be one greater than the offset defined in the JSL on the printer.*

DJDE skip – specifies the number of bytes from the beginning of the DJDE identifier to the first byte of DJDE parameters.

RAUX - the character string used by Metacode printers to identify RAUX records. The value may be prefaced with the letter ‘E’, ‘A’, or ‘X’ to indicate that the string should be translated as EBCDIC, ASCII, or hexadecimal (respectively).

RAUXoffset – specifies the number of bytes from the beginning of the record at which the RAUX identifier will be located. *Note: this value includes the carriage control byte, so generally this value will be one greater than the offset defined in the JSL on the printer.*

RPAGE - the character string used by Metacode printers to identify RPAGE records. The value may be prefaced with the letter ‘E’, ‘A’, or ‘X’ to indicate that the string should be translated as EBCDIC, ASCII, or hexadecimal (respectively).

RPAGEoffset – specifies the number of bytes from the beginning of the record at which the RPAGE identifier will be located. *Note: this value includes the carriage control byte, so generally this value will be one greater than the offset defined in the JSL on the printer.*

ROFF - the character string used by Metacode printers to identify ROFF records. The value may be prefaced with the letter 'E', 'A', or 'X' to indicate that the string should be translated as EBCDIC, ASCII, or hexadecimal (respectively).

ROFFoffset – specifies the number of bytes from the beginning of the record at which the ROFF identifier will be located. *Note: this value includes the carriage control byte, so generally this value will be one greater than the offset defined in the JSL on the printer.*

RSTACK - the character string used by Metacode printers to identify RSTACK records. The value may be prefaced with the letter 'E', 'A', or 'X' to indicate that the string should be translated as EBCDIC, ASCII, or hexadecimal (respectively).

RSTACKoffset – specifies the number of bytes from the beginning of the record at which the RSTACK identifier will be located. *Note: this value includes the carriage control byte, so generally this value will be one greater than the offset defined in the JSL on the printer.*

BFORMdisabled – valid values are "YES" and "NO". If "YES" then normal BFORM processing does not occur.

FORMSdisabled – valid values are "YES" and "NO". If "YES" then normal FORMS processing does not occur.

SIDEdisabled – valid values are "YES" and "NO". If "YES" then normal SIDE processing does not occur.

XeroxMinScan – specifies the scan (“X”) address of the page origin.

XeroxMinDot – specifies the dot (“Y”) address of the page origin.

XeroxHighlight – specifies an index into the default Windows color palette. This determines the color displayed for “highlight color” references. There are three special values: “RED”, “BLUE”, and “GREEN” which are translated to the values 249, 252, and 250 respectively.

XeroxDefaultJDL – specifies the name of the JDL that will be used until another is located in the print stream.

XeroxDefaultJDE – specifies the name of the JDE that will be used until another is located in the print stream.

NOTE: Note: the values for XeroxDefaultJDL and XeroxDefaultJDE are used to construct a section name of the form [JDL:jdl – JDE:jde]. For each Metacode document this section is read to determine processing options that remain in effect until a JDL/JDE is encountered in the print stream.

XeroxDefaultPMODE - specifies the default coordinate system of Xerox graphic and images until another is located in the print stream or set from JDL/JDE section with PMODE=.

FORMS – specifies the name of an FRM that will appear on every page of the output.

FORMAT – specifies the name of a PDE that will be used to construct a section name of the format “[PDE:pdename]”, which will then be interpreted.

FONTScount – specifies the number of fonts that are present in this section.

FONTS*nnn* – specifies the name of font *nnn* in the font list. *nnn* may be 000-999.

PMODE – "LAND" or "PORT". Specifies the requested page orientation.

BEGINcount – specifies the number of logical page definitions that are present in this section.

BEGIN*nn* – specifies the position on the displayed page of a logical page. *nn* may be 00-99.

AFP Only Entries

NOTE: Docuview has its own method for locating printer resources. The following directories will be used to load printer resources as a last option only if Docuview is unable to locate them otherwise.

AfontDir – specifies the path where AFP fonts are stored

FormdefDir – specifies the path where AFP formdefs are stored

OverlayDir – specifies the path where AFP overlays are stored

PsegDir – specifies the path where AFP page segments are stored

OverlayBuffers – specifies the maximum number of AFP overlays that will be buffered in memory

DefaultFormdef – specifies the name of the AFP formdef that will be used to resolve copygroup references

DropShading – allows the elimination of a certain kind of AFP shading patterns from the rendered images

Appendix C: Bldcache Utility

BldCache.exe is a new command line utility that constructs a Docuview Resource Cache from your Documanage System. It is installed in the Docusave Server application directory.

