Documaker

Documaker Server Installation Guide

version 11.3

Skywire Software, L.L.C.
3000 Internet Boulevard
Suite 200
Frisco, Texas 75034
www.skywiresoftware.com

Phone:	**(U. S.)**	**972.377.1110**
	(EMEA)	**+44 (0) 1372 366 200**
FAX:	**(U. S.)**	**972.377.1109**
	(EMEA)	**+44 (0) 1372 366 201**
Support:	**(U. S.)**	**866.4SKYWIRE**
	(EMEA)	**+44 (0) 1372 366 222**
		support@skywiresoftware.com

PUBLICATION COPYRIGHT NOTICE

Copyright © 2008 Skywire Software, L.L.C. All rights reserved.
Printed in the United States of America.
This publication contains proprietary information which is the property of Skywire Software or its subsidiaries. This publication may also be protected under the copyright and trade secret laws of other countries.

TRADEMARKS

Skywire® is a registered trademark of Skywire Software, L.L.C.

Docucorp®, its products (Docucreate™, Documaker™, Docupresentment™, Docusave®, Documanage™, Poweroffice®, Docutoolbox™, and Transall™), and its logo are trademarks or registered trademarks of Skywire Software or its subsidiaries.

The Docucorp product modules (Commcommander™, Docuflex®, Documerge®, Docugraph™, Docusolve®, Docuword™, Dynacomp®, DWSD™, DBL™, Freeform®, Grafxcommander™, Imagecreate™, I.R.I.S. ™, MARS/NT™, Powermapping™, Printcommander®, Rulecommander™, Shuttle™, VLAM®, Virtual Library Access Method™, Template Technology™, and X/HP™ are trademarks of Skywire Software or its subsidiaries.

Skywire Software (or its subsidiaries) and Mynd Corporation are joint owners of the DAP™ and Document Automation Platform™ product trademarks.

Docuflex is based in part on the work of Jean-loup Gailly and Mark Adler.

Docuflex is based in part on the work of Sam Leffler and Silicon Graphic, Inc.
Copyright © 1988-1997 Sam Leffler.
Copyright © 1991-1997 Silicon Graphics, Inc.

Docuflex is based in part on the work of the Independent JPEG Group.
The Graphic Interchange Format© is the Copyright property of CompuServe Incorporated. GIF℠ is a Service Mark property of CompuServe Incorporated.

Docuflex is based in part on the work of Graphics Server Technologies, L.P.
Copyright © 1988-2002 Graphics Server Technologies, L.P.

All other trademarks, registered trademarks, and service marks mentioned within this publication or its associated software are property of their respective owners.

SOFTWARE COPYRIGHT NOTICE AND COPY LIMITATIONS

Your license agreement with Skywire Software or its subsidiaries, authorizes the number of copies that can be made, if any, and the computer systems on which the software may be used. Any duplication or use of any Skywire Software (or its subsidiaries) software in whole or in part, other than as authorized in the license agreement, must be authorized in writing by an officer of Skywire Software or its subsidiaries.

PUBLICATION COPY LIMITATIONS

Licensed users of the Skywire Software (or its subsidiaries) software described in this publication are authorized to make additional hard copies of this publication, for internal use only, as long as the total number of copies does not exceed the total number of seats or licenses of the software purchased, and the licensee or customer complies with the terms and conditions of the License Agreement in effect for the software. Otherwise, no part of this publication may be copied, distributed, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, manual, or otherwise, without permission in writing by an officer of Skywire Software or its subsidiaries.

DISCLAIMER

The contents of this publication and the computer software it represents are subject to change without notice. Publication of this manual is not a commitment by Skywire Software or its subsidiaries to provide the features described. Neither Skywire Software nor it subsidiaries assume responsibility or liability for errors that may appear herein. Skywire Software and its subsidiaries reserve the right to revise this publication and to make changes in it from time to time without obligation of Skywire Software or its subsidiaries to notify any person or organization of such revision or changes.

The screens and other illustrations in this publication are meant to be representative, not exact duplicates, of those that appear on your monitor or printer.

Contents

Introduction

2 System Requirements

2 Operating systems

3 Networks

4 Documaker Requirements

5 On Windows

6 On z/OS (OS/390)

7 On UNIX Systems

7 On AIX systems

8 On Linux systems

9 On Solaris systems

10 Docupresentment Requirements

10 Docupresentment Workstation

10 Docupresentment Server

10 On Windows systems

11 On AIX systems

11 On Solaris systems

12 On Linux systems

13 Web Server

14 iDocumaker Requirements

14 Basic requirements

14 Client requirements

14 Server requirements

15 Application server requirements

15 Using the right Java environment

15 Additional recommendations

16 iPPS Requirements

16 Basic requirements

16 Client requirements

16 Server requirements

17 Web server

17 Additional recommendations

18 EWPS Requirements

18 Web application server

19 Message bus systems

20 Docupresentment, JRE, and Documaker versions

21 Downloading JAXB
22 Documanage Requirements
22 Workstation and Administrative Applications
22 On Windows systems
23 Server and Router Applications
23 On Windows systems
24 On Linux systems
24 On AIX systems
25 On Solaris Systems
26 Databases Supported
26 Document Retention Management Requirements
26 Third-Party Storage Solutions

Installing Documaker Server on z/OS

28 Overview
28 Installing the System
28 Downloading the system
28 Installing from a cartridge tape
29 Configuring Your Printer Types
29 Configuring your system for Xerox Metacode
29 Configuring your system for AFP printers
29 Configuring your system for PostScript printers
29 Configuring your system to produce PDF files
29 Configuring Host Archive
30 Checking Your Installation
30 Optimizing Performance
30 Additional Information
31 Installing the System
31 Installing from the COSS Site
33 Installing from a 3480 Cartridge Tape
33 Copying the LOADDAP Job
34 Modifying the LOADDAP Job
34 Submitting the LOADDAP Job
35 Binding the Documaker DB2 Plan to Your DB2 Subsystem
36 DB2 Bind considerations
38 Upgrading Your System
38 Determining your System's Patch Level
39 Customizing Your System

39 If You Do Not Have a Source Code License
39 Running the COMPSDK Job
40 Running the GENDATAL Job
40 Running the GENPRTL Job
40 If You Have a Source Code License
41 Running the COMPALL Job
42 Running the LINKALL Job
43 Configuring Your Printer Type
43 Creating Xerox Metacode Print Streams
43 Changing the Xerox JSL
43 Viewing the JSL
44 JSL requirements
45 Changing the FSISYS INI File
46 PrtType:XER control group
47 Printer1 control group
47 Pre-compiling FAP Files into Metacode Files
48 Making Printer Resources Available
48 Fonts
48 Loading fonts onto your printer
48 Logos
49 Changing the JCL
49 DAP procedure (or the GenPrtX job)
49 GENERXER job
49 Creating AFP Print Streams
49 Allocating AFP-related Datasets
49 Making Printer Resources Available
50 Fonts
50 Page segments
50 FormDefs
50 Overlays
50 Pre-compiling FAP Files into AFP Overlays
50 Changing the FSISYS INI File
50 PrtType:AFP control group
51 Printer control group
51 Printer1 control group
51 Changing the JCL
51 Using the DAP procedure (or the GenPrtX job)
51 GENERAFP job
52 Creating PostScript Print Streams
52 Changing the FSISYS INI File
53 Changing the JCL

53 Printing PostScript Print Streams
54 Creating PDF Files
54 Changing the FSISYS INI File
55 Generating PDF files using DAL scripts
55 Generating PDF files using the MultiFilePrint rule
56 Including PostScript and TrueType fonts
57 Changing the JCL
58 Creating the PDF Print Stream
59 Configuring Host Archive
60 Configuring Host Archive Using DB2
60 Creating DB2 Objects
60 Running the BINDDB2 (or COMPDB2) Jobs
61 Running ALLOCDB2 and COMPDB2
61 Modifying the FSISYS INI File for Host Archive
61 ArcRet control group
61 Archival control group
62 DBHandler:DB2 control group
62 DB2_FileConvert control group
62 DBTable control group
63 Trigger2Archive control group
63 Changing the JCL for Host Archive
64 Archiving to Documanage
64 Modifying the FSISYS File
65 Adding IP Addresses and System/Domain Names
65 Changing the JCL
66 Archiving Using Stacked DPA
66 Changing the JCL
67 Checking Your Installation
67 Checking Your FSIUSER and FSISYS Settings
70 Checking the JCL
70 Running GenTrnX, GenDataX, and GenPrtX separately
71 Running DAPRUN (or GenTrnX, GenDataX, and GenPrtX) and DMRUNLM
71 Running DAPRUNC to Improve Performance
71 Logging messages when running RPEX1
73 Logging messages when running UTEX1
75 Using the DMS1 Resources
76 Checking Your FSIUSERL Settings
78 Checking the JCL
80 Running GenArcDX
80 RPEX1 messages

81 UTEX1 messages
82 Running GenArcX
82 RPEX1 messages
82 UTEX1 messages
83 Printing the Output
83 Xerox Metacode printer
83 BARR-attached Metacode printer
83 IBM AFP printer
84 Optimizing Performance
85 Compile Options
85 OPTIMIZE
85 NOVALIDATEHANDLE
85 INLINE
85 Language Environment (LE) or c/370 Runtime Options
85 REPORT
85 HEAP
86 File Types and Characteristics
86 Block sizes
86 Record formats
86 Number of files
86 Extract Files
86 Defining the extract file as a VSAM ESDS
86 DEFLIB
86 Placing DDT files into a VSAM KSDS
87 SETRCPTB
87 Moving the SETRCPTB member into a VSAM KSDS
87 FAPLIB
87 TRNFILE
87 NEWTRN
87 NAFILE
87 Creating a NAFILE as a VSAM file
87 POLFILE
87 Creating a POLFILE as a VSAM file
87 Recipient Batch Files
88 Pre-compiled Xerox Metacode Files (PMETLIB)
88 Placing PMETs into a VSAM KSDS
88 Print Files
88 Setting Your FSISYS INI Options
88 Caching Options
88 Caching FAP files
88 DDT files

89 Using/Caching Xerox PMET files
89 Using AFP Overlays
89 Logging Options
89 LogTransactions option
89 LogToConsole option
90 Debug Options
90 If_Rule control group
90 RunMode control group
90 Other Options
90 MaxRecsPerTransaction option
90 AliasPrintBatches option
91 Defining the Extract File as a VSAM ESDS
93 Moving DDT Files into a VSAM KSDS
95 Moving JDTs into a VSAM KSDS
97 Moving PMETs into a VSAM KSDS
99 Moving SETRCPTB to a VSAM KSDS
101 Creating NAFILEs and POLFILEs as VSAM KSDSs
102 The LOADDAP Job
108 z/OS Libraries
111 Uploading and Downloading Resources
112 Transferring Files
113 Handling International Characters
113 Running CPCNV on a PC
113 Running CPCNV on z/OS
114 Xerox Image, Font, and Form Files
114 Xerox Pre-compiled Metacode (PMET) Files
115 z/OS Programs and Utilities
115 AFPCOPY utility
115 AFPOPT utility
115 AFPFMDEF utility
115 AFP2MVS utility
115 BARR2MVS utility
115 BARRWRAP utility
115 CPCNV utility
115 FAP2AFP utility
116 FAP2CFA utility
116 FAP2MET utility
116 FAP2OVL utility

116 FDT2CFA utility
116 FIXOFFS utility
116 FSIVER utility
116 SEQ2KSDS utility
116 GenArc program
117 GenData program
117 GenPrint program
117 GenTrn program
117 LOG2PSEG utility
117 PDFKEY utility
117 PDS2VSAM utility
117 RCP2VSAM utility
117 SAMPAFP utility
117 TRANSLAT utility
118 XERDNLD utility

Installing Documaker Server on UNIX/Linux

120 Overview
120 What's on the Installation CD
120 Installing the System
120 Checking Your Installation
120 Using the Documentation
120 Start-Up Tips
120 Optimizing Performance
120 Uploading and Downloading Resources
120 Directory Structures
121 What's on the Installation CDs
121 Installation routines
121 On-line documentation
122 Installing the System
122 Before You Begin
122 Setting up libraries
122 Setting up users
124 Checking Your Installation
124 Solaris and Linux
124 AIX
124 HP-UX
124 Testing with the RPEX1 Sample Resources

125 Processing the RPEX1 Sample Resources
125 Printing the RPEX1 Output
126 Testing WIP with RPEX1
126 Viewing Archived RPEX1 Forms
127 Using the Documentation
128 Start-Up Tips
128 Using long file names
128 Naming conventions for printers in Windows
129 Optimizing Performance
129 CPU Time
129 EXCPs
129 Wall Clock Time
129 Batch Window
129 Setting Your FSISYS INI Options
129 Caching FAP files
130 DDT files
130 Caching Xerox PMET files
130 Using AFP Overlays
130 Logging Options
130 LogTransactions option
130 LogToConsole option
131 Debug Options
131 Run Options
131 Other Options
131 MaxRecsPerTransaction option
131 AliasPrintBatches option
132 Uploading and Downloading Resources
132 Transferring Resource Files Between UNIX/Linux and Windows
132 Uploading a Library from PC to UNIX
132 Downloading Print Streams from UNIX to PC
133 Directory Structures
133 Run Time Environment (RTE)
133 Master Resource Library (MRL)
134 Software Development Kit (SDK)

Installing Documaker Server on Windows

136 Overview
136 What's on the Installation CD
136 Installing the System
136 Checking Your Installation
136 Using the Documentation
136 Start-Up Tips
136 Optimizing Performance
137 What's on the Installation CD
137 On-line documentation
138 Installing the System
138 Before You Begin
138 Using NTFS drive formats
138 Setting up libraries
138 Installation
139 Checking Your Installation
139 Testing with the RPEX1 Sample Resources
139 Processing the RPEX1 Sample Resources
140 Printing the RPEX1 Output
140 For an AFP printer
140 For a Metacode printer
140 For a PCL printer
140 For a PostScript printer
141 Testing WIP with RPEX1
141 Viewing Archived RPEX1 Forms
142 Using the Documentation
143 Start-Up Tips
143 Using long file names
143 Naming conventions for printers in Windows
143 Displaying objects in color in Windows
144 Optimizing Performance
145 Setting Your FSISYS INI Options
145 Caching Options
145 Caching FAP files
145 DDT files
145 Using/Caching Xerox PMET files
145 Using AFP Overlays
146 Logging Options
146 LogTransactions option
146 LogToConsole option

146 Debug Options
146 If_Rule control group
146 Run Options
146 RunMode control group
147 Other Options
147 MaxRecsPerTransaction option
147 AliasPrintBatches option

CHAPTER 1

Introduction

This manual provides instructions for installing Documaker Server software on computers running z/OS (MVS), UNIX/Linux, and 32-bit Windows, such as Windows 2000 and Windows Vista.

This table will help you find the information you need:

To install Documaker on	See
z/OS (MVS)	Installing Documaker Server on z/OS on page 27
UNIX/Linux	Installing Documaker Server on UNIX/Linux on page 119
Windows	Installing Documaker Server on Windows on page 135

A summary of the system requirements follows.

System Requirements

The following Oracle Insurance applications run on a variety of operating systems and hardware platforms. Make sure you have these components before you install the following applications.

Operating systems

The following applications run on a variety of operating systems, principally Windows 32-bit operating systems such as Windows 2000, Windows 2003, Windows Vista, and Windows XP, UNIX/Linux 32-bit operating systems such as AIX, Solaris, and Linux x86, and z/OS (OS/390). This table shows the various product offerings and the operating systems under which they run.

NOTE: To store a Documaker version 11.0 resource library in Documanage, you must have Documanage version 6.3 SR 2 or version 6.4 SR 1 or higher.

	Windows+	z/OS	AIX	Linux	Solaris
Docucreate	Yes	No	No	No	No
Documaker Studio	Yes	No	No	No	No
Documaker Workstation	Yes	No	No	No	No
Docupresentment	Yes	No	Yes	Yes	Yes
iDocumaker ++	Yes	No	Yes	Yes	Yes
iPPS +++	Yes	No	No	No	No
EWPS ++	Yes	No	Yes	Yes	Yes
Documanage	Yes	No	No	Yes	No
Documaker					
GenTrn	Yes	Yes	Yes	Yes	Yes
GenData	Yes	Yes	Yes	Yes	Yes
GenPrint	Yes	Yes	Yes	Yes	Yes
GenWIP	Yes	Yes	Yes	Yes	Yes
GenArc	Yes	Yes	Yes	Yes	Yes

+ Includes Windows 2000, Windows 2003, Windows XP, Windows Vista, Windows 2000 Server, and Windows 2003 Server. Be sure to have the latest service packs installed.