BldCache.exe can be executed from the command line within the Docusave Server application directory and uses the settings in the Docusave Server.ini by default.

BldCache.exe requires the following command line syntax when executed from a different location:

```
BldCache  domain id pw router dir “resourceSet”
```

Parameter	Explanation
domain	Type the Domain for which the Documanage User ID you are logging into.
id	Type the Documanage user ID.
pw	Type the Documanage user password.
router	Type the TCP/IP address for the Documanage Router.
dir	Type the directory location where the printer resources will be copied. Note: The DVCache folder name must be one level. Here is an example: C:\DVCache\
resourceSet	Type the Documanage Cabinet where the printer resources are stored.

Appendix C: Bldcache Utility

Appendix D: Utility and Batch File

Windows

Addres Utility and AddResources Batch File

The *addres* command line utility submits printer resources to the input queue one printer resource at a time. It is installed in the Docusave Server application directory.

The *addres* utility requires the following command line syntax:

addres comProtocol queueName serverName fileName

comProtocol = QueueType

queueName = Input Queue Name

serverName = Input Server Name

fileName = Directory Path of Printer Resource + Printer Resource Name

Example:
addres tcpip Docusave Server localhost
C:\Dmkr_FP\Resources\C0tr172.382

The *addResources* batch file is used to submit a series of printer resources to the input queue by using wildcard characters. It is installed in the Docusave Server application directory.

The commend syntax is the same as for the addres utility with exception of the wildcard designations at the end command line.

Example 1:
addResources tcpip Docusave Server localhost
C:\Dmkr_FP\Resources*.382

Example 2:
addResources tcpip Docusave Server localhost
C:\Dmkr_FP\Resources*.*

UNIX

Addres Utility and AddResources Batch File

The *addres* command line utility submits printer resources to the input queue one printer resource at a time. It is installed in the Docusave Server application directory.

The *addres* utility requires the following command line syntax:

addres comProtocol queueName serverName fileName

comProtocol = QueueType

queueName = Input Queue Name

serverName = Input Server Name

fileName = Directory Path of Printer Resource + Printer Resource Name

Example:
addres tcpip Docusave Server localhost /home/username/resources/C0TR172.382

The *addResources* script file is used to submit a series of printer resources to the input queue by using wildcard characters. It is installed in the Docusave Server application directory.

The commend syntax is the same as for the addres utility with exception of the wildcard designations at the end command line.

New Example 1:
addResources tcpip Docusave Server localhost /home/username/resources/*.382

New Example 2:
addResources tcpip Docusave Server localhost /home/username/resources/*.*

Appendix E: Commcommander Imagecreate Driver

Overview

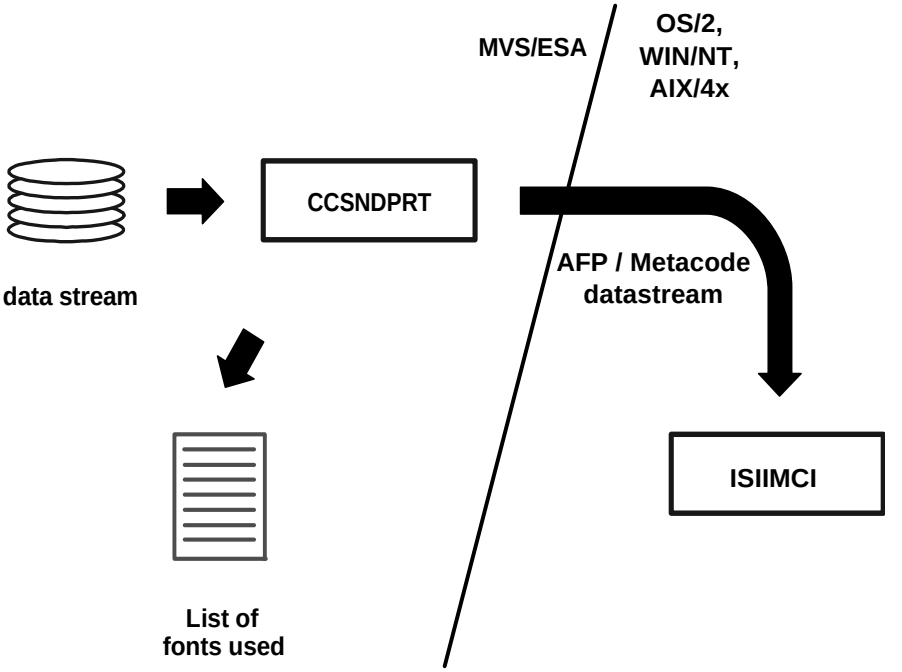
NOTE: Skywire Software's Commcommander is a prerequisite installation for the Imagecreate Driver.

The Commcommander Imagecreate Driver sends a data stream from MVS, across your network, to your PC. The Imagecreate Driver contains two components, CCSNDPRT and ISIIMCI, that run on a MVS host and PC server machine, respectively.

NOTE: CCSNDPRT replaces the DFCIMCO utility from the Imagecreate product.

The data transfer is handled either by TCP/IP or IBM's Advanced Program-to-Program Communications (APPC). In an APPC context, the Imagecreate Driver and ISIIMCI are transaction programs using LU 6.2 to handle data and message transfers.

The driver makes sure the mainframe-based Printing Resources needed in the upcoming data stream-to-image conversions are available. If not available, the required Printing Resources are downloaded at this time.



Downloading a Data Stream (Mainframe to PC)

Commcommander Imagecreate Driver/MVS is a batch program that downloads an image data stream, one document at a time, to be converted by transferring data from a mainframe running MVS to a server running Windows or RS6000 AIX. Separate JCL is provided for Xerox Metacode and IBM AFP data streams.

Commcommander Imagecreate Driver/NT/AIX can be directed to store the Data Stream into either a Docucorp Queue Systems queue or a file (e.g., LocalDisk). The file destination is specified in the Imagecreate Driver Initialization (.ini) file with the QueueType keyword.

The Commcommander Imagecreate Driver is started differently depending on the communications protocol you are using.

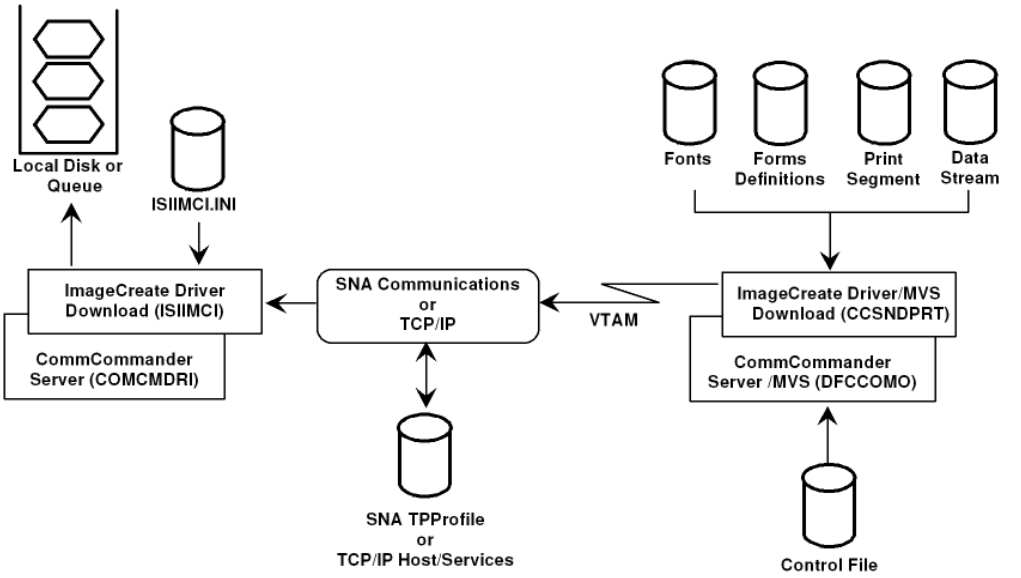
TCPIP

For TCPIP, the Commcommander Server (comcmdri) inbound message handler must be up and running. This will start a PC session that connects and communicates with the Imagecreate Driver on the Host MVS machine.

SNA

For SNA, the Imagecreate Driver/NT/AIX is started from Microsoft's or IBM's SNA services communication products, Advanced Program-to-Program Communications (APPC). When the Commcommander Imagecreate Driver/MVS program is executed, it automatically initiates a conversation with the PC through APPC, which automatically starts a PC session and initiates the PC transaction program. The PC transaction program, in turn, executes Commcommander Imagecreate Driver/NT/AIX.