++ Runs under any operating system that supports the Java Virtual Machine.

+++ Runs under Microsoft Windows 2000 Server and Windows 2003 Server.

	Windows+	z/OS	AIX	Linux	Solaris
Printers**					
AFP	Yes	Yes	Yes	Yes	Yes
GDI	Yes	No	No	No	No
HTML	Yes	No	Yes	Yes	Yes
Metacode	Yes	Yes	Yes	Yes	Yes
PCL	Yes	No	Yes	Yes	Yes
PCL 6*	Yes	No	Yes	Yes	Yes
PDF	Yes	Yes	Yes	Yes	Yes
PostScript	Yes	Yes	Yes	Yes	Yes
RTF	Yes	No	Yes	Yes	Yes
VIPP	Yes	Yes	Yes	Yes	Yes
XML**	Yes	Yes	Yes	Yes	Yes

+ Includes Windows 2000, Windows 2003, Windows XP, Windows Vista, Windows 2000 Server, and Windows 2003 Server.

* You must have PCL 6 or higher for Unicode support on PCL-compatible printers. PCL 6 support became available in version 10.2.

** Printer support depends on licensing. For example, PDF and HTML are licensed separately for the PPS market and PDF is licensed separately for the z/OS market.

Networks

The system does not use any specific network calls and is expected to work on any network compatible with Microsoft programs.

NOTE: The network file server you use with Documaker Studio or Documaker Workstation must be a 100% Windows network compatible. Some UNIX systems that offer NFS support are not 100% Windows compatible and some UNIX systems do not honor Windows file locking calls and may not be suitable for use as a file server in a true multi-user environment.

Documaker Requirements

Your computer must have certain software and hardware components to run the programs that comprise the Documaker system. Depending on your software license, operating environment, and the market your solution was created for, these requirements vary.

The following tables outline the minimum hardware we use to test Documaker and are included for informational reasons only. We suggest that you run the system on a configuration designed specifically to meet your data processing needs.

Note: For more specific information on the GenArc program and the additional archive and retrieval capabilities available, refer to the Documaker Server System Reference.

Should your company have special needs, contact your sales representative and keep in mind that, by using upload and download programs, additional functionality is available.

On Windows

This table outlines the minimum requirements and the minimum hardware we use to test the system on a single user Windows 2000 workstation. We suggest you run the system on computer configured apprpriately for the version of Windows you use and with the processing power to meet your needs.

	Requirements
CPU	A Pentium or compatible 400 MHz processor or higher
Memory*	512mb RAM
Hard disk**	250mb
Printer	Any printer which supports PCL level 5 (HP IV or higher), PostScript level 2, IBM AFP, or Xerox Metacode
Printer memory***	8mb for HP printers
Compiler****	Microsoft Visual Studio .NET 2003 for Windows XP Microsoft Visual Studio .NET 2003 for Windows 2000

* Additional memory, while not required, will improve system performance.
** Depends on the volume of data you must process
*** Additional memory may be required if printing complicated graphics or using a lot of fonts.
**** Only required if you need to recompile custom code you have developed.

	Tested on
CPU	Pentium II 400 MHz
Memory	512mb RAM
Hard disk	250mb
Printer	HP-compatible printers supporting PCL5
Printer memory	8mb RAM

On z/OS (OS/390)

The DAP and Documaker products run on the following versions/releases of IBM's operating systems:

- OS/390 version 1.1 to version 2.10
- z/OS version 1.6 and higher

NOTE: Following OS/390 version 2.10, new versions were named z/OS. Documaker runs on OS/390 and z/OS. In this manual, OS/390 and z/OS are referred to as *z/OS* unless otherwise noted.

No product upgrades are required and no incompatibility problems have been reported when running Documaker on any of these operating system releases.

	Requirements
Hard disk	150mb
Printer	Any printer which supports IBM AFP, Xerox Metacode, or Adobe PostScript
Runtime library	IBM Language Environment for OS/390 version 2.10 or higher
Compiler	(Only necessary if adding custom code to the system) IBM C/C++ Compiler for OS/390 version 2.10 or higher

NOTE: Regardless of the type of computer you run the system on, to print charts on Xerox Metacode printers, you must have a GVG card. To print charts on IBM AFP printers, you must have a GOCA card.

The amount of hard disk space you will need depends mainly on the volume of data you must process. Keep in mind too, that the C/C++ compiler is only required if you plan to write your own custom rules and recompile the source modules provided in the Software Developer's Kit (SDK).

On UNIX Systems

For all UNIX systems, you can use any printer that supports IBM AFP, PCL, PostScript level 2, or Xerox Metacode. For HP printers, you need at least 8mb of memory, more if you are printing complicated graphics or using a lot of fonts. The amount of hard disk space you need depends on the volume of data you process. Keep in mind too, that a compiler is only required if you plan to recompile the system, such as if you customize the source code or use a runtime library other than the one shown for your operating system.

NOTE: For any UNIX installation, first make sure you have the uudecode, uncompress, and awk utilities installed.

On AIX systems

	Requirements
Operating system	AIX version 5.2 or higher
Model*	pSeries - Power RISC
Compiler	(Only necessary if adding custom code to the system) IBM Visual Age C/C+ version 6 IBM C/C++ Enterprise Edition for AIX v7
Runtime library	C Set ++ Runtime for AIX 5.0 or higher

	Tested on
Model*	pSeries p650
CPU	6 x 1.45GHz Power4+ processors
Memory*	12GB
Hard disk**	Two 36.4GB 10, 000RPM Ultra3 SCSI drives

*Additional memory and a faster CPU is not required, but will improve performance.
** Additional space required for your customized forms

On Linux systems

	Requirements
Operating system	GNU/Linux distributions RedHat Enterprise Linux (RHEL) version 5.1 or higher SuSE Linux Enterprise Server (SLES) version 9.3 or higher
Model	Intel/AMD based systems
Compiler	(Only necessary if you are adding custom code to the system) GNU C/C++ compiler: gcc-3.3.5-5 v3.3.5 or higher, gcc-c++-3.3.5-5 v3.3.5 or higher
Runtime library	libgcc-3.3.5-5 v3.3.5 or higher or compatibility module libstdc++-3.3.5-5 v3.3.5 or higher or compatibility module
Additional requirements	- The installer requires the sharutil package for the uudecode applet. - RedHat Enterprise Linux version 5.1 or higher needs these compatibility modules for the C++ runtimes: compat-libstdc++-33 compat-libstdc++-296 (required to use the current ORA native DBHandler for Oracle 10g in Documaker) - Any other utilized third-party software required packages, such as Oracle, DB2, MySQL, and so on.

	Tested on
Model*	Dell PWS450
Operating system	RedHat Enterprise Linux (RHEL) version 5.1 VMWare guest SuSE Linux Enterprise Server (SLES) version 9.4 VMWare guest
CPU	2 x 2.40GHz Xeon processors
Memory*	3GB
Hard disk**	70GB SCSI drive

*Additional memory and a faster CPU is not required, but will improve performance.
** Additional space required for your customized forms

On Solaris systems

	Requirements
Operating system	Sun Solaris 9/SunOS 5.9 (SPARC based) or higher
Model*	UltraSPARC based
Compiler	(Only necessary if adding custom code to the system) Sun ONE Studio 8
Runtime library	Core Solaris 9

	Tested on
Model*	Sun Fire v240 Server
Operating system	Solaris v9 SPARC
CPU	2 x 1.28GHz UltraSPARC IIIi Cu
Memory*	2GB
Hard disk**	Four 36GB SCSI drives

*Additional memory and a faster CPU is not required, but will improve performance.
** Additional space required for your customized forms

DOCUPRESENTMENT REQUIREMENTS

Your computer must have certain software and hardware components to run Docupresentment. Depending on your software license and operating environment, these requirements vary.

Docupresentment Workstation

For a Docupresentment workstation, you must have a personal computer equipped with the following:

- Microsoft Internet Explorer version 6.0 or later for Windows 2000 or Windows XP or higher
- Adobe ® Acrobat Reader version 7.0 or higher

Docupresentment Server

You can run Docupresentment on the following operating systems:

- Windows
- AIX
- Linux
- Solaris

NOTE: For Docupresentment Workstation and Server, you must have Java 1.5 or higher.

These tables provide more detailed information on our minimum platform requirements for testing Docupresentment implementations.

On Windows systems

	Tested on
Operating system	Windows 2000
CPU*	1xPentium II - 400 mHz
Memory*	256 MB
Hard disk (RTE)	25 MB
Hard disk (MRL)**	75 MB

*Additional memory and a faster CPU is not required, but will improve performance.
** Additional space required for your customized forms

On AIX systems

	Requirements
Operating system	AIX version 5.2 or higher
Model*	pSeries - Power RISC
Compiler	(Only necessary if adding custom code to the system) IBM Visual Age C/C+ version 6
Runtime library	C Set ++ Runtime for AIX 5.0 or higher

	Tested on
Model*	pSeries p650
CPU	6 x 1.45GHz Power4+ processors
Memory*	12GB
Hard disk**	Two 36.4GB 10, 000RPM Ultra3 SCSI drives

*Additional memory and a faster CPU is not required, but will improve performance.
** Additional space required for your customized forms

On Solaris systems

	Requirements
Operating system	Sun Solaris 9/SunOS 5.9 (SPARC based) or higher
Model*	UltraSPARC based
Compiler	(Only necessary if adding custom code to the system) Sun Workshop C/C++ v5.0
Runtime library	Core Solaris 9

	Tested on
Model*	Sun Fire v240 Server
Operating system	Solaris 9/SunOS 5.9 (SPARC based)
CPU	2 x 1.28GHz UltraSPARC IIIi Cu
Memory*	2GB
Hard disk**	Four 36GB SCSI drives

*Additional memory and a faster CPU is not required, but will improve performance.
** Additional space required for your customized forms

On Linux systems

	Requirements
Operating system	GNU/Linux distributions RedHat Enterprise Linux (RHEL) version 5.1 or higher SuSE Linux Enterprise Server (SLES) version 9.3 or higher
Model	Intel/AMD based systems
Compiler	(Only necessary if you are adding custom code to the system) GNU C/C++ compiler: gcc-3.3.5-5 v3.3.5 or higher, gcc-c++-3.3.5-5 v3.3.5 or higher
Runtime library	libgcc-3.3.5-5 v3.3.5 or higher or compatibility module libstdc++-3.3.5-5 v3.3.5 or higher or compatibility module
Additional requirements	- The installer requires the sharutil package for the uudecode applet. - RedHat Enterprise Linux version 5.1 or higher needs these compatibility modules for the C++ runtimes: compat-libstdc++-33 compat-libstdc++-296 (required to use the current ORA native DBHandler for Oracle 10g in Documaker) - Any other utilized third-party software required packages, such as Oracle, DB2, MySQL, and so on.

	Tested on
Model*	Dell PWS450
Operating system	RedHat Enterprise Linux (RHEL) v5.1 VMWare guest SuSE Linux Enterprise Server (SLES) v9.4 VMWare guest
CPU	2 x 2.40GHz Xeon processors
Memory*	3GB
Hard disk**	70GB SCSI drive

*Additional memory and a faster CPU is not required, but will improve performance.
** Additional space required for your customized forms

Web Server

This table outlines the web server requirements for each operating system:

Operating system	Web server
Windows	2000 Server or 2003 Server (or higher), such as Microsoft Internet Information Server 4.0 (or higher).
AIX	Web server for AIX 5.2, such as IBM's HTTP Server for AIX version 1.3.3.1 or higher with the Java Runtime Environment and/or JDK for AIX, version 1.4.0 or higher.
Linux	Web server for Linux, such as Apache 1.3.12 or higher or IBM HTTP Server 1.3.9 or higher.
Solaris	Web server for Sun Solaris 7 or higher on SPARC, such as Java Web Server 2.0 or Apache 1.3.9 with the Java Runtime Environment and/or JDK for Solaris, version JRE 1.4.0 or higher. IBM HTTP Server 1.3.9 or higher can also be used.

NOTE: We test Docupresentment version 2.2 and iDocumaker version 3.2 implementations on WebSphere Application Server (WAS) version 6.1.x and Tomcat 6.x Application Server.

iDocumaker Requirements

Your computer must have certain software and hardware components to run iDocumaker. This table outlines those requirements:

Basic requirements

	Requirements
Operating system	Windows XP, Windows 2003 Server, Linux, AIX, and Sun Solaris
Java Runtime Environment (JRE)	version 1.5 or higher
Physical memory/JVM heap memory*	1024MB/256MB RAM
Hard disk**	100 MB free
Other components	Keyboard and mouse or compatible pointing device
Monitor	Color SVGA monitor

* Additional memory will improve system performance.
** The amount of hard disk space you will need depends mainly on the volume of data you must process.

Client requirements

In addition to the basic requirements, each client should have the following:

- WIP Edit, version 11.3, patch 05 or higher
- Microsoft Windows XP Professional or later
- Adobe Acrobat Reader 7.0 or higher
- Microsoft Internet Explorer 7.0 or higher with these Internet security options enabled:
  - Run ActiveX controls and plug-ins
  - Script ActiveX controls safe for scripting
  - Allow cookies that are stored on your computer
  - Allow per-session cookies (not stored)
  - Active scripting

Server requirements

In addition to the basic and client requirements, the computer you will use as a server should be configured with the following:

- Apache Tomcat, IBM WebSphere MQ, or BEA WebLogic version 9.1
- A database such as Oracle version 8i or higher, Microsoft SQL 2000 or higher, IBM DB2 version 8.1 or higher, Sun MySQL version 4.3 or higher, or other JDBC version 2.0-supported databases
- Application servers that support the J2EE 1.4 specification. We test on Tomcat version 5.5 and WebSphere MQ version 6.1.

Application server requirements

In addition to the basic requirements, for the application server, you should have:

- Windows Server (Tomcat plug-in), Tomcat version 5.5 or higher, WebSphere MQ (formerly MQSeries) version 6.1, or WebLogic version 9.1
- One of these message busses:
  - WebSphere MQ
  - JMS
  - ActiveMQ
  - MSMQ
  - HTTP
- A database such as Oracle version 8i or higher, Microsoft SQL 2000 or higher, IBM DB2 version 8.1 or higher, Sun MySQL version 4.3 or higher, or other JDBC version 2.0-supported databases
- Docupresentment version 10.2 (IDS version 1.8) or higher

NOTE: We test iDocumaker version 3.2 implementations on WebSphere Application Server (WAS) version 6.1.x and Tomcat 6.x Application Server.

Using the right Java environment

This table shows various application servers we have tested with and the Java version you should use with those web servers and with iDocumaker.

For	Use
Tomcat 5.5 (or higher)	Tomcat using Java 1.5 (iDocumaker using the Servlet Specification 2.4 or higher)
WebSphere Application Server 6.1.x	WebSphere Application Server using Java 1.5 (iDocumaker using the J2EE standard 1.4)

Additional recommendations

We also recommend these additional products for your iDocumaker implementation:

Product	Description
Documaker	Use for rules publishing.

iPPS Requirements

Your computer must have certain software and hardware components to run iPPS. This table outlines those requirements:

Basic requirements

	Requirements
CPU	Pentium III or higher
Operating systems	Windows 2000, Windows XP, or Windows 2003 Server
Memory*	256MB RAM
Hard disk**	400 MB free
Other components	Keyboard and mouse or compatible pointing device
Monitor	Color SVGA monitor

* Additional memory, while not required, will improve system performance.
** The amount of hard disk space you will need depends mainly on the volume of data you must process.

Client requirements

In addition to the basic requirements, each client should have the following:

- WIP Edit, version 11.3, patch 05 or higher
- Windows 2000 Professional or XP Professional or later
- Adobe Acrobat Reader 7.0 or higher
- Microsoft Internet Explorer 6.0 or higher with these Internet security options enabled:
  - Run ActiveX controls and plug-ins
  - Script ActiveX controls safe for scripting
  - Allow cookies that are stored on your computer
  - Allow per-session cookies (not stored)
  - Active Scripting

Server requirements

In addition to the basic and client requirements, the computer you will use as a server should be configured with the following:

- Internet Information Services (IIS) with World Wide Web Server and File Transfer Protocol (FTP) Server
- A database such as Microsoft Access (Access 97 or higher), xBase, or SQL
- Visual Basic runtimes
- Microsoft XML Core Services 4.0 SP2 (msxml4) for iPPS 3.11 and below
- Microsoft XML Core Services 6.0 (msxml6) for iPPS 3.12 and higher
- ADO 2.6 or later
- A static IP address

Web server

For the web server, you should have:

- Minimum Pentium III with 512MB of RAM
- Windows 2003 Server
- Component Services or Microsoft Transaction Server
- Microsoft Visual Basic 6 Runtimes (included/installed with iPPS version 3.1 or higher)
- Microsoft Active Data Objects 2.6 or higher (included/installed with iPPS version 3.1 or higher)
- IBM WebSphere MQ (formerly MQSeries) or Microsoft Message Queue client
- ODBC-compliant database (Oracle, SQL Server, or DB2 recommended for production)
- Microsoft's XML parser:
  - MSXML 4.0 sp2 for iPPS version 3.11 and below
  - MSXML 6.0 for iPPS version 3.12 and higher
- Docupresentment version 10.2 (IDS version 1.8) or higher

Additional recommendations

We also recommend these additional products for your iPPS implementation:

Product	Description
Documaker	Use for rules publishing.

EWPS Requirements

The EWPS Java web application deployment supports Java Runtime Environment (JRE) version 1.5 or higher. For best results, use JRE version 1.6. This table shows you the version of Java you need for each supported platform:

For this platform	You need this version of Java
Microsoft Windows (x86-32 and x86-64)	
XP (SP 2) Server 2003 (SP2)	Sun Java 5 or higher
Linux (x86-32 and x86-64) 32- and 64-bit kernels	
SuSE Linux Enterprise Server (SLES) version 9.4 or higher RedHat Enterprise Linux (RHEL) version 5.1 or higher	Sun Java 5 or higher or IBM Java 5 or higher
Sun Solaris (SPARC) 32-and 64-bit kernels	
Solaris 9/SunOS 5.9 or higher)	Sun Java 5 or higher
IBM AIX 5L pSeries (RISC) 32- and 64-bit kernels	
version 5.2 TL 5200-09	IBM Java 5 or higher
version 5.3 TL 5300-07	IBM Java 6 or higher

Web application server

Whether running on Windows, Linux, Solaris, or AIX, you can use either of the following Java Web Application Servers:

- IBM WebSphere AS, version 6.1 or higher
- Tomcat version 5.5 or higher

NOTE: We test with both WebSphere and Tomcat. Other Java application servers should also work.

An EWPS Java web application deployment requires the following version of Docupresentment to process web services requests:

- Docupresentment version 2.2, patch 04 or higher

Message bus systems

EWPS communicates to Docupresentment using a message bus provider such as JMS, WebSphere MQ, or MSMQ. This table shows the possible message bus systems. Tested systems are indicated with a single asterisk (*).

For this platform	You can use one of these message busses
Microsoft Windows (x86-32 and x86-64) · XP (SP 2 · Server 2003 (SP2)	HTTP/SOAP(*) (**) MSMQ (*) IBM WebSphere MQ, version 5.3 or higher (*) ActiveMQ JMS (*) (***) Other JMS providers
Linux (x86-32 and x86-64) **** · SuSE Linux Enterprise Server (SLES) version 9.4 or higher · RedHat Enterprise Linux (RHEL) version 5.1 or higher	HTTP/SOAP(*) (**) IBM WebSphere MQ, version 5.3 or higher (*) ActiveMQ JMS (*) (***) Other JMS providers
Sun Solaris (SPARC) · Solaris 9/SunOS 5.9 or higher	HTTP/SOAP(*) (**) IBM WebSphere MQ, version 5.3 or higher (*) ActiveMQ JMS (*) (***) Other JMS providers
IBM AIX 5L pSeries (RISC) 32-bit and 64-bit · version 5.2 TL 5200-09 · version 5.3 TL 5300-07	HTTP/SOAP(*) (**) IBM WebSphere MQ, version 5.3 or higher (*) ActiveMQ JMS (*) (***) Other JMS providers

(*) Tested providers
(**) This provider has not been tested under heavy concurrency and load.
(***) At the time of publication, we noticed problems with the ActiveMQ JMS during heavy load testing, We expect Apache to address these issues in the future. Contact Apache for more information.
(****) Other Linux distributions should work at kernel v2.6.11.4-21 or higher but have not been tested.

Docupresentment, JRE, and Documaker versions

This table shows, for each supported and tested platform, the version of Docupresentment, the corresponding Java Runtime Environment (JRE) for Docupresentment, and the version of Documaker necessary to support EWPS.

For this platform	Docupresentment	JRE	Documaker
Microsoft Windows (x86-32 and x86-64) • XP (SP 2) • Vista (SP 1) • Server 2003 (SP2)	Version 2.2, patch 04* or higher	Sun JRE v1.5** Sun JRE v1.6	Version 11.3, patch 05 or higher
Linux (x86-32 and x86-64)**** • SuSE Linux Enterprise Server (SLES) version 9.4 or higher • RedHat Enterprise Linux (RHEL) version 5.1 or higher	Version 2.2, patch 04* or higher	Sun JRE v1.5** Sun JRE v1.6 IBM JRE v1.5** IBM JRE v1.6***	Version 11.3, patch 05 or higher
Sun Solaris (SPARC) • Solaris 9/SunOS 5.9 or higher	Version 2.2, patch 04* or higher	Sun JRE v1.5** Sun JRE v1.6	Version 11.3, patch 05 or higher
IBM AIX 5L pSeries (RISC) 32-bit and 64-bit • version 5.2 TL 5200-09 • version 5.3 TL 5300-07	Version 2.2, patch 04* or higher	IBM JRE v1.5** IBM JRE v1.6***	Version 11.3, patch 05 or higher

* A Docupresentment Windows installation includes a bundled Sun J2SE JRE version 1.6 which is used by default by the Docupresentment server.
** Running the Docupresentment web services interface under JRE version 1.5 requires JAXB 2. See Downloading JAXB on page 21 for more information.
*** IBM's J2SE JRE version 1.6 is only supported in Docupresentment version 2.2, patch 05 or higher and Documaker Shared Objects version 11.3, patch 06 or higher.
**** Other Linux distributions should work at kernel version 2.6.11.4-21 or higher but have not been tested.

NOTE: To find the latest version of Documaker, Docupresentment, iDocumaker, iPPS, or EWPS, log onto the following web site:

http://www.oracle.com/skywiresoftware/index.html

Downloading JAXB

JAXB provides a way to map XML and Java code and extend applications with XML and Web Services technologies. You must install and implement JAXB 2 to run the Docupresentment web services interface under JRE version 1.5.

Follow these steps to download the latest version of JAXB:

1. Go to the JAXB web site:

   https://jaxb.dev.java.net/

2. Click Download Now for the latest version of JAXB.
3. On the new page, click the Download the Binary link and save the ZIP file to your local machine.
4. Unzip the downloaded ZIP file into directories. There will be a \jaxb-ri directory with a \lib subdirectory.
5. From the \lib subdirectory, copy the jaxb-api.jar and jaxb-impl.jar files to the \lib directory of your Docupresentment installation.

DOCUMANAGE REQUIREMENTS

The hardware and software described here only includes what is required by the Documanage applications. These hardware recommendations are cumulative. Running multiple software applications on a single system requires that the system support the sum of all of the hardware recommended by all of the software vendors. For example, if each of two software applications require 50MB of free disk space and 256MB of available memory, then the required hardware should have 100MB of free disk space and 512MB of available memory.

For server products, the required minimum amount of hardware varies depending on the client load it supports, including the type and size of documents being processed. For instance, rendering Metacode documents with a 24-bit color depth on the server requires more server memory for fewer workstation connections.

NOTE: These recommendations are not intended as guidelines for high-volume production systems. They are for installing and running a workgroup with fewer than ten users, and they assume that the systems are not running other applications. Larger systems should be set up while working with Professional Services. The number of users, the amount of data, along with usage profiles and their impact on system operation should be considered.

Workstation and Administrative Applications

The Workstation and Administrative Applications can run on Microsoft Windows operating systems with Intel-compatible hardware.

Normally, the Workstation component is installed on end-user workstations by itself. You can, however, install the Workstation and Services applications on the same machine to test an installation and to perform trouble shooting.

On Windows systems

	Requirements
Operating systems*	Windows 2000 Professional (NT5) or XP Professional Be sure to install all available service packs. No additional software is required. All support libraries are installed or updated by the Documanage installers.
CPU	Intel-compatible; 256-MHz or faster Pentium III-or-better processor required; 512-MHz Intel Pentium/Celeron family, AMD K6/Athlon/Duron family, or compatible processor recommended
Memory	128MB available RAM or more recommended (64MB minimum supported; may limit performance and some features)
Hard Disk	40MB of available hard-disk space required for installation (harddisk usage will vary based on configuration)

* Windows XP Professional does not support Workstation scanning functions until Documanage release 6.4.

	Requirements
Monitor	Recommend 19" monitor or better at 1024 x 768 resolution or better for viewing and working with documents using Workstation software
Networking	Network adapter for the type of network to which you want to connect, and access to an appropriate network

* Windows XP Professional does not support Workstation scanning functions until Documanage release 6.4.

Server and Router Applications

The Server and Router Applications can run on Microsoft Windows, Linux, AIX, and Solaris operating systems and Intel-compatible hardware.

On Windows systems

	Requirements
Operating systems	Windows 2000 Professional (NT5), XP Professional Be sure to install all available service packs. No additional software is required. All support libraries are installed or updated by the Documanage installers.
CPU	Intel-compatible; 512-MHz or faster Pentium III-or-better processor required; 1-GHz Intel Pentium/Celeron family, AMD K6/Athlon/Duron family, or compatible processor recommended; multi-processor systems are fully supported by a multi-threaded server
Memory	Router: 50MB available RAM or more. Server: 38 MB available RAM or more (256MB minimum supported; this may limit performance and the availability of some features)
Hard disk*	Router: 80MB of available hard-disk space is required for installation Server: 100MB of available hard-disk space is required for installation
Peripheral	Additional peripheral storage (hard disk or other mass storage or networked storage) is required for all document storage.
Networking	Network adapter appropriate for the type of network to which you want to connect, and access to an appropriate network

* Disk usage will vary based on configuration.

On Linux systems

	Requirements
Operating systems	Red Hat Advanced Enterprise Server, versions 2.1, 3 and 4. All current fixes are required for each kernel release. The latest Linux kernel build announcements are available at: http://www.spinics.net/lists/announce-kernel/ No additional software is required. Free DCE/RPC for Linux is supplied in the libdcerpc.so.0.0.1 file packaged with your installation. It incorporates modifications that allow Documanage to run with the latest versions of Linux. ODBC and DCE libraries are included in your installation.
CPU	Intel-compatible only; 512-MHz or faster Pentium Pro-or-better processor required; 1-GHz Intel Pentium/Celeron family, AMD K6/Athlon/Duron family, or compatible processor recommended
Memory	Router: 50MB available RAM or more recommended Server: 384MB available RAM or more recommended (128MB minimum supported; may limit performance and some features)
Hard disk*	Router: 80MB of available hard-disk space required for installation Server: 100MB of available hard-disk space required for installation
Peripheral	Additional peripheral storage (hard disk or other mass storage or networked storage) is required for document storage.
Networking	Network adapter appropriate for the type of network to which you want to connect, and access to an appropriate network

* Disk usage will vary based on configuration.

On AIX systems

	Requirements
Operating systems	AIX 4.3 or later (for pSeries processors). We recommend that you install all available fixes. Encryption Support: /usr/lib/libcrypt.a (Encryption library) POSIX Thread Support (see http://www-306.ibm.com/software/awdtools/caix/) - libpthreads.a (POSIX thread library) - libpthreads_compat.a (POSIX thread additional compatibility layer) Note: Both libraries are included in the bos.rte.libpthreads fileset, which is installed by default. IBM DCE/RPC for AIX (see http://www-306.ibm.com/software/network/dce/) - libdce.a (IBM DCE main library) - libdcepthreads.a (IBM DCE support library) - libdcelibc_r.a (IBM DCE support library)
CPU	IBM pSeries supporting required AIX versions

* Disk usage will vary based on configuration.

	Requirements
Memory	Router: 50MB available RAM or more recommended Server: 384MB available RAM or more recommended (128MB minimum supported; may limit performance)
Hard disk*	Router: 80MB of available hard-disk space required for installation Server: 100MB of available hard-disk space required for installation
Peripheral	Additional peripheral storage (hard disk or other mass storage or networked storage) is required to store documents.
Networking	Network adapter for the type of network to which you want to connect, and access to an appropriate network

* Disk usage will vary based on configuration.

On Solaris Systems

	Requirements
Operating systems	The Documanage Server and Router run on SunOS 5.7 (Solaris 7) or later. We recommend that you install all available fixes. For recent announcements, see: http://www.spinics.net/lists/announce-kernel/ IBM DCE/RPC for Solaris; not compatible with Sun's ONC/RPC (see http://www-306.ibm.com/software/network/dce/)
CPU	Sun SPARC Server platform only, supporting required SunOS versions.
Memory	Router: 50MB available RAM or more recommended Server: 384MB available RAM or more recommended (128MB minimum supported; may limit performance)
Hard disk*	Router: 80MB of available hard-disk space required for installation Server: 100MB of available hard-disk space required for installation
Peripheral	Additional peripheral storage (hard disk or other mass storage or networked storage) is required for document storage.
Networking	Network adapter for the type of network to which you want to connect, and access to an appropriate network

* Disk usage will vary based on configuration.

Databases Supported

The Documanage Server supports several database management systems, including Microsoft Sybase Adaptive Server, DB2, Adaptive Server Anywhere, and Oracle.

Database/version	Driver/version
MSSQL 2000	Microsoft Data Access Components (MDAC) 2.8
DB2 7.2, 8.2	IBM DB2 Connect 7.2 or 8.2
Sybase Adaptive Server 12.5	Sybase Adaptive Server Enterprise (ASE) ODBC driver 4.20.00.67 or later
ORACLE 9i, 10g	Latest Oracle drivers (ODBC version 9.00.01.08 or later)
MySQL 4.1	Latest MySQL drivers

Document Retention Management Requirements

Optional Nearstor Network Attached Storage (NAS) with Snaplock compliance from Network Appliance, Inc., securely stores documents that Documanage has under Retention Management. Nearstor NAS is currently available with capacities between one terabyte and 96 terabytes.

Third-Party Storage Solutions

Documanage interfaces to third-party storage systems, which augment Documanage's enforcement of retention policies and provide other managed storage-system benefits. Contact your sales representative for the latest listing of storage system versions which have been certified for use with Documanage. Documanage interfaces to the storage systems listed here at this time:

Vendor	Product	Supported Versions*
Network Appliance, Inc.	NearStor/SnapLock	Data ONTAP 7.1 or newer
EMC, Inc.	Centera	3.0.1 or newer with Compliance (Governance) Edition or CE Plus

* Oracle Insurance applications support newer product versions only if they are backward compatible with the versions listed here.

Plug-in modules interface these storage systems to the Documanage Server. These modules have been developed by Oracle Insurance, which is solely responsible for their performance.

CHAPTER 2

Installing Documaker Server on z/OS

This chapter provides information on how to install and configure Documaker on a z/OS system.

NOTE: Following MVS version 5.2.2, new versions of this operating system were named OS/390. Following OS/390 version 2.10, new versions were named z/OS. Documaker Server runs on MVS, OS/390, and z/OS. In this manual, MVS, OS/390, and z/OS are referred to as *z/OS* unless otherwise noted.

You will find information on these topics:

- Overview on page 28
- Installing the System on page 31
- Upgrading Your System on page 38
- Customizing Your System on page 39
- Configuring Your Printer Type on page 43
- Configuring Host Archive Using DB2 on page 60
- Checking Your Installation on page 67
- Optimizing Performance on page 84

In addition, you'll also find information about...

- The LOADDAP Job on page 102
- z/OS Libraries on page 108
- Uploading and Downloading Resources on page 111
- z/OS Programs and Utilities on page 115

OVERVIEW

This chapter provides you with steps to install and configure Documaker Server on your z/OS system. This overview discusses the contents of this chapter.

NOTE: See System Requirements on page 2 for information about system requirements.

Installing the System

You can install Documaker Server for z/OS by...

- Downloading it from the Oracle Customer Online Support System (COSS) web site
- Installing it from a 3480 cartridge tape

Downloading the system

Downloading and Installing Documaker for z/OS from the Oracle COSS web site involves these steps:

1. Download the zip file from the COSS site to a Windows workstation.
2. Unzip the zip file on Windows and upload the unzipped file to z/OS.
3. Run the LOADDAPW Job to perform a RECEIVE of the contents of the file that was uploaded.
4. Run the BINDDB2 Job to BIND the DB2 Plan to the DB2 subsystem.