The following diagram illustrates the downloading procedure, from the mainframe to the PC.



Setting Up Commcommander Imagecreate Driver

NOTE: The installation of the Commcommander Imagecreate Driver is part of the Docusave Server product installation procedure.

To Set Up Commcommander Imagecreate Driver/NT

Commcommander Imagecreate Driver (ISIIMCI) is automatically invoked whenever a request is received from the partner node. ISIIMCI is considered a transaction program (TP). A TP must be predefined, via transaction profile, before invocation.

Defining Transaction Profiles and Initialization Entries

TCP/IP

- **NT** — For information on defining transaction program profiles, see section “Setting Up (Inbound) Commcommander Initialization Section Entries for NT/AIX” in the Commcommander manual.

- **AIX** — For information on defining transaction program profiles, see section “Setting Up (Inbound) Commcommander Initialization Section Entries for NT/AIX” in the Commcommander manual.

- **MVS** — For information on the Communication Initialization File, see section “To Customize the Outbound Communication Initialization File for MVS” in the Commcommander manual.

SNA

- **NT** — For information on defining transaction program profiles, see section “Defining Transaction Programs” in the Commcommander manual. Use the following values when defining the Image Create TP:

TP Name	ISIIMCI
Command line	C:\ISIIMC\ISIIMCI.EXE C:\ISIIMC (this parameter specifies the location of the Imagecreate Driver .INI file).

- **AIX** — For information on defining transaction program profiles, see section “Defining Transaction Programs” in the Commcommander manual. Use the following values when defining the Image Create TP:

Profile name	ISIIMCI
Command line	/u/isiimc
TPN	ISIIMCI
Full Path	/u/isiimc/isiimci

Standard output	/u/isiimc
Standard error	/u/isiimc

- **MVS** — For information on the Communication Initialization File, see section “To Customize the Outbound Communication Initialization File for MVS” in the Commcommander manual.

To Set Up the Commcommander Imagecreate Driver INI File

Because ISIIMCI operates automatically, you must set the options before the transaction profile is invoked.

Determine whether the Imagecreate output will be written to a file or a Docucorp Queue Systems queue, and set the “QueueType=” parameter accordingly.

Determine if the outbound message handler, Commcommander Imagecreate Driver/MVS, will specify the file or queue name to which the Imagecreate output will be written. If the Commcommander Imagecreate Driver is to override the MVS specification, change the appropriate parameters in the ISIIMCI.INI file.

To set these options, edit the appropriate “.INI” file.

NOTE: Remember to remove the semicolon (;) from Column 1 to activate the selected parameters.

```
[Imagecreate]
;       ISIIMCI.INI
;       (a semicolon in position one is a comment line)
;                       ---------------------------------------------------
;                       Parameters for writing a resource to a queue
 ;                  (default none)
```

```
;                The name of your server machine
QueueServer=
;               Name given when the Queue Daemon started (default IMCI)
QueueName=IMCI
;                Queue Type (default LocalDisk)
;                    NAMEDPIPE   (NetBEUI)
;                    SOCKET      (TCP/IP)
;                    LocalDisk   (Local Disk Queue)
QueueType=SOCKET
;               ------------------------------------------------
;              Transaction Program Name (SNA only) (default ISIIMCI)
;              The name of the TP used to invoke the Imagecreate driver
;                    ISIIMCI
TPName=ISIIMCI
;               ------------------------------------------------
;              Parameter for transport communications (default SOCKET)
;             The name of your server machines output protocol
;                    SOCKET
;                    CPIC
InboundProtocol=SOCKET
;               ------------------------------------------------
;             VFont Directory (default current directory)
VfontDir=
;              ------------------------------------------------
;             AFont Directory (default current directory)
AfontDir=
;             ------------------------------------------------
;            PSEG Directory (default current directory)
PsegDir=
;           ------------------------------------------------
;          Overlay Directory (default current directory)
OverlayDir=;
;           ------------------------------------------------
;         Forms Definition Directory (default current directory)
```

```
FormdefDir=
;               ---------------------------------------------
;               MFont Directory (default current directory)
MfontDir=
;               ---------------------------------------------
;               FRM Directory (default current directory)
FrmDir=
;               ---------------------------------------------
;               IMG Directory (default current directory)
ImgDir=
;               ---------------------------------------------
;              Temporary Directory (default current directory)
TempDir=
;               ---------------------------------------------
;               Trace Level, used for problem determination
TraceLevel=0
```