Installing from a cartridge tape

Installing Documaker for z/OS from the 3480 cartridge tape involves these steps:

1. Unload the LOADDAP Job from the tape.
2. Modify and submit the LOADDAP Job.
3. Run the BINDDB2 Job to BIND the DB2 Plan to the DB2 subsystem.

Once you have installed the Documaker software, perform the steps discussed in Binding the Documaker DB2 Plan to Your DB2 Subsystem on page 35.

Configuring Your Printer Types

The system produces output for Xerox Metacode, IBM AFP, PostScript, PDF, or compatible printers. The steps differ, depending on the printer you will use.

Configuring your system for Xerox Metacode

Here is an overview of the steps required to configure the system for Xerox Metacode or compatible printers. For detailed instructions, see Creating Xerox Metacode Print Streams on page 43.

1. Change the Xerox JSL.
2. Change the FSISYS file.
3. Pre-compile your FAP files.
4. Make your printer resources available. This can include uploading Xerox fonts, loading the fonts onto your printer, and converting logos.
5. Change the JCL for the DAP procedure (or the GenPrtX job) and the GENERXER job.

Configuring your system for AFP printers

Here is an overview of the steps required to configure the system for AFP or compatible printers. For detailed instructions, see Creating AFP Print Streams on page 49.

1. Allocate AFP-related datasets.
2. Make your printer resources available. This can include uploading AFP fonts, loading page segments, setting up FormDefs, and creating overlays.
3. Pre-compile your FAP files into AFP overlays.
4. Modify the FSISYS file.
5. Modify the JCL for the DAP procedure (or the GenPrtX job) and the GENERAFP job.

Configuring your system for PostScript printers

Here is an overview of the steps required to configure the system for PostScript printers. For detailed instructions, see Creating PostScript Print Streams on page 52.

1. Change the FSISYS INI file.
2. Modify the JCL.

Configuring your system to produce PDF files

Here is an overview of the steps required to configure the system to produce PDF files. For detailed instructions, see Creating PDF Files on page 54.

1. Change the FSISYS INI file.
2. Modify the JCL.

Configuring Host Archive

Here is an overview of the steps required to configure host archive. This is an optional step. For detailed instructions, see Configuring Host Archive on page 59.

Checking Your Installation

Here is an overview of the steps required to configure host archive. This is an optional step. For detailed instructions, see Checking Your Installation on page 67.

1. Check your FSIUSER and FSISYS settings.
2. Check your JCL.
3. Run the DAPRUN procedure (or GenTrnX, GenDataX, and GenPrtX).
4. Run GenArcDX (or GenArcX).
5. Print the output.

Optimizing Performance

Once you have installed, configured, and tested your system, you can review this topic to find ways to improve system performance. For more information, see Optimizing Performance on page 84.

Additional Information

At the end of this guide you will also find information about the LOADDAP job, z/OS libraries, uploading and downloading resources, and z/OS programs and utilities.

Installing the System

You can get Documaker Server for z/OS by...

- Downloading the software from the Oracle Customer Online Support System (COSS) web site.
- Ordering the software on a 3480 cartridge tape. See Installing from a 3480 Cartridge Tape on page 33 for more information.

NOTE: The object and executable modules were built using IBM's C/C++ for z/OS version 1.7 compiler and corresponding Language Environment (LE) product.

Installing from the COSS Site

Follow these instructions to install Documaker for z/OS by downloading the software from the Oracle Customer Online Support System (COSS) web site.

1 From the COSS Documaker installation web site, click the version of Documaker for z/OS you want to install.

If your license...	Click
Includes the PDF print driver	z/OS PDF
Does not include the PDF print driver	z/OS No PDF

On the page that appears, right click on the following files and select Save Target As to save these files to your workstation:

- readme.txt
- loaddapw.jcl
- Documakerzosrel113.zip

NOTE: The readme.txt file contains much of the same information listed below.

2 Unzip the Documakerzosrel113.zip file. It contains a file named *DAP113Pxx.xmit*. This file contains several z/OS PDSs and sequential datasets and is in a TRANSMIT (XMIT) format.

To perform the Documaker z/OS installation, you must transfer this file to z/OS in binary mode, then unzip (RECEIVE) it. The RECEIVE creates and populates the PDSs and sequential datasets that comprise the Documaker installation. Follow the remaining steps to transfer this file to z/OS and perform the RECEIVE.

3 Allocate a dataset on z/OS to which you can upload the DAP113Pxx.xmit file. Allocate this dataset using attributes similar to these:

```
Data Set Name . . . : FSI.DAP.XMIT   <= whichever name you choose
Organization  . . . : PS
Record format . . . : FB
Record length . . . : 80
Block size  . . . . : 3120
1st extent cylinders: 200            <= primary space needed
Secondary cylinders : 10             <= secondary space needed
```

4 Upload the DAP113Pxx.xmit file from Windows to the file on z/OS you allocated in the previous step. Be sure to upload this file to z/OS in binary mode.

5 Upload the LOADDAPW.JCL file (in text mode) to your JCL library on z/OS. Read the comments in the JCL and make the necessary modifications so it will run on your system.

NOTE: The LOADDAPW job is included in the standard JCLLIB on the 3480 cartridge tape. If you have previously installed Documaker z/OS from a 3480 cartridge tape, you may already have the LOADDAPW job in your JCLLIB PDS on z/OS.

6 Submit the LOADDAPW job. The LOADDAPW job consists of two steps. Each step executes the IKJEFT01 (batch TSO) program and invokes the RECEIVE command.

The first step receives the FSI.DAP.XMIT file into a PDS named *FSI.V113.DAP.XMIT.PDS* (or the name you chose). This PDS contains a member for each PDS or sequential dataset that needs to be installed.

The second step receives each of the members of this PDS and creates a PDS or sequential dataset for each of these members.

Once the LOADDAPW job finishes, it will have created and populated 38 or so Documaker installation datasets, just as installing from a 3480 cartridge tape would have done.

Now go to Binding the Documaker DB2 Plan to Your DB2 Subsystem on page 35.

Installing from a 3480 Cartridge Tape

Follow these instructions to install Documaker for z/OS from a 3480 cartridge tape. The tape contains several files including JCL files, object modules, executable modules, and sample resources you can use to test your installation.

To install from a tape, you perform these tasks:

- Copying the LOADDAP Job on page 33
- Modifying the LOADDAP Job on page 34
- Submitting the LOADDAP Job on page 34

Copying the LOADDAP Job

The LOADDAP job is included on the installation tape in the JCLLIB dataset. You can use the JCL listed below to unload the LOADDAP job. Refer to any documentation that came with the tape to verify the exact version number of the files and the volume serial number of the tape.

Make any necessary modifications to this JCL (jobcard, dataset names, and so on) and submit the job. This job should receive a return code of *oo*.

```
//JWCI    JOB (33005),'LOAD LOADJCL',CLASS=T,MSGCLASS=X,
//        NOTIFY=JWC
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*       LOADLOAD - COPY THE LOADDAP JOB FROM TAPE TO MVS.
//*
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//S1      EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//INDD     DD  DSN=FSI.V110.JCLLIB,DISP=SHR,UNIT=CART,
//         VOL=SER=D11001
//OUTDD    DD  DSN=FSI.V110.TEMP.JCLLIB,DISP=(,CATLG),UNIT=SYSDA,
//         SPACE=(CYL,(1,1,1))
//SYSIN    DD  *
  C I=INDD,O=OUTDD
  SELECT MEMBER=LOADDAP
//*
```

Modifying the LOADDAP Job

Before submitting the LOADDAP Job, you should modify it to meet your system requirements.

Change any jobcard, dataset names, unit names, and so on so it will run on your system. The job uses an instream proc called LOAD. At the bottom of the job, you can specify the parameters that will be passed into the LOAD proc. This portion of the job is listed below along with a description of the parameters.

This job should receive a return code of *04* or less in each step.

```
...

//STEP1 EXEC LOAD,TUNIT=CART,
//          TVOLSER=D11001,                <= CHANGE
//          HLQ='FSI.V110',                <= CHANGE (seldom needed)
//          CUSTHLQ='CUSTOMER.V110.INST',  <= CHANGE
//          DUNIT=SYSDA,
//          PDSTYPE=LIBRARY
```

Parameter	Description
TUNIT	The unit name that should be used for this tape device.
TVOLSER	The tape volume serial number. The volume serial number is printed on the installation tape.
HLQ	The High Level Qualifier of the installation datasets. Normally, the value already specified for this parameter is correct.
CUSTHLQ	The High Level Qualifier that should be used when creating the installation datasets on your z/OS system.
DUNIT	A generic or esoteric name for your z/OS system's DASD units.
PDSTYPE	The type of PDS to create. By default, the LOADDAP job creates the installation partitioned datasets as PDS/Es, not PDSs. If your installation does not support PDS/Es or if you choose not to use PDS/Es, change the PDSTYPE parameter in the LOADDAP job from LIBRARY to PDS.

Submitting the LOADDAP Job

The LOADDAP job reads the installation tape, creates the system datasets, sample resource datasets, font datasets, and so on and populates these datasets from the tape. LOADDAP should receive a return code of *00* in every step.

The LOADDAP job is listed on page 102.

NOTE: Once the LOADDAP job is finished, the JCLLIB will contain a sample JOBCARD member which can be copied into and used as a jobcard in subsequent jobs.

Now go to Binding the Documaker DB2 Plan to Your DB2 Subsystem on page 35.

Binding the Documaker DB2 Plan to Your DB2 Subsystem

Perform the steps in this topic after you have installed the Documaker for z/OS base product either by downloading it from the COSS web site or by unloading the contents of the 3480 cartridge to z/OS.

If you are running Documaker on z/OS, in most cases your resources (BDF, GRP, FOR, DAL resources, and so on) are stored in a library. On z/OS, this library is comprised of four DB2 tables. You must perform a DB2 BIND to run Documaker on z/OS and access these tables.

The BIND process creates a DB2 PLAN which the Documaker batch processes, such as GenTrn, GenData, and GenPrint use to access the library tables. Note that the GenArc and GenWIP batch processes also require the use of the DB2 PLAN if you use DB2 to store the archive or WIP data.

The job to perform the DB2 BIND is called *BINDDB2* and is provided in the JCLLIB. This job uses the DBRMLIB dataset created during the installation of Documaker for z/OS. The BINDDB2 job performs these tasks:

- BINDs the DB2LIB member of the DBRMLIB into a COLLECTION named DAPxxx_COL.
- BINDs the DB2LIB2 member of the DBRMLIB into a COLLECTION named DAPxxx_COL.
- BINDs the DAPxxx PLAN, consisting of the PACKAGE List of all PACKAGEs that are in the COLLECTION named DAPxxx_COL.

You can assign any name that meets your DB2 requirements to the COLLECTION and the PLAN. The PLAN name must match the PLAN name used when DB2 is invoked in the SYSTSIN DD statement in the GenTrn, GenData, GenPrint, GenArc, and GenWIP batch processes.

For example, the sample job DMRUNLM (Documaker Run using Library Manager) in JCLLIB invokes the DMLM Proc in PROCLIB. The DMRUNLM job contains a JCL override for the SYSTSIN DD statement. The content of this overriding SYSTSIN DD statement invokes the DB2 subsystem named DB8G, then runs the Documaker batch program named *GenTrn* and specifies that the DB2 PLAN named *DAP113* is to be used. The GenTrn program is stored in the library named *FSI.V113.LINKLIB*.

```
//DMRUNLM  EXEC DMLM
//GENTRN.SYSTSIN  DD  *
DSN SYSTEM(DB8G)
RUN PROGRAM(GENTRN) PLAN(DAP113) -
    LIB('FSI.V113.LINKLIB')
END
/*
```

Modify the BINDDB2 job to conform to your system requirements, then submit this job.

DB2 Bind considerations

When you migrate to a new version or patch level of Documaker, you typically need to run both the old and new versions for a period of time. Because of the connection between the Documaker executables (such as GenTrn, GenData, and GenPrint) and the DB2 PLAN, in most cases you can not simply change the batch job to use the appropriate LINKLIB executable (the new or old version). Instead, you must also generate a new DB2 PLAN for the new version of the Documaker executables to use.

As an example, assume you have Documaker version 11.3, patch 03 installed. When you installed this version of Documaker, you ran the BINDDB2 job and created a PACKAGE LIST named DAP113_COL and a PLAN named *DAP113*. You now need to install Documaker version 11.3, patch 05.

Using the names from the example above, here are two ways to set up two versions of Documaker on z/OS:

1. In the BINDDB2 job, BIND the new DB2LIB and DB2LIB2 DBRMLIB members (from version 11.3, patch 05) into a new PACKAGE LIST named *DAP113P05_COL* (for instance).

```
//SYSTSIN   DD  *
  DSN SYSTEM(DB8G)
  BIND PACKAGE(DAP113P05_COL) MEMBER(DB2LIB) ACT(REP) -
    ISOLATION(CS)
  END
/*
```

```
//SYSTSIN   DD  *
  DSN SYSTEM(DB8G)
  BIND PACKAGE(DAP113P05_COL) MEMBER(DB2LIB2) ACT(REP) -
    ISOLATION(CS)
  END
/*
```

2. Choose from these options:

   - BIND the current PLAN (DAP113) and include both the *old* PACKAGE LIST (DAP113_COL) and the *new* PACKAGE LIST (DAP113P05_COL).

     ```
     //SYSTSIN   DD  *
       DSN SYSTEM(DB8G)
       BIND PLAN(DAP113) ACT(REP) -
         ISOLATION(CS) RETAIN -
         PKLIST(DAP113_COL.*,DAP113P05_COL.*)
       END
     /*
     ```

     If you choose this option, be sure to include this FSISYS INI option:

     ```
     < DBHandler:DB2 >
         CurrentPackageSet = DAP113P05_COL
     ```

     The CurrentPackageSet INI option tells DB2 to use only the PACKAGEs in the DAP113 PLAN that are associated with the PACKAGE LIST named *DAP113P05_COL*. These PACKAGEs are the new ones you bound in step 1.

- BIND a new PLAN (D113P05) and include only the new PACKAGE LIST (DAP113P05_COL).

```
//SYSTSIN   DD  *
  DSN SYSTEM(DB8G)
  BIND PLAN(D113P05) ACT(REP) -
    ISOLATION(CS) RETAIN -
    PKLIST(DAP113P05_COL.*)
  END
/*
```

  If you choose this option, you must also change the SYSTSIN DD statement input to point to the new PLAN name, *D113P05*. Here is an example:

```
//GENTRN.SYSTSIN  DD  *
  DSN SYSTEM(DB8G)
  RUN  PROGRAM(GENTRN) PLAN(DP113P05) -
    LIB('FSI.V113.P05.LINKLIB')
  END
/*
```

3 Regardless of which option you chose, you must point the LIB parameter in the SYSTSIN DD statement to the *new* LINKLIB that contains the new version of Documaker (GenData, GenPrint, GenArc, and so on).

Upgrading Your System

Upgrading to a new version of Documaker or applying corrective service (patches) to your Documaker system requires that you perform a standard Documaker installation from tape.

Determining your System's Patch Level

You can determine the patch level of your Documaker system by running a Job provided in the FSIVERX member of JCLLIB. This Job (see below) runs the FSIVER utility which examines the contents of a *LINKLIB* dataset and looks for patch identifiers contained in the Documaker program modules in the LINKLIB dataset.

The FSIVER utility then produces a report listing the patches it found. Unless instructed otherwise, FSIVER produces both a detailed patch report and a summary patch report. See the Docutoolbox Reference for more information on the FSIVER utility.

```
//ZDA      JOB   (33005),'FSIVER - 110  ',CLASS=T,MSGCLASS=X,
//         NOTIFY=&SYSUID
//*
//         SET HLQ='FSI.V110'   <== SET HIGH LEVEL QUALIFIER
//        SET RES='RPEX1'     <== SET RESOURCE (E.G. RPEX1, UTEX1)
//*
//         JCLLIB  ORDER=&HLQ..PROCLIB
//*
//
*****************************************************************
//*  PROGRAM  : FSIVER
//*  PURPOSE  : CREATES A REPORT THAT LISTS WHICH PATCHES HAVE BEEN
//*             APPLIED TO THE PROGRAMS IN THE LINKLIB REFERENCED BY
//*              THE LINKLIB DD STATEMENT.
//*
//*  PARMS    : /I=PROGRAM (NAME OF MEMBER IN DD:LINKLIB)
//*                 OR '*' TO LIST PATCH LEVEL OF ALL PROGRAMS IN
//*                  DD:LINKLIB.
//*
//
*****************************************************************
//FSIVER EXEC PGM=FSIVER,PARM='/ /I=*'
//*
//STEPLIB  DD DSN=&HLQ..LINKLIB,DISP=SHR
//         DD DSN=SYS1.SCEERUN,DISP=SHR
//LINKLIB  DD DSN=&HLQ..LINKLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
```

Customizing Your System

This step is optional.

Documaker is designed as a rules-based engine. Many rules are provided in the base product. These rules are located in the various source modules that make up a segment of the product called RULLIB. The Documaker architecture also lets you add your own rules, called *custom* rules. Some custom rules are provided, as examples, in a segment of the product called CUSLIB. If you want to modify these custom rules or add custom rules of your own, you should make your changes or additions to the CUSxxxxx modules that are provided in the Documaker Software Developer's Kit (SDK).

After making changes to any source modules in the SDK, you will need to compile these source modules into object modules and then link a new GenData program. The GenData program is also known as the Rules Processor.

If You Do Not Have a Source Code License

The standard Documaker license includes a subset of the Documaker source and header modules referred to as the Software Developer's Kit (SDK). On z/OS, this SDK comprises the SOURCE and INCLIB datasets. Follow the steps outlined in the following topics.

Running the COMPSDK Job

To compile the modules provided in the Documaker SDK, modify and submit the job in COMPSDK. The COMPSDK job invokes the CBC compile proc for approximately 56 source modules. Each source module is compiled into an object module, and the object modules are written to the OBJLIB. The COMPSDK job should receive a return code of *04* or less in each step. A portion of the COMPSDK job is shown below.

```
//*** COPY JOBCARD HERE...
//*
//* COMPSDK - COMPILE JCL FOR THE DOCUMAKER SOFTWARE DEVELOPER KIT
//*          (SDK), WHICH IS A SUBSET OF THE COMPLETE DOCUMAKER
//*           SOURCE.
//*
//CUSARC   EXEC CBC,MEM=CUSARC
//CUSBANNR EXEC CBC,MEM=CUSBANNR
//CUSBAT   EXEC CBC,MEM=CUSBAT
//CUSBITMP EXEC CBC,MEM=CUSBITMP
//CUSCALLB EXEC CBC,MEM=CUSCALLB
...
```

Running the GENDATAL Job

You can link-edit a new GenData program to incorporate any changes or additions you have made to the CUSxxxxx source modules. Run the job in JCLLIB member GENDATAL to link-edit GenData. The GENDATAL job first invokes the OBJUTIL proc to rebuild the object library directory, then it invokes the LINK proc to link-edit GenData. GENDATAL should receive a return code of *04* or less in each step. The GENDATAL job is shown below.

```
//*  COPY JOBCARD HERE
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*
//*         LINK-EDIT GENDATA
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*
//ALIAS    EXEC OBJUTIL
//*
//GENDATA  EXEC LINK,MEM='GENDATA'
```

Running the GENPRTL Job

You can link-edit a new GenPrint program to incorporate any changes or additions you have made to the CUSxxxxx source modules. Run the job in JCLLIB member GENPRTL to link-edit GenPrint. The GENPRTL job first invokes the OBJUTIL proc to rebuild the object library directory, then it invokes the LINK proc to link-edit GenPrint. GENPRTL should receive a return code of 04 or less in each step. The GENPRTL job is shown below.

```
//*  COPY JOBCARD HERE
//
*********************************************************************
//*
//*         LINK-EDIT GENPRINT
//*
//
*********************************************************************
//*
//ALIAS    EXEC OBJUTIL
//*
//GENPRINT EXEC LINK,MEM=GENPRINT
```

NOTE: If you purchased a license for PDF or PDF417 or both, the object modules you received include these features and will enable the resulting GenData and GenPrint to have the capabilities you licensed.

IF YOU HAVE A SOURCE CODE LICENSE

Follow the steps outlined in the following topics.

Running the COMPALL Job

This task is optional. If you do have a source code license, you can compile the full source using the COMPALL member in JCLLIB. The COMPALL member invokes one of the compile procs, which can be found in PROCLIB, for each of the 950 or so source members. You will usually need to separate the COMPALL member into several jobs. COMPALL compiles each source module into an object module. The object modules are written to the OBJLIB. The COMPALL job should receive a return code of *04* or less in each step.

A portion of the COMPALL Job is shown below.

```
 //*** COPY JOBCARD HERE...
//ADLER32A EXEC CBC,MEM=ADLER32A
//AFPAEG   EXEC CBC,MEM=AFPAEG
//AFPA2E   EXEC CBC,MEM=AFPA2E
//AFPBAG   EXEC CBC,MEM=AFPBAG
//AFPBARCD EXEC CBC,MEM=AFPBARCD
...
```

Running the LINKALL Job

This task is optional. Once the source modules are compiled into object modules, you need to link-edit the Documaker programs and utilities by running the LINKALL job. The LINKALL job first invokes the OBJUTIL proc, which rebuilds the object library directory, then it invokes the LINK proc for each program.

The LINKALL job should receive a return code of *04* or less in each step. A portion of the LINKALL job is shown below.

```
//*  COPY JOBCARD HERE ...
//
*********************************************************************
//*     LINKALL -  LINK ALL EXECUTABLES.
//*
//*     1. RUNS THE OBJECT UTILITY TO REBUILD THE @@DC370$ MEMBER
//*        OF THE OBJECT LIBRARY.
//*
//*     2. LINK-EDIT ALL OF THE EXECUTABLES FOR DAP.
//*
//
*********************************************************************
//*
//ALIAS1   EXEC OBJUTIL
//*
//AFP2MVS  EXEC LINK,MEM=AFP2MVS      <==  USED FOR AFP PRINT
//AFPCOPY  EXEC LINK,MEM=AFPCOPY
//AFPFMDEF EXEC LINK,MEM=AFPFMDEF     <==  USED FOR AFP PRINT
//BARR2MVS EXEC LINK,MEM=BARR2MVS     <==  USED FOR XEROX / BARR
//BARRWRAP EXEC LINK,MEM=BARRWRAP     <==  USED FOR XEROX / BARR
...
```

Configuring Your Printer Type

On z/OS, you can use the system to create print streams for these printing environments:

- Xerox Metacode (see Creating Xerox Metacode Print Streams below)
- AFP (see Creating AFP Print Streams on page 49)
- PostScript (see Creating PostScript Print Streams on page 52)
- PDF (see Creating PDF Files on page 54)

The following topics tell you how to configure your system for these environments.

Creating Xerox Metacode Print Streams

Configuring your system to produce Xerox Metacode print streams involves these tasks:

- Making changes to the Xerox JSL
- Making changes to the FSISYS INI file
- Pre-compiling your FAP files into MET files (this task is optional)
- Making printer resources available
- Making modifications to the JCL

Changing the Xerox JSL

Configuring the system to print to a Metacode printer involves setting various initialization file settings, specifically those in the PrtType:XER control group in the FSISYS initialization file, which is usually a member of the DEFLIB PDS.

To know what values need to be used for each setting, look at the Xerox printer's configuration file. Xerox configuration settings are kept on the printer in a JSL file. The JSL file contains the initial settings under which the printer operates.

You can have multiple JSL files resident on a single printer, with the desired JSL file specified when you start the printer. It is not necessary to start the printer with a specific JDE,JSL combination to print Metacode streams. You must, however, have a JDE,JSL combination which can be dynamically switched to for the duration of that print stream that satisfies the requirements explained below.

Specific values from the JDE,JSL combination must be available to the GenPrint program. You make these values available by placing them in the FSISYS file.

Viewing the JSL

Before you look at any JSL files, first determine which JSL file will be active when print files are sent to the printer. The JSL file name can contain up to six characters and always has a JSL extension. Most Xerox Metacode printers (except the 4235) contain a built-in console editor for creating, viewing, modifying, and printing JSL files.

NOTE: You may want to print your Xerox printer's current JSL file before you continue.

JSL requirements

To print a Metacode stream on your Xerox printer, you must have a Job Description Entry (JDE) with the parameter *VOLUME CODE=NONE*. If your JSL's default JDE entry or any other JDE entry in the JSL contains this parameter, you do not need to add another JDE entry. If the parameter is missing, you must add it. Follow your accepted procedures for making such JSL additions and modifications.

Once there is a JDE entry (or the default entry) that satisfies the VOLUME CODE requirement, enter values in the table below that correspond either to this JDE entry specifically or to the JSL file in general.

Use the table below to note the values in the Xerox JSL you will use. You will enter these values when you modify the FSISYS file.

JSL entry	Description	Values
DJDEIden	Corresponds to the IDEN statement's PREFIX= keyword (or PRE=). Indicates E for EBCDIC, A for ASCII, X for hexadecimal and a string value. If there is no E,A or X before the string value, an *E* is assumed by default.	
DJDEOffset	Corresponds to the IDEN statement's OFF= keyword.	
DJDESkip	Corresponds to the IDEN statement's SKIP= keyword.	
ImageOpt	If the JSL contains a "GRAPHICS = YES" statement, set ImageOpt to Yes. Otherwise, set it to No. You must set this option to Yes if you need the system to dynamically draw charts.	
JDEName	If you had to create a new JDE entry to specify *VOLUME CODE = NONE*, use the name of the new JDE entry. If a JDE entry already existed that contained *VOLUME CODE = NONE,* specify the name of that JDE (if it's the default, it's probably called *DFLT*).	
JDLName	The name of the JSL file.	
JDLHost	Corresponds to the VOLUME parameter's HOST= keyword. This value will be either IBMONL or IBMOS. If not specified, the default is *IBMOS*.	
JDLCode	Corresponds to the VOLUME statement's CODE= keyword for the initial JSL setting (at the top of the JSL file). This value will be either EBCDIC, ASCII, or NONE.	

JSL entry	Description	Values
JDLData	Corresponds to the LINE statement's DATA= keyword. Should be two numbers separated by a comma (omit the parentheses).	
PrinterInk	If the printer supports a highlight color, specify that color here. Otherwise, leave blank.	
JDLRStack	To find this value, find the RSTACK command (if there is one) and work backwards. The RSTACK command refers to a TEST label, which sets up a string test and refers to a CONSTANT, string, or value. For example... RSTACK: TEST=RST,DELIMITER=YES; RST: CRITERIA CONSTANT =(0,10,EQ,RSTK); RSTK: TABLE CONSTANT =X'13131313131313131313'; ...the value you would use would be: JDLRStack=0,10,EQ,X'1313131313131 3131313'	
JDLROffset	Locate the ROFFSET command (if there is one) and work backwards. See JDLRStack above.	
JDLRPage	Locate the RPAGE command (if there is one) and work backwards. See JDLRStack above.	

Changing the FSISYS INI File

NOTE: Two sets of resources are included on the installation tape. The RPEX1 set of resources is designed for the insurance market. The UTEX1 resources are designed for the utility market. You can verify your installation by running the system using one or both of these sets of resources.

The following instructions may ask you to modify the FSISYS initialization member of DEFLIB, so the DEFLIB you must modify is the one for the set of resources you plan to use, such as *hlq.v103.RPEX1.DEFLIB* or *hlq.v103.UTEX1.DEFLIB*, or both if you plan to use both.

Metacode data sets on z/OS should be created with a Variable Blocked Machine character (VBM) Record Format (RECFM) and should have a Logical Record Length (LRECL) that is four bytes longer than the length of the longest Metacode record you plan to write.

z/OS uses four bytes of each record in a Variable Blocked data set to store data about that record. For example, in your INI file you may have the JDLData option set to

```
0,250
```

which indicates the longest Metacode record will be 250 bytes long. If this is the case, your Metacode data set should be allocated with an LRECL of 254.

PrtType:XER control group

You must place the values you gathered from or added to the Xerox printer's JSL file into the PrtType:XER control group in the FSISYS INI file. This file is generally stored as a member called *FSISYS* within the sample DEFLIB PDS.

Below are the options in the PrtType:XER control group relevant to z/OS environments. Using the table you completed earlier, make changes as necessary to the following values. The example below shows the default FSISYS values.

NOTE: You can comment out an INI control group or option by placing a semicolon (;) before the control group or option.

```
< PrtType:XER >
    ColorCharts         = Yes
    CompileInstream     = No
    DownLoadFonts       = No,Disabled
    InitFunc            = XERInit
    Module              = XERW32
    OutputFunc          = XEROutput
    OutMetFunc          = XEROutMet
    PageNumbers         = Yes
    PrintFunc           = XERPrint
    PrinterInk          = Blue
    TermFunc            = XERTerm
    SendColor           = Yes
    DJDEIden            = A'@@@DJDE'
    DJDEOffset          = 0
    DJDESkip            = 8
    Environment         = MVS
    OutMode             = JES2
    ImageOpt            = Yes
    JDEName             = DFLT
    JDLData             = 0,255
    JDLHost             = IBMONL
    JDLName             = DFAULT
    JDLCode             = NONE
    JDLRStack           = 0,10,EQ,X'13131313131313131313'
    JDLRPage            = 0,5,EQ,X'FFFF26FFFF'Printer control group
```

The Printer control group tells the GenPrint program what type of output to produce. For the RPEX1 sample resources, the PrtType option is set to *XER*. For the UTEX1 sample resources, the PrtType option is set to *AFP*. Make sure the option is set correctly for your environment. For example, this setting tells the GenPrint program to produce Xerox Metacode print streams:

```
< Printer >
    ;PrtType = AFP
    PrtType = XER
```

Printer1 control group

The Printer1 control group contains PORT options which specify a DD name to which the system writes output for that recipient batch.

```
< Printer1 >
    PORT = DD:PRTBAT1
```

There may be other options present but they will be commented out with a semi-colon. Make sure the PORT option is set appropriately for your environment.

Pre-compiling FAP Files into Metacode Files

During the GenPrint process, FAP files (images) are compiled into Metacode files and arranged into the print stream. To speed execution, you can pre-compile—compile before running GenPrint—these FAP files.

NOTE: The FSISYS file included with the sample resources assumes you will be using pre-compiled Metacode files, or *PMETs*, since the use of PMETs increases performance. If, however, you *do not* want to run with PMETs, change the CompileInStream option in the PrtType:XER control group and DownloadFAP INI option in the RunMode control group to *Yes* and skip to the topic Making Printer Resources Available on page 48.

You can use the FAP2MET utility to compile FAP files into Metacode files. To run the system using pre-compiled Metacode files you must run the FAP2MET utility for every FAP file in FORMLIB. Here's how:

1. Run the FAP2METS job in JCLLIB. This job deletes or re-allocates the PMETLIB PDS and runs the FAP2MET utility for every FAP file in FORMLIB or FAPLIB. This creates a corresponding *PMET* and stores it in the PMETLIB PDS. The FAP2MET job should receive a return code of 00 for each step.
2. Make sure the CompileInstream option in the PrtType:XER control group is set to *No* and that the DownloadFAP option in the RunMode control group is set to *No*.
3. Modify the JCL for the DAP procedure or the GenPrtX job (please first read the topic Changing the JCL on page 49) so the PMETLIB DD statement is uncommented (it is included in the JCL but is commented out). This lets the GenPrint program open and read the PMETs you created.

Making Printer Resources Available

The resources your Xerox Metacode printer needs fall into two categories:

- Fonts
- Logos

The following topics tell you how to make these resources available to your printer.

Fonts

By default, the verification test (see Checking Your Installation on page 67) uses the AGFA fonts included with the system. If you installed the system from tape, the fonts were loaded into datasets called &HLQ..AGFA.XEROX.FONTLIB and &HLQ..AGFA.XEROX.ROTATED.FONTLIB when you ran the LOADDAP job. If that's the case, go to the topic, Loading fonts onto your printer on page 48.

Loading fonts onto your printer

To load the AGFA fonts onto a Xerox printer, follow these steps:

NOTE: The names of the Xerox fonts begin with the characters *FX*. You may want to list the existing fonts on the printer to see whether any of your existing fonts have the same names as the fonts you are about to send to the printer.

1. Run the XERDNLDS member in JCLLIB. This job uses the FSISYS file you customized earlier. It converts the fonts into a format your Xerox printer can store. XERDNLDS should complete all steps with a return code of *oo*.
2. Run the GENDNLDS member in JCLLIB. This job runs the IBM utility IEBGENER to copy each font to the Xerox printer. Modify GENDNLDS as necessary to fit your standards. Be sure to set the SYSOUT output class to the JES2 output class for the Xerox printer. GENDNLDS should complete all steps with a return code of *oo*.

Logos

The verification test references FAP files which contain logos (bitmap-type graphics). You must place these logos on the Xerox printer either as FNT files or as IMG files.

If your Xerox printer has a graphics card, it can print IMG files and you should place the LOGOs on the printer as IMG files. If your printer does not have a graphics card, it cannot print IMG files—place the logos on the printer as FNT files.