Field	Value
QueueServer (Required)	The name of the server machine where the Docucorp queue is running. (Check with your LAN Administrator for the network server name.) If you're running the Commcommander Imagecreate Driver on the same machine as the Docucorp queue management program (ISIQMSD.EXE), you **must** leave this parameter blank (e.g., code the parameter as “QueueServer=”). There is no default value.
QueueName (Required)	If the Data Stream is routed to a queue, this specifies the name of the Docucorp queue in which the Imagecreate output is written. This parameter overrides any name supplied from the partner node. The default value is **IMCI**.
QueueType (Required)	The type of Docucorp queue to which the Imagecreate output is written. The default value is **SOCKET**.

Field	Value
TPName (Windows SNA only)	This value is the name of the TP defined in Microsoft SNA Server Symbolic Destination used to invoke the VRF Inbound driver. The default value is **ISIIMCI**.
InboundProtocol	This value must match the protocol used when the Commcommander Server Inbound was started. The valid values are **CPIC** for SNA or **SOCKET** for TCP/IP. The default value is **SOCKET**.
VfontDir	The directory where Font Printer Resources are stored. For convenience, you can specify the same directory or sub-directory where you installed the Imagecreate executable modules. The default value is **the current directory**.
AfontDir	The directory where AFP Font Printer Resources are stored. For convenience, you can specify the same directory or sub-directory where you installed the Imagecreate executable modules. The default value is **the current directory**.
PsegDir	The directory where AFP PSEG Printer Resources are stored. For convenience, you can specify the same directory or sub-directory where you installed the Imagecreate executable modules. The default value is **the current directory**.
OverlayDir	The directory where AFP Overlay Printer Resources are stored. For convenience, you can specify the same directory or sub-directory where you installed the Imagecreate executable modules. The default value is **the current directory**.
FormDefDir	The directory where Forms Definitions are stored. For convenience, you can specify the same directory or sub-directory where you installed the Imagecreate executable modules. The default value is **the current directory**.

Field	Value
MfontDir	The directory where Xerox Metacode font printer resources are stored. For convenience, you can specify the same directory or sub-directory where you installed the Imagecreate executable modules. The default value is **the current directory**.
FrmDir	The directory where Xerox Metacode FRMs are stored. For convenience, you can specify the same directory or sub-directory where you installed the Imagecreate executable modules. The default value is **the current directory**.
ImgDir	The directory where Xerox Metacode IMGs are stored. For convenience, you can specify the same directory or sub-directory where you installed the Imagecreate executable modules. The default value is **the current directory**.
TempDir	The directory where temporary work files, encountered during normal processing, are stored. For convenience, you can specify the same directory or sub-directory where you installed the Data Stream executable modules. For more information about the appropriate entry, see the Path= explanation above. The default value is **the current directory**.
TraceLevel	Trace Level number used for problem determination. If you are having protocol trouble, Skywire Software Support may ask you to generate a trace log, which contains detailed information about the Commcommander API call sequence. Before calling Support, try running with a tracelevel of "5". This will be helpful information used by the support analyst when solving your problem.

Setting Up Imagecreate Driver/MVS

NOTE: The installation and setup of the Commcommander Imagecreate Driver/MVS is part of the Commcommander product installation procedure. For more information, see the Commcommander manual.

Using Commcommander Imagecreate Driver

To Use Commcommander Imagecreate Driver/NT/AIX

1. **(SNA/NT) Make sure SNA Server communication software is up and running.**
   -or-
   (TCP/IP) Make sure Skywire Software's Commcommander Server Inbound program is up and running.

2. **Make sure the proper settings are specified in the Commcommander Imagecreate Driver ISIIMCI.INI file.**

3. **Make sure the directory named in the ISIIMCI.INI file already exists if you are directing the resource to a file. If you are directing the resource to a Docucorp Queue Systems queue, make sure the queue program ISIQMSD is up and running.**
   When using the SNA protocol, the Imagecreate transaction program (TP) is automatically invoked by Microsoft's or IBM's SNA Server communications software whenever the partner node's Imagecreate Driver starts the conversation with this local node Imagecreate Driver. For TCP/IP, the TP is invoked by Skywire Software's Commcommander Server Inbound.
   Multiple Transaction Program Profiles may be defined, with each pointing to different directories containing individual copies of Commcommander Imagecreate Driver and its ISIIMCI.INI file. One example in which you might want to use multiple profiles is to have one TPN set up to write to a file, and another to put the resource in a Docucorp Queue Systems queue. By selecting the TPN name in the partner node control cards, you can select the type of destination to which the resource will be routed.