When you installed the system, two PDSs were created called *&HLQ..LOG2IMG* and *&HLQ..LOG2XFNT*. These PDSs contain logos which have been converted into IMG and FNT files, respectively. You should download the contents of one of these PDSs to your Xerox printer.

For steps 1 and 2, choose either to load the IMG files (choose DD statement for LOG2IMG) or the FNT files (choose DD statement LOG2XFNT) to your Xerox printer:

1. Run the XERLOGOS job in JCLLIB. Edit the job to read from and write to the appropriate type of file (LOG2IMG or LOG2XFNT). This job should complete with return codes of *oo* in all steps.
2. Run the GENLOGOS job in JCLLIB. Edit the job so it will read from the dataset you created in step 1 and so it will send the output to the correct JES sysout class. This makes sure the Xerox resources are sent to the Xerox printer. This job should complete with a completion code of *oo* in all steps.

Changing the JCL

DAP procedure (or the GenPrtX job)

In the next topic you will run the system with the sample resources to make sure it has been installed and configured correctly. You must modify the jobs you will run to conform to your local JCL and environment standards. The last job, GenPrint, will reference the DD names you set up in the topic, Printer1 control group on page 47.

When you run the system with the sample resources, you can run each step as a separate job (GenTrnX, GenDataX, and GenPrintX) or in sequence together by running a job called DAPRUN. The DAPRUN job invokes the DAP procedure which executes GenTrn, GenData, and GenPrint in this order.

For the verification test, you should use the DAPRUN job and edit the DAP procedure in PROCLIB. If, however, you want to run each step separately, edit the GenPrtX member in JCLLIB.

GENERXER job

You use the GENERXER job in JCLLIB to send the print streams created by the GenPrint program to the JES spool to be printed by your Xerox printer. Modify this job as necessary to match the file names you have used.

Creating AFP Print Streams

Configuring your system to produce AFP print streams involves these tasks:

- Allocating AFP-related datasets
- Making printer resources available.
- Pre-compiling FAP files into overlays (this task is optional)
- Making changes to the FSISYS initialization file
- Making changes to the JCL

Allocating AFP-related Datasets

To allocate the datasets required to perform AFP printing, use the ALLOCAFP job in the JCLLIB member. This job allocates the Form Definition Library (FDEFLIB), the Page Segment Library (PSEGLIB), and the Overlay Library (OVERLIB). The job should complete with a return code of oo.

Making Printer Resources Available

The resources your IBM AFP printer needs fall into these categories:

- Fonts
- Page segments
- FormDefs
- Overlays

The following topics tell you how to make these resources available to your printer.

Fonts

By default, the verification test (see Checking Your Installation on page 67) uses the Agfa fonts included with the system. If you will print using IBM's Advanced Function Presentation (AFP), provided by the IBM Print Services Facility (PSF), you need to use these Agfa fonts.

The installation tape includes the Agfa 240-dpi fonts. If you want the full compliment of Agfa fonts, which includes 300-dpi fonts, contact Support Services and ask for the Agfa Font CD.

Page segments

Logo Manager generates bitmap-type graphics into *logo* files which, on the PC, have an extension of *.LOG. The verification test uses images that contain these logos. For these images to print correctly, you must convert these logos into AFP Page Segments (PSEGs). You can run the LOG2PSEG utility on either the workstation or the z/OS system to perform this task.

To convert logos for the verification test, run the LOG2PSGS job in JCLLIB. This job converts several logos from the sample FORMLIB or FAPLIB into page segments and places them into a PSEGLIB. The job should complete all steps with a return code of *00*.

FormDefs

Next, submit the AFPFMDFX job in JCLLIB to compile a *source* Form Definition File into an AFP-ready FormDef. This job should complete with a return code of *00*.

Overlays

To speed performance in the GenPrint process, you can pre-compile FAP files (images) into AFP overlays before you run the GenPrint program. Follow the directions in the next topic, Pre-compiling FAP Files into AFP Overlays, to create the AFP overlays.

> **NOTE:** The FSISYS file included with the sample resources assumes you will be using AFP overlays, since the use of overlays increases performance. If, however, you do not want to run with overlays, change the SendOverlays option to *No* and the DownLoadFAP option to *Yes* and skip to the topic Changing the FSISYS INI File on page 50.

Pre-compiling FAP Files into AFP Overlays

The FAP2OVL utility pre-compiles FAP files into AFP overlays, which can improve performance. To pre-compile FAP files into AFP overlays, run the FAP2OVLS job in JCLLIB. This job converts the FAP files in the sample FAPLIB PDS into overlays and places them into the OVERLIB PDS.

Changing the FSISYS INI File

PrtType:AFP control group

Here are the options in the PrtType:AFP control group in the FSISYS file which apply to z/OS environments. The FSISYS INI file should have been supplied with the value below:

```
< PrtType:AFP >
    SendOverlays=Yes
```

If you chose not to create AFP overlays (in the prior topic above), set the SendOverlays option to *No*. If you created overlays, make sure the option is set to *Yes*.

Printer control group

The Printer control group tells the GenPrint program what type of output to produce. For the RPEX1 sample resources, the PrtType option is set to *XER*. For the UTEX1 sample resources, the PrtType option is set to *AFP*. Make sure the option is set correctly for your environment. For example, this setting tells the GenPrint program to produce AFP print streams:

```
< Printer >
    PrtType = AFP
    ;PrtType = XER
```

Printer1 control group

The Printer1 control group contains PORT options which specify a DD name to which the system writes output for that recipient batch.

```
< Printer1 >
    PORT = DD:PRTBAT1
```

There may be other options present but they will be commented out with a semi-colon. Make sure the PORT option is set appropriately for your environment.

Changing the JCL

Using the DAP procedure (or the GenPrtX job)

In the topic, Checking Your Installation on page 67, you will run the system using the sample resources. This will help you make sure your system is installed and configured correctly. Before you run the test, you must modify the jobs you will use during the test to match your local JCL and environment standards. The last job, GenPrint, references the DD names you may have added or uncommented (see PrtType:AFP control group on page 50).

When you run the system with the sample resources, you can run each step as a separate job (GenTrnX, GenDataX, and GenPrintX) or in sequence together by running the DAPRUN job. The DAPRUN job invokes the DAP procedure which executes GenTrn, GenData, and GenPrint in this order.

For the verification test, we recommend you use the DAPRUN job. If you choose to use the DAPRUN job, edit the DAP procedure in PROCLIB. If, however, you want to run each step separately, edit the GenPrtX member in JCLLIB.

NOTE: The DAP procedure included in PROCLIB (or the GenPrtX member of JCLLIB) for the UTEX1 resources is set up to produce AFP output file names such as AFPBAT1, AFPBAT2, and so on. You may want to inspect the procedure to make sure these names are correct, but you should not have to change the names.

GENERAFP job

You will use the GENERAFP job in JCLLIB to send the print streams created by the GenPrint program to the JES spool to be printed by your AFP printer. Modify this job as necessary to match the file names you have used.

Creating PostScript Print Streams

Configuring the system to produce PostScript print streams involves these tasks:

- Making changes to the FSISYS INI file
- Making changes to the JCL
- Printing the PostScript print stream

Changing the FSISYS INI File

Here are the options in the PrtType:PST control group in the FSISYS INI file that are relevant to z/OS environments:

```
< PrtType:PST >
    Module          = PSTW32
    PageNumbers     = Yes
    PrintFunc       = PSTPrint
;   PrinterModel    = (PPD file)
    SendOverlays    = Yes
    SendColor       = Yes
    DownloadFonts   = Yes
```

If you are using a PostScript Printer Description (PPD) file, upload your PPD file (as text) to DEFLIB, uncomment the PrinterModel option by removing the semicolon (;), and set it to your PPD file in DEFLIB. For example, if your PPD file in DEFLIB is HP8000, set your option as shown here:

```
PrinterModel = HP8000
```

Setting the SendOverlays option to Yes tells the GenPrint program to use PostScript overlays to speed its process. To create PostScript overlays, first run the OVLCOMP utility to convert your FAP files into PostScript overlays (the OVLCOMP utility only runs on Windows). You then upload (as binary) the PostScript overlays to the dataset referred to by OVERLIB DD in your JCL.

The Printer control group tells the GenPrint program what type of output to produce. For the RPEX1 sample resources, the PrtType option is set to XER. For the UTEX1 sample resources, the PrtType option is set to AFP. Make sure the option is set correctly for your environment. For example, this setting tells the GenPrint program to produce PostScript print streams:

```
< Printer >
;   PrtType                 = AFP
    PrtType                 = PST
;   PrtType                 = XER
;   PrtType                 = PDF
;   EnableTransBanner       = Yes
;   EnableBatchBanner       = Yes
;   BatchBannerBeginScript= PreBatch
;   TransBannerBeginScript= PreTrans
;   BatchBannerEndScript    = PstBatch
;   TransBannerEndScript    = PstTrans
```

The Printer1 control group contains PORT options which specify a DD name to which the system writes output for that recipient batch.

```
< Printer1 >
    Port = DD:PRTBAT1
```

There may be other options commented out with semicolons (;). Make sure the PORT option is set appropriately for your environment.

Changing the JCL

In the next topic, Checking Your Installation on page 67, you run the system with sample resources to make sure it has been installed and configured correctly. You must modify the jobs you will run to conform to your local JCL and environment standards. The last job, GenPrint, references the DD names you set up in the Printer1 control group.

When you run the system with the sample resources, you can run each step as a separate job (GenTrnX, GenDataX, and GenPrtX) or in sequence together by running a job called DAPRUN. The DAPRUN job invokes the DAP procedure which executes GenTrn, GenData, and GenPrint in this order.

For the verification test, use the DAPRUN job and edit the DAP procedure in PROCLIB. If, however, you want to run each step separately, edit the GenPrtX member in JCLLIB.

Printing PostScript Print Streams

To print PostScript print streams to a PostScript-capable printer, download (as binary) the print streams created by the GenPrint program to your Windows environment and do a binary copy (see the command below) of the downloaded file to your PostScript printer.

```
>copy /b your_file your_printer
```

Creating PDF Files

Note: Creating PDF files from a z/OS installation requires additional licensing. Contact your sales representative for more information.

Configuring the system to produce PDF files involves these tasks:

- Making changes to the FSISYS INI file
- Making changes to the JCL
- Creating the PDF file

Changing the FSISYS INI File

In the FSISYS INI file, below are the options in the PrtType:PDF control group relevant to z/OS environments. The FSISYS INI file includes the values shown here:

```
< PrtType:PDF >
;   Device          = E:\TEST.PDF
    Bookmark        = Yes,Page
    DownloadFonts   = Yes
    LanguageLevel   = Level1
    Module          = PDFW32
    PageNumbers     = Yes
    PrintFunc       = PDFPrint
    SendOverlays    = No
    SendColor       = Yes
    Encrypt         = No
    SecurityGroup   = PDFEncryption
;   PrintViewOnly   = No
;   SplitText       = No
;   SplitPercent    = 50
;   Class           = PDF
;   PrePrintedPaper= (not supported)
;   PaperSize       = 0
    Compression     = 0
    FontCompression= 0
< PDFEncryption >
    KeyLength       = 128
    User            = M
    Owner           = Oracle
    AllowModify     = FALSE
```

To produce secured PDF print streams, set the Encrypt option to Yes. Under the PDFEncryption control group, set the KeyLength, User, and Owner options as appropriate for your implementation.

The Printer control group tells the GenPrint program what type of output to produce. For the RPEX1 sample resources, the PrtType option is set to XER. For the UTEX1 sample resources, the PrtType option is set to AFP. Make sure the option is set correctly for your environment.

Generating PDF files using DAL scripts

For example, these options and settings tell the GenPrint program to produce a PDF file for each transaction in your print stream:

```
< Printer >
;   PrtType                = AFP
;   PrtType                = PST
;   PrtType                = XER
    PrtType                = PDF
    EnableTransBanner      = Yes
    EnableBatchBanner      = Yes
    BatchBannerBeginScript= PreBatch
    TransBannerBeginScript= PreTrans
    BatchBannerEndScript   = PstBatch
    TransBannerEndScript   = PstTrans
```

Note that these INI settings require the use of a BANNER DAL library and a WRITE DAL script. The BANNER DAL library contains the PreBatch, PreTrans, PstBatch, and PstTrans DAL scripts specified in the INI file. The WRITE DAL script is called from PreTrans.

These DAL scripts split a recipient batch PDF file into separate PDF files for each transaction in the batch. You can see examples of the BANNER DAL library and the WRITE DAL script in the DEFLIB of RPEX1 sample resources.

You also need to add this option:

```
< DALLibraries >
    LIB = Banner
```

Generating PDF files using the MultiFilePrint rule

These options and settings show another way to produce a PDF file for each transaction in your print stream:

```
< Printer >
;   PrtType                = AFP
;   PrtType                = PST
;   PrtType                = XER
    PrtType                = PDF
;   EnableTransBanner      = Yes
;   EnableBatchBanner      = Yes
;   BatchBannerBeginScript= PreBatch
;   TransBannerBeginScript= PreTrans
;   BatchBannerEndScript   = PstBatch
;   TransBannerEndScript   = PstTrans
< Print >
    CallbackFunc           = MultiFilePrint
    MultiFileLog           = DD:PDFLOG
```

These INI settings tell the system to use the MultiFilePrint rule to split a recipient batch PDF file into separate PDF files, one for each transaction in the batch. For more information about the MultiFilePrint rule, see the Rules Reference.

NOTE: The MultiFileLog option is optional. Include it if you want the system to create a log of the PDF file names written to the PDFLOG DD statement in the JCL. You can see an example of the PDFLOG DD statement in the DAP member of PROCLIB.

The Printer1 control group contains Port options which specify the DD name to which the system writes output for that recipient batch. The first transaction in the batch is written to member BAT10000 (or BAT10001 if you are using the MultiFilePrint rule) of PDFLIB PDS. The next transaction is written to member BAT10001 (or BAT10002 if you are using the MultiFilePrint rule), and so on.

The member name can consist of any four characters followed by 0000 (or 0001 if you are using the MultiFilePrint rule) for the first transaction of each batch. You only need to specify this first member name for each recipient batch in the INI file. The BANNER DAL library and the WRITE DAL script generate the subsequent member names and increment the numbers for that particular batch.

```
< Printer1 >
    Port = DD:PDFLIB(BAT10000)
< Printer2 >
    Port = DD:PDFLIB(BAT20000)
```

If there are other options present, comment out those options with semicolons (;). Make sure the Port option is set appropriately for your environment.

Including PostScript and TrueType fonts

To include PostScript fonts or TrueType fonts or both in PDF print streams, include these INI options and settings:

```
< PrtType:PDF >
    DownloadFonts  = Yes
< MasterResource >
    FontLib        = DD:FONTLIB()
```

Add a FONTLIB DD statement into your JCL (for the GenPrint step) and point to the FONTLIB PDS that has PostScript fonts and the PDS that has TrueType fonts. The FONTLIB for both PostScript and TrueType can have a record format (RECFM) of FB and logical record length (LRECL) of 128. You can see an example of a FONTLIB DD statement in the DAP member of PROCLIB.

PostScript and TrueType fonts can be uploaded as binary into the FONTLIB PDS. Keep in mind that you should change underscores (_) in the font name to @ symbols when uploading. For example, *ALBB____.PFB* is uploaded as *ALBB@@@@* since z/OS can not have underscore in the member name.

Understanding the System

The PDF Print Driver produces linearized PDF files. Linearized PDF files support page-at-a-time downloading which makes it faster to view a PDF file over the Internet. The nature of PDF files requires each transaction be in a separate PDF file.

In PDF files, information is added at the beginning of the PDF file that tells Acrobat where every object (file offsets for page, text, bitmap, fonts, and so on) is located inside the PDF file. The only way to gather this information is to have the entire file stored in memory before writing it out.

In Documaker, because of the volume of transactions that may be processed in a batch, only a single transaction is loaded into memory at a time. The print driver is then called. That transaction is processed and then the next transaction is loaded into memory. The PDF Print Driver only has information about the current transaction in memory, not all of the transactions that comprise a batch.

If you do not split transactions into separate PDF files, you end up with multiple PDF files concatenated into a single file, but the beginning of the file only contains information (file offsets for page, text, bitmap, font, and so on) about the last transaction that was processed. Since the last transaction can look a lot like the first PDF file, Acrobat may be fooled into displaying the first transaction. It will not, however, display the other transactions included in the file.

Other printer languages that Documaker supports, such as PCL, AFP, PostScript, and Metacode, do not require information about all the objects in the file to be placed at the beginning of the file. So for those printer languages, Documaker can append transactions into a single print stream.

The RunMode control group in the FSISYS.INI file includes these options and values:

```
< RunMode >
    NAUnload          = Yes
    DownloadFAP       = No
    CompiledFAP       = Yes
    ChkDstLenExceeded = No
    LoadCordFAP       = No
;   MergedDDTs        = No
```

Be sure to set the DownloadFAP option to Yes when you are producing PDF files.

Changing the JCL

In the topic, Checking Your Installation on page 67, you will run the system with sample resources to make sure it has been installed and configured correctly. You must modify the jobs you will run to conform to your local JCL and environment standards. The last job, GenPrint, references the DD names you set up in the Printer1 control group.

When you run the system with the sample resources, you can run each step as a separate job (GenTrnX, GenDataX, and GenPrtX) or in sequence together by running a job called DAPRUN. The DAPRUN job invokes the DAP procedure which executes GenTrn, GenData, and GenPrint in this order.

For the verification test, use the DAPRUN job and edit the DAP procedure in PROCLIB. If, however, you want to run each step separately, edit the GenPrtX member in JCLLIB.

Creating the PDF Print Stream

To print PDF print streams to the printer, download (as binary) the print streams created by the GenPrint program to your Windows environment. Then use Adobe Reader to open the downloaded file and send it to your printer.

Configuring Host Archive

The system lets you archive form sets that have been created by the GenData program. Archiving is performed by running the GenArc program and can be configured to archive the form sets in one of several ways:

- To the standard DAP archive. For more information, see Configuring Host Archive Using DB2 on page 60.
- To Documanage. For more information, see Archiving to Documanage on page 64.
- To Stacked DPA files. For more information, see Archiving Using Stacked DPA on page 66.

Configuring Host Archive Using DB2

In addition to (or instead of) simply printing form sets, you can optionally archive your form sets and data using the GenArc program. The GenArc program on z/OS systems supports archival to IBM's DB2 database management system.

Configuring a host archive system involves performing these tasks:

- Creating DB2 objects
- Running BINDDB2 or the ALLOCDB2 and COMPDB2 jobs
- Modifying the FSISYS file
- Making JCL changes

Creating DB2 Objects

The GenArc program, as configured for the sample library, uses these DB2 tables:

Table	Description
APPIDX	An application index table which contains a row for every archived form set.
ARCHIVE	An archive table which contains the actual NAFILE and POLFILE data for each archived form set.
CATALOG	A table which stores unique catalog IDs which the system uses as it builds the ARCKEY used in the APPIDX and ARCHIVE tables.
RESTART	A table in which GenArc restart information is stored.

You should create these tables using the ARCSQLR Job for the RPEX1 resources and the ARCSQLU Job for the UTEX1 resources. Modify the Jobs to suit your site's DB2 naming conventions and submit. The job should receive a return code of *oo*.

Running the BINDDB2 (or COMPDB2) Jobs

If you have a standard license, run the BINDDB2 job in JCLLIB as described below. If you have a source code license, skip to the following topic, Running ALLOCDB2 and COMPDB2 on page 61.

The BINDDB2 Job performs DB2 binds for the DB2LIB and DB2LIB2 packages and the DAP110 plan. In the BINDDB2 job, change the DB2 subsystem, package names, and plan name to fit your standards. The BINDDB2 job should finish with return codes of *oo* in all steps. Proceed to Modifying the FSISYS INI File for Host Archive on page 61.

Running ALLOCDB2 and COMPDB2

The COMPDB2 job invokes the DB2 pre-compiler to pre-compile source files and place the resulting DBRMs into a PDS called DBRMLIB. If you installed Documaker from tape, this DBRMLIB dataset should exist, but if it is not available, you can allocate it by running the job ALLOCDB2. The ALLOCDB2 job should finish with a return code of *00*.

The GenArc program uses programs called DB2LIB and DB2LIB2 to make all SQL calls. These SQL calls are prepared dynamically and are not static SQL calls.

To prepare the GenArc program for execution, run the job in member COMPDB2 of JCLLIB. The COMPDB2 member uses the DSNHC procedure in PROCLIB. Essentially, the COMPDB2 job will...

- DB2- pre-compile the DB2LIB and DB2LIB2 programs
- Compile the DB2LIB, DB2LIB2, and the GenArc programs
- Bind DB2LIB and DB2LIB2 into packages
- Bind the DB2LIB and DB2LIB2 packages into a plan
- Link-edit GenArc into an executable program, placing it in LINKLIB.

Depending on your DB2 system file names and C compiler file names, you will probably have to modify these procedures to meet your needs. The COMPDB2 job should finish with return codes of *00* or *04* in all steps.

Modifying the FSISYS INI File for Host Archive

Just as the FSIUSER and FSISYS initialization files provide the GenTrn, GenData, and GenPrint programs with configuration information, they also provide configuration information to the GenArc program.

The following control group and option settings must exist in either the FSIUSER or FSISYS members of DEFLIB. The FSISYS file shipped with the base product contains the values shown below. You only need to change these options if you have modified the names of the DB2 objects, such as the database name, table name, and so on.

ArcRet control group

The ArcRet control group contains information about the file (or table) names used for archiving information. The values for table names in this group cannot exceed eight characters. To specify a table name with more than eight characters use the DB2_FileConvert control group to map the eight character name you enter here to a longer (up to 18 characters) name.

```
< ArcRet >
    AppIdxDfd   = APPIDX
    AppIdx      = APPIDX
    CARFile     = ARCHIVE
    CATALOG     = CATALOG
    RestartTable= RESTART
```

Archival control group

The Archival control group indicates whether archived information is stored via a DBMS. For DAP archive to DB2, specify:

```
< Archival >
    ArchiveMem = Yes
```

DBHandler:DB2 control group

The DBHandler:DB2 control group tells the GenArc program that DB2 will be used to access the tables used during the GenArc process.

The Database option identifies the name of the DB2 database that contains the tables used by the GenArc program. Database names can consist of up to eight characters.

The CreateTable option indicates whether the GenArc program can create the tables it needs or if the tables must exist before it begins.

Enter	Description
No	Do not allow the GenArc program to create the DB2 tables.
Yes	Allow the GenArc program to create the DB2 tables.

The CreateIndex option indicates whether indexes are to be created for the APPIDX DB2 table if the APPIDX (DFD) member of DEFLIB contains keys information.

Enter	Description
No	Do not allow the GenArc program to create the DB2 indexes.
Yes	Allow the GenArc program to create the DB2 indexes.

```
< DBHandler:DB2 >
    CreateTable = No
    CreateIndex = No
```

DB2_FileConvert control group

The DB2_FileConvert control group lets you use DB2 table names that are longer than eight characters. The option on the left (APPIDX) is the value associated with the corresponding option in the ArcRet control group on page 61. The value on the right (DAP110_APP_R1) is the name of the DB2 table as it exists in the DB2 subsystem. If your table names are not longer than eight characters, you do not need to make any entries in the DB2_FileConvert control group.

```
< DB2_FileConvert >
    APPIDX      = DAP110_APP_R1
    Archive     = DAP110_ARC_R1
    Catalog     = DAP110_CAT_R1
    Restart     = DAP110_RES_R1
```

DBTable control group

Use the DefaultTag option to specify the default tag for ODBC and DB2. This tag is used by the ORDER BY clause in the SQL database to sort records.

```
< DBTable:MYTABLE >
    DefaultTag =
```

For the DefaultTag option, enter the name of the key from the DFD file. Keep in mind this only works with ODBC and DB2. It does not work with xBase files.

Trigger2Archive control group

The Trigger2Archive control group designates the fields in the NEWTRN DAT file (right side) that are copied to the columns in the APPIDX table (left side) during the GenArc process. Remember that the APPIDX table columns are defined by the APPIDX member of DEFLIB and that this APPIDX member is in a DFD format. For the sample resources, the Trigger2Archive control group might be set up as follows:

```
< Trigger2Archive >
    Company     = Company
    LOB         = Lob
    PolicyNum   = PolicyNum
    RunDate     = RunDate
```

Changing the JCL for Host Archive

Examine the GenArcDX member of JCLLIB. Make sure the JCL is appropriate for your site and that the dataset names correspond to the names you have used for your files. Additionally, make sure the DB2 subsystem name and other DB2 object names, such as the plan names, are correct.

ARCHIVING TO DOCUMANAGE

Configuring the system to archive to Documanage involves performing these tasks:

- Setting up Documanage (see the Documanage manuals for more information)
- Modifying the FSISYS file (FSISYSM member in DEFLIB)
- Adding IP addresses and system/domain names of the Documanage machine to the TCPIP host data sets
- Making JCL changes

Modifying the FSISYS File

NOTE: All references of the FSISYS file in this topic refer to FSISYSM member in DEFLIB.

In the ArcRet control group, the FSISYS file includes the values shown here:

```
< ArcRet >
    AppIdxDfd       = AppIdx
    AppIdx          = APPIDX
    CARFile         = ARCHIVE
    CARFileDFD      = CARFILE
    CARPath         =
;   Catalog         = CATALOG
;   RestartTable    = RESTART
```

For archiving to Documanage, you need to set the CARFileDFD option to *CARFILE* (which resides in DEFLIB) and comment out both the Catalog and RestartTable options.

The Archival control group in the FSISYS file should have these values:

```
< Archival >
    ArchiveMem      = Yes
    UseRestartTable= No
```

For archiving to Documanage, set the UseRestartTable option to No.

In the following DBTable control groups, the FSISYS file should have these values:

```
< DBTable:APPIDX >
     DBHandler = DMIA
< DBTable:ARCHIVE >
     DBHandler = DMIA
```

NOTE: You must use *DMIA* as a DBHandler for archiving to Documanage.

The DBHandler:DMIA control group tells the GenArc program that DMIA is used to access Documanage during the GenArc process. Your FSISYS file includes these values:

```
< DBHandler:DMIA >
    RPCHost     = (IP address of the Documanage machine)
    Cabinet     = RPEX1OS390
    Domain      = (system/domain name)
    UserID      = (Documanage user ID)
    Password    = (Documanage password)
;   Debug       = Max
```

You must add the IP address, system/domain name, user ID, and password to the appropriate INI options.

NOTE: For whichever cabinet name is being set in the DBHandler:DMIA control group, you must include that name in the DMIA control group name. RPEX1OS390 is the cabinet name in this case. So, the DMIA control group name would be *DMIA:RPEX1OS390*.

For the DMIA:RPEX1OS390 control group, the FSISYS file includes these values:

```
< DMIA:RPEX1OS390 >
    FileType    = DPA
    FolderBy    = company,lob,policynum
    NameDocBy   = arckey
    StackedDPA  = Yes
```

Set the StackedDPA option to No when archiving to Documanage.

Adding IP Addresses and System/Domain Names

You will need to add the IP address and system/domain name of your Documanage machine to the local TCPIP host data sets on your z/OS. The steps below are examples you could follow to accomplish this task.

NOTE: You first may need to consult your z/OS system programmer on how to get this task done as the steps you need to do may differ from below.

1. Add an entry with your IP and system/domain name to hlq.HOSTS.LOCAL.
2. Execute a TSO MAKESITE command to generate hlq.HOSTS.SITEINFO and hlq.HOSTS.ADDRINFO datasets from your hlq.HOSTS.LOCAL.

Changing the JCL

Examine the GenArcX member of JCLLIB. Make sure the JCL is appropriate for your site and that the dataset names correspond to the names you have used for your files.

ARCHIVING USING STACKED DPA

Configuring the system to use stacked DPA to archive to a CARDATA file involves performing these tasks:

- Checking the FSISYS file (FSISYSM member of DEFLIB)
- Making JCL changes

For the DBHandler:DMIA control group below, you do not need to add IP address, system/domain name, user ID, or password to run GenArc with stacked DPA.

```
< DBHandler:DMIA >
    RPCHost     = (IP address of the Documanage machine)
    Cabinet     = RPEX1OS390
    Domain      = (system/domain name)
    UserID      = (Documanage user ID)
    Password    = (Documanage password)
;   Debug       = Max
```

In the DMIA:RPEX1OS390 control group, make sure the StackedDPA option is set to Yes.

```
< DMIA:RPEX1OS390 >
    FileType    = DPA
    FolderBy    = company,lob,policynum
    NameDocBy   = arckey
    StackedDPA  = Yes
```

NOTE: All other INI options for archiving to Documanage are the same for stacked DPA.

Changing the JCL

Examine the GenArcX member of JCLLIB. Make sure the JCL is appropriate for your site and that the data set names correspond to the names you have used for your files.

Checking Your Installation

To make sure the base system has been installed correctly you should:

- Install DMS1 test resources.
- Check the INI files (FSIUSER, FSISYS, and FSIUSERL).
- Check the JCL for the DAPRUN and DAP procedure (or GentrnX, GenDataX, GenPrtX), and the DMRUNLM and DMLM procedure.
- Run DAPRUN (or GenTrnX, GenDataX, GenPrtX) and DMRUNLM.

Note: Two sets of resources are included on the installation tape and are available from the COSS dowload site. The RPEX1 resources are designed for the insurance market and the UTEX1 resources are designed for the utility market. You can verify your installation by running the system using one or both of these sets of resources. When you run the DAPRUN job later in this guide, you will need to specify as part of the jobcard, which set of resources the DAPRUN job is using.

You indicate the resource set by specifying the "*// SET RES=xxxxx*" statement, where *xxxxx* is either *RPEX1* or *UTEX1*. By default, the JOBCARD is set up to use the RPEX1 resources.

The DMS1 resources are the RPEX1 resources in Documaker Studio format. These resources are stored in DB2 library. Use the DMRUNLM job to run with DMS1 resources. Set the JOBCARD to use DMS1 resources.

Checking Your FSIUSER and FSISYS Settings

You loaded the FSIUSER and FSISYS members from the tape into the DEFLIB PDS during installation. These two files have similar control groups and options but different purposes. Generally, the FSIUSER member contains customer-specific initialization options while the FSISYS member contains general system-wide options.

Among other things, the FSIUSER and FSISYS members contain the names of various input and output files. On the PC, these file names include the path and file name. On z/OS, these names consist of DD statement names or PDS member names or both.

The FSIUSER and FSISYS you uploaded as part of the sample resources are in a format you can run on z/OS. To verify these values, consider the following but note that this list of options is not complete.

The relevant value in the FSIUSER INI file is...

```
< Environment >
   FSISYSINI   = DD:FSISYS
```

The relevant values in FSISYS that indicate the DD names of various libraries, the PDS members, or are required for the system to run are:

```
< BATCH1 >
    Printer = Printer1
< BATCH2 >
    Printer = Printer2
< BATCH3 >
    Printer = Printer3
```

```
< Manual >
    Printer = Pmanual
< Error >
    Printer = Perror
< CONFIG:Batch Processing >
    CompLib    = DD:COMPLIB()
   DDTLib     = DD:DDTLIB()
    DefLib     = DD:DEFLIB()
    FormDef    = FORM
    FormLib    = DD:FAPLIB()
    XRFFile    = REL95SM
< Control >
    ImageExt   = .FAP
    LogoExt    = .LOG
    XrfExt     = .FXR
< Data >
    DataPath        =
    ExtrFile        = DD:EXTRACT
    DbLogFile       = DD:DBLOGFLE
    ErrFile         = DD:ERRFILE
    LogFile         = DD:LOGFILE
    MsgFile         = DD:MSGFILE
    NAFile          = DD:NAFILE
    NewTrn          = DD:NEWTRN
    PolFile         = DD:POLFILE
    TrnFile         = DD:TRNFILE
    WIPBatch        = DD:MANUAL
    Manual          = DD:MANUAL
    TrnDfdFile      = TrnDfdFl
    AfgJobFile      = DD:AFGJOB
    RcbDfdFile      = RcbDfdFl
    SetRcpTb        = SetRcpTb
    TextTbl         = TextTbl
    TblFile         = TblFile
< Print_Batches >
    Batch1 = DD:BATCH1
    Batch2 = DD:BATCH2
    Batch3 = DD:BATCH3
    Manual = DD:MANUAL
    Error  = DD:ERROR
< Printer >
    PrtType = AFP
    ;PrtType = XER
< Printer1 >
    PORT = DD:PRTBAT1
< Printer2 >
    PORT = DD:PRTBAT2
< Printer3 >
    PORT = DD:PRTBAT3
< PManual >
    PORT = DD:PMANUAL
< PError >
    PORT = DD:PERROR
< PrinterInfo >
    Printer = Printer1
```

```
    Printer = Printer2
    Printer = Printer3
    Printer = PManual
    Printer        = PError
< PrtType:XER >
    ColorCharts        = Yes
    CompileInstream    = No
    DownLoadFonts      = No,Disabled
    InitFunc           = XERInit
    Module             = XERW32
    OutputFunc         = XEROutput
    OutMetFunc         = XEROutMet
    PageNumbers        = Yes
    PrintFunc          = XERPrint
    PrinterInk         = Blue
    TermFunc           = XERTerm
    SendColor          = Yes
    DJDEIden           = A'@@@DJDE'
    DJDEOffset         = 0
    DJDESkip           = 8
    Environment        = MVS
    OutMode            = JES2
    ImageOpt           = Yes
    JDEName            = DFLT
    JDLData            = 0,255
    JDLHost            = IBMONL
    JDLName            = DFAULT
    JDLCode            = NONE
    JDLRStack          = 0,10,EQ,X'13131313131313131313'
    JDLRPage           = 0,5,EQ,X'FFFF26FFFF'Printer control group
< PrtType:AFP >
    SendOverlays       =  Yes
< TRN_File >
    MaxExtRecLen     = 119(for RPEX1 only)
    MaxExtRecLen       = 450  (for UTEX1 only)
    BinaryExt          = N
```

Checking the JCL

As stated in the topic, Configuring Your Printer Type on page 43, when you run the system with the RPEX1 and UTEX1 sample resources you can run each step as a separate job (GenTrnX, GenDataX, GenPrintX) or together in sequence using the DAPRUN job which invokes the DAP procedure in PROCLIB.