To Review Messages

- Browse any and all Commcommander error messages by viewing the ISIIMCI.LOG file, which is located in the same directory as the ISIIMCI

executable file. The ISIIMCI.LOG file contains any protocol errors or trace messages produced during a Commcommander session. The messages are written chronologically from top to bottom. Therefore to view the most recent messages go to the bottom of the file and scroll backwards.

To Use Commcommander Imagecreate Driver/MVS

- The Commcommander Imagecreate Driver/MVS initiates the conversation with the partner PC node. This is a program that can be executed as a job step within an existing job, or as a separate job. You have the option of transferring the Data Stream to the PC in the same job that creates the Data Stream or, by allocating the Data Stream as a permanent file, you can transfer it at anytime using a separate job.

NOTE: Commcommander Imagecreate Driver/MVS is part of the Commcommander product. For more information, see the Commcommander manual.

Appendix E: Commcommander Imagecreate Driver

Using Commcommander Imagecreate Driver

Appendix F: Installation Verification for Documanage

To verify installation and configuration, use the list below.

- Identify all the printer resources required by the print streams that will be archived. (For Documaker FP users, the common font list is a good place to find the complete list of fonts.)
- Add all the required printer resources to the Docusave Server input queue using *addres*. (Be aware that some printer resources themselves have resource dependencies.)
- Archive the printer resources using Docusave Server.
- Use the Documanage client to verify that the resources appear in the appropriate folders in the specified printer resources cabinet.
- Obtain a small print stream that's representative of the documents that will be archived.
- Configure the Docusave Server profile to match the JSL setup on the printer to properly identify the DJDE flag, RSTACK, RPAGE, and RAUX identifiers (Metacode only).
- Modify the index configuration file to accurately reflect the layout of the USER.IMAGE.INDEX comment records in the sample print stream.
- Add the sample print stream to the Docusave Server input queue using *qcutil* (AIX and NT only).

- Archive the print stream using Docusave Server.
- Use the Documanage client to verify that the document appears in the appropriate folder and that the folder properties and document properties are set as expected.
- Export the document from the Documanage client, send this export file to the printer, and verify the appearance of the printed document.

Appendix G: Glossary

Glossary of Docusave Server terms and concepts

Concept/Term	Docusave Server Definition
Accesscommander	Accesscommander archives print streams and printer resources to an imaging system for subsequent viewing with Docuview LFS.
AFP	IBM's Advanced Function Printing format.
Bursting	The process of separating a job into individual documents for processing.
CCITT group 3 and 4	Internationally standardized methods for compressing bitmap representations of printed pages, first developed for FAX systems but also used in electronic file formats such as TIFF, AFP, and PDF.
DAPI	Docucorp Application Programming Interface is a library of utility functions including those used to assemble or disassemble jobs for Docusave Server.
Data Map	A comment record optionally inserted into AFP and Metacode documents during Docusave Server processing that describes the format of Index information elsewhere in the document
Dither	A process for improving the visual quality of pixel images during scaling. Dithering is often used to improve the appearance of scaling on rasterized typographic elements.

Concept/Term	Docusave Server Definition
DJDE normalization	Metacode documents in a job may inherit dynamic job control settings from control statements located “upstream.” Docusave Server normally accumulates these statements, eliminates redundancy, and emits the resulting DJDE statements within each document as it is archived to an imaging system. This supports reprinting and reuse.
DocID	The document designator used by a DRR record to identify which document to retrieve. The format of the DocID varies depending on the imaging or archive system.
Docucreate IC	The software formerly known as IRIS, Docucreate IC loads megafiles and TIFF images as printable forms into EDLs for subsequent inclusion in documents.
Document	An atomic unit of work on which Docusave Server document processing occurs. Documents are logical and may correspond with abstract datasets, which are not necessarily able to be imaged (e.g., “smart documents” such as CAR and DCD formats). Typically each document processed by Docusave Server corresponds with a single document (often a single file) sent to the output imaging system.
Document Archive	A document storage facility. Typically a long term system for electronic documents.
Document Management System	A document archive that includes version control, check-in/out to sequence updating of documents, automated distribution or workflow, and sophisticated query operations.
DRP (Document RePrint Record)	A Docusave Server job input record that designates a compressed AFP or Metacode print stream to be retrieved and bundled with its dependant printer resources for subsequent printing.
DRR (Document Retrieval Request)	A Docusave Server job input record that designates image pages to be retrieved and formatted for subsequent processing by Docucreate IC.
EDL (Electronic Document Library)	A library of publishing elements which are combined by Documaker into a document.

Concept/Term	Docusave Server Definition
FNBI	FileNet® Banded Image. A FileNet proprietary image file format
Imagecreate	Imagecreate renders print streams (using printer resources) into bitmapped page images and archives these images to an imaging system for subsequent viewing using the viewer native to the imaging system.
Imaging System	An archival imaging, document management, or storage system. The output repository that forms the primary output location for Docusave Server processing.
Indexes or “filing keys”	In addition to document contents, documents to be filed have certain external properties. Examples include the document name, creation date, document type, account number, policy number, etc. These properties, when used to control filing in a repository, are often called indexes or “filing keys.”
Input queue	A structured collection of jobs, each containing one or more documents. Jobs in an input queue form the primary inputs for Docusave Server.
Job	A compound unit of work, corresponding to a single Docusave Server input queue element, containing one or more documents.
Job Type	The specifier of the input format of a job in the input queue. Job Types are three letter codes, such as ‘PRT,’ which are used to determine job processing libraries based on the job library naming convention.
Library	A named package of programmed functionality. Docusave Server uses interchangeable libraries to locate and invoke functional operations on jobs and documents.
Library Naming Convention	Docusave Server uses a naming convention to associate libraries with job and processing functions. For example, document pre-processing libraries share a suffix that distinguishes them from document post-processing libraries located in the same file directory.

Concept/Term	Docusave Server Definition
Megafile	An image file format suitable for loading into an EDL by Docucreate IC.
Metacode	A Xerox® proprietary format for printing.
MO:DCA	IBM®'s Mixed Object Document Content Architecture. A format for representing document content
MP TIFF	Multi-page TIFF. A single file containing multiple page images in TIFF format.
PCL	Printer Control Language is a language (a set of command codes) that enables applications to control Hewlett-Packard DeskJet, LaserJet, and other HP printers.
PDF	Adobe®'s Portable Document Format. A popular file format for expressing document contents
Pixel	From **pic**ture-**el**ement, a miniature display element used in a raster printing or display device. Many pixels in an array can represent a rasterized image.
Postprocessor	An optional processing step, after a job or document has been processed. These are sometimes used to modify error handling and completion or error recovery behavior.
Preprocessor	An optional processing step, before a job or document is processed. These are sometimes used to verify, correct, or extend the index information or contents of a job or document, before normal processing begins.
Printer Resources	Certain types of documents depend upon external files called resources. Examples are Metacode documents that depend upon externally named fonts, overlays, logos, etc. These must be loaded into a Metacode printer before a document can be printed on it. Even among the same type of document, there may be variation in printer resources used.
Process Type	The specifier of the desired processing for a job in the input queue. Process Types are three letter codes, such as 'RAS,' which are used to determine the document processing library based on the processing library naming convention.

Concept/Term	Docusave Server Definition
QMSAPI	Queue Management System Application Programming Interface is a library that allows programmers to manipulate queues in the format used by Docusave Server.
Rasterize	Convert to a pixel representation for display. Many modern output devices, such as display screens and high-speed printers output pixels in an array, which together approximate a continuous image. Rasterizing an image converts it from another format into a series of discrete pixels for display or printing.
Resource Set	A named collection of resources, as stored in an archive or document management system. The name and date of a resource is local in scope to the resource set in which it resides. Use of multiple resource sets allows Docusave Server to track multiple printing configurations at once.
Roll-back	Job level processing after error detection, to revert partially processed jobs. If all the documents in a job cannot be successfully archived together, roll-back processing deletes those which were processed – so the entire job can be subsequently run without the risk of duplicated documents.
TIFF	Tagged Image File Format, an industry standard file format often used for bitmap images, especially of formatted documents. When used as an alternative to MP TIFF, it signifies that each page of a document is stored in a separate file.

Appendix G: Glossary

Glossary of Docusave Server terms and concepts