For the verification test, we recommend you use the DAPRUN job. If you want to use the DAPRUN job and DAP procedure, inspect and modify those files. If you want to run GenTrnX, GenDataX, and GenPrtX separately, inspect and modify those files.

If you plan to use the DAPPRUN job...

1. Check the DAPRUN job in JCLLIB. This job invokes the DAP procedure. Make sure DAPRUN has a valid jobcard and that the PROCLIB name is correct.
2. Check the DAP procedure in PROCLIB. This procedure contains these steps:

	Description
GENTRND	Deletes files in preparation for the GenTrn program
GENTRN	Runs the GenTrn program
GENDATAD	Deletes files in preparation for the GenData program
GENDATA	Runs the GenData program
GENPRTD	Deletes files in preparation for the GenPrint program
GENPRT	Runs the GenPrint program

3. Check the JCL statements in the DAP procedure and change the dataset names to match the names you chose during installation. Also, change any UNIT names to conform to your site's standards.

   Remember that the DD names for various input files (DEFLIB, FAPLIB, TRNFILE, and so on) and output files (BATCH1, NAFILE, and so on) need to correspond to the values that point to these files in the FSISYS file.

   You should not need to modify the actual DD names for these files either in the DAP procedure nor in the FSISYS member because these files have been set up to run the verification test as is.

Running GenTrnX, GenDataX, and GenPrtX separately

Look at the GenTrnX, GenDataX and GenPrtX jobs in JCLLIB and modify any jobcards, dataset names, unit names, and so on, so these jobs will run on your system.

Remember that the DD names for various input files (DEFLIB, FAPLIB, TRNFILE, and so on) and output files (BATCH1, NAFILE, and so on) should correspond to the values that point to these files in the FSISYS member. You should not need to modify the actual DD names for these files either in the JCL jobs or in the FSISYS member because these files are set up to run the verification test as is.

Running DAPRUN (or GenTrnX, GenDataX, and GenPrtX) and DMRUNLM

Submit the job DAPRUN (or DMRUNLM). DAPRUN (or DMRUNLM) should receive return codes of *oo* or *04* in each step. The output for DAPRUN (or DMRUNLM) should include logging messages, which are sent to SYSPRINT. Depending on whether you are using the RPEX1 or UTEX1 sample resources, the messages will differ. The messages for DMS1 are similar to those for RPEX1. Examples are shown below and on the following pages.

NOTE: If you are running GenTrnX, GenDataX, and GenPrtX separately you should submit each job in sequence and check the return codes.

Running DAPRUNC to Improve Performance

The DAPRUNC job invokes the DAPC procedure in PROCLIB. The DAPC procedure contains JCL that assumes the GenTrn and GenPrint steps have been combined into the GenData step. The GenTrn and GenPrint steps are run as rules within the GenData step. This is called *single-step processing* and can improve system performance. To use single-step processing, submit DAPRUNC. It should receive a return code of *oo* or *04* in both steps.

NOTE: When using single-step processing, only the SYSOUT messages for GenData are displayed.

Logging messages when running RPEX1

```
--- GenTrn ---
Filtering extract file...
Building transaction file...
==> Processing: TransactionId-GroupName1-GroupName2-GroupName3-
TransactionType
==> Processing: 1234567-SAMPCO-LB1--T1
==> Processing: 2234567-SAMPCO-LB1--T1
==> Processing: 5SAMPCO-SAMPCO-LB2--T1
==> Processing: 6SAMPCO-SAMPCO-LB2--T1
==> Processing: 7SAMPCO-SAMPCO-LB2--T1
==> Processing: 8SAMPCO-SAMPCO-LB2--T1
==> Processing: 9SAMPCO-SAMPCO-LB2--T1
==> Processing: 4234567-FSI-CPP--T1
==> Processing: 5234567-FSI-GL--T1
==> Warning count:    0
==> Error   count:    0
Elapsed Time: 3 seconds
--- GenTrn Completed ---

--- GenData ---
==> Processing: TransactionId-GroupName1-GroupName2-GroupName3-
TransactionType
==> Processing: 1234567-SAMPCO-LB1--T1
==> Processing: 2234567-SAMPCO-LB1--T1
==> Processing: 5SAMPCO-SAMPCO-LB2--T1
```

```
==> Processing: 6SAMPCO-SAMPCO-LB2--T1
==> Processing: 7SAMPCO-SAMPCO-LB2--T1
==> Processing: 8SAMPCO-SAMPCO-LB2--T1
==> Processing: 9SAMPCO-SAMPCO-LB2--T1
==> Processing: 4234567-FSI-CPP--T1
==> Processing: 5234567-FSI-GL--T1
==> Warning count:    0
==> Error   count:    0
Elapsed Time: 44 seconds
--- GenData Completed ---

--- GenPrint ---
Processing Batch:<BATCH1> File:<DD:BATCH1> Port:<DD:XERBAT1>
==> Processing: TransactionId-GroupName1-GroupName2-GroupName3-
TransactionType
==> Processing: 1234567-SAMPCO-LB1--T1
==> Processing: 2234567-SAMPCO-LB1--T1
==> Processing: 5SAMPCO-SAMPCO-LB2--T1
==> Processing: 6SAMPCO-SAMPCO-LB2--T1
==> Processing: 7SAMPCO-SAMPCO-LB2--T1
==> Processing: 8SAMPCO-SAMPCO-LB2--T1
==> Processing: 9SAMPCO-SAMPCO-LB2--T1
==> Processing: 4234567-FSI-CPP--T1
==> Processing: 5234567-FSI-GL--T1
9 records processed in BATCH1 batch.
Processing Batch:<BATCH2> File:<DD:BATCH2> Port:<DD:XERBAT2>
==> Processing: 1234567-SAMPCO-LB1--T1
==> Processing: 2234567-SAMPCO-LB1--T1
==> Processing: 5SAMPCO-SAMPCO-LB2--T1
==> Processing: 6SAMPCO-SAMPCO-LB2--T1
==> Processing: 7SAMPCO-SAMPCO-LB2--T1
==> Processing: 8SAMPCO-SAMPCO-LB2--T1
==> Processing: 9SAMPCO-SAMPCO-LB2--T1
==> Processing: 4234567-FSI-CPP--T1
==> Processing: 5234567-FSI-GL--T1
9 records processed in BATCH2 batch.
Processing Batch:<BATCH3> File:<DD:BATCH3> Port:<DD:XERBAT3>
==> Processing: 1234567-SAMPCO-LB1--T1
==> Processing: 2234567-SAMPCO-LB1--T1
==> Processing: 5SAMPCO-SAMPCO-LB2--T1
==> Processing: 6SAMPCO-SAMPCO-LB2--T1
==> Processing: 7SAMPCO-SAMPCO-LB2--T1
==> Processing: 8SAMPCO-SAMPCO-LB2--T1
==> Processing: 9SAMPCO-SAMPCO-LB2--T1
7 records processed in BATCH3 batch.
Processing Batch:<ERROR> File:<DD:ERROR> Port:<DD:PERROR>
Processing Batch:<MANUAL> File:<DD:MANUAL> Port:<DD:PMANUAL>
==> Warning count:    3
==> Error   count:    0
Elapsed Time: 49 seconds
--- GenPrint Completed ---
```

Logging messages when running UTEX1

```
--- GenTrn ---
Filtering extract file...
Building transaction file...
==> Processing: TransactionId-GroupName1-GroupName2-GroupName3-
TransactionType
==> Processing: 2222-3445-UTIL-BILL--
==> Processing: 1234-5555-UTIL-BILL--
==> Processing: 3456-7777-UTIL-BILL--
==> Processing: 3999-9999-UTIL-BILL--
==> Processing: 4444-1234-UTIL-BILL--
==> Processing: 4999-R777-UTIL-BILL--
==> Processing: 5111-8888-UTIL-BILL--
==> Processing: 5555-7010-UTIL-BILL--
==> Processing: 6750-R758-UTIL-BILL--
==> Processing: 7000-7000-UTIL-BILL--
==> Processing: 2428-1927-UTIL-BILL--
==> Processing: 8234-5555-UTIL-BILL--
==> Warning count:    0
==> Error   count:    0
Elapsed Time: 3 seconds
--- GenTrn Completed ---

--- GenData ---
==> Processing: TransactionId-GroupName1-GroupName2-GroupName3-
TransactionType
==> Processing: 2222-3445-UTIL-BILL--
==> Processing: 1234-5555-UTIL-BILL--
==> Processing: 3456-7777-UTIL-BILL--
==> Processing: 3999-9999-UTIL-BILL--
==> Processing: 4444-1234-UTIL-BILL--
==> Processing: 4999-R777-UTIL-BILL--
==> Processing: 5111-8888-UTIL-BILL--
==> Processing: 5555-7010-UTIL-BILL--
==> Processing: 6750-R758-UTIL-BILL--
==> Processing: 7000-7000-UTIL-BILL--
==> Processing: 2428-1927-UTIL-BILL--
==> Processing: 8234-5555-UTIL-BILL--
==> Warning count:    0
==> Error   count:    0
Elapsed Time: 148 seconds
--- GenData Completed ---

--- GenPrint ---
Processing Batch:<BATCH1> File:<DD:BATCH1> Port:<DD:AFPBAT1>
==> Processing: TransactionId-GroupName1-GroupName2-GroupName3-
TransactionType
==> Processing: 2222-3445-UTIL-BILL--
==> Processing: 1234-5555-UTIL-BILL--
==> Processing: 3456-7777-UTIL-BILL--
==> Processing: 3999-9999-UTIL-BILL--
==> Processing: 4444-1234-UTIL-BILL--
==> Processing: 4999-R777-UTIL-BILL--
==> Processing: 5111-8888-UTIL-BILL--
==> Processing: 5555-7010-UTIL-BILL--
==> Processing: 6750-R758-UTIL-BILL--
==> Processing: 7000-7000-UTIL-BILL--
```

```
==> Processing: 2428-1927-UTIL-BILL--
==> Processing: 8234-5555-UTIL-BILL--
12 records processed in BATCH1 batch.
Processing Batch:<BATCH2> File:<DD:BATCH2> Port:<DD:AFPBAT2>
Processing Batch:<BATCH3> File:<DD:BATCH3> Port:<DD:AFPBAT3>
Processing Batch:<ERROR> File:<DD:ERROR> Port:<DD:PERROR>
Processing Batch:<MANUAL> File:<DD:MANUAL> Port:<DD:PMANUAL>
==> Warning count:    5
==> Error   count:    0
Elapsed Time: 75 seconds
--- GenPrint Completed ---
```

USING THE DMS1 RESOURCES

Follow these instructions to install DMS1 resources for z/OS. You can download the software and these resources from the Customer Online Support System (COSS) web site.

1 From the COSS Documaker installation web site, click the version of Documaker for z/OS you want to install. Then right click the following files and select Save Target As to save these files to your workstation:

- readme.txt
- loaddms1.jcl
- loaddb2.jcl
- DMS1.zip

NOTE: The readme.txt file contains much of the same information included in this topic.

2 Unzip the DMS1.zip file. It contains a file named *DMS1.xmit*. This file contains several z/OS datasets and is in a TRANSMIT (XMIT) format.

To install the DMS1 resources, you must transfer this file to z/OS in binary mode and then unzip (RECEIVE) it. The RECEIVE creates and populates the datasets that comprise the DMS1 test resources. Follow the remaining steps to transfer this file to z/OS and perform the RECEIVE.

3 Allocate a dataset on z/OS into which you can upload the DMS1.xmit file. Allocate this dataset using attributes similar to these:

```
Data Set Name . . . : FSI.DMS1.XMIT <= any name you choose

Organization  . . . : PS
Record format . . . : FB
Record length . . . : 80
Block size  . . . . : 3120
1st extent cylinders: 10      <= primary space needed
Secondary cylinders : 2       <= secondary space needed
```

4 Upload the DMS1.xmit file from Windows to z/OS. Be sure to upload this file in binary mode.

5 Upload LOADDMS1.JCL as text to your JCL library on z/OS. Read the comments in the JCL and make the necessary modifications for your system. Submit the LOADDMS1 job. It should create several datasets that contain the DMS1 sample resources.

NOTE: Make sure you have completed the steps in Binding the Documaker DB2 Plan to Your DB2 Subsystem on page 35 before you continue to the next step.

6 Use LBYSQLR member in the JCLLIB to create DB2 tables for the Documaker library. Make the necessary modifications, including changing table names for your system, before submitting.

NOTE: You can have all of the tables in one tablespace or you can place each table in a different tablespace.

7 Edit the FSI.LBYD.SYSPUNCH and FSI.LBYI.SYSPUNCH datasets, replacing the user ID *ZD* and the table names with the user ID and table names you used in the LBYSQLR member. Do not change anything else in this file. This file loads the DMS1 sample resources into your DB2 tables.

```
INTO TABLE "ZD"."DMS1_LBYD"  <= this line is in FSI.LBYD.SYSPUNCH
INTO TABLE "ZD"."DMS1_LBYI"  <= this line is in FSI.LBYI.SYSPUNCH
```

8 Upload the LOADDB2.JCL to your JCLLIB. Make the necessary modifications, including changing the name of the SYSPUNCH and SYSREC datasets for your system.

Submit LOADDB2 to load the tables. You will need to run this job twice — once with the SYSIN DD statement set to FSI.LBYD.SYSPUNCH and once with the SYSIN DD set to FSI.LBYI.SYSPUNCH.

NOTE: A certain level of DB2 authority may be needed to run this job.

Checking Your FSIUSERL Settings

The FSIUSERL is a member of the DEFLIB PDS in the DMS1 installation. Besides those options as described in FSIUSER and FSISYS, the FSIUSERL member contains initialization options to retrieve resources from DB2 library.

The following partial list of options is from the FSIUSERL member. The DB2 database is set with the Database INI option. The DB2 tables are set in the DB2_FileConvert control group.

Be sure to make the necessary changes for your implementation.

```
< CONFIG:DMS1 >
    ArcPath          =
    BaseDef          = DMS1
    BDFFile          = LBYI
    BDFLib           =
    CARPath          =
    DalFile          = LBYI
    DDTFile          = LBYI
    DefLib           = DD:DEFLIB()
    DictionaryFile   = xdb.dbf
    FieldBaseFile    = fdb.dbf
    FNTFile          = rel103sm
    FontLib          = DD:FONTLIB()
    FORFile          = LBYI
    FORLib           =
```

```
    FormDef          = FORM
    FormFile         = LBYI
    FormLib          =
    GRPFile          = LBYI
    GRPLib           =
    HelpLib          =
    LbyLib           = DD:DEFLIB()
    LogoFile         = LBYI
    TableLib         = DD:DEFLIB()
    WIPPath          =
    XDDFile          = LBYI
    XRFFile          = rel103sm
< CONFIGURATIONS >
    Config           = DMS1
< DBHANDLER:DB2 >
    Database    = LBYDB
    CreateTable = No
    CreateIndex = No
    Debug       = No
< DBTable:LBYI >
    DBHandler   = DB2
< DBTable:LBYD >
    DBHandler   = DB2
    UniqueTag   = ARCKEY+SEQ_NUM
< DBTable:LBYLog >
    DBHandler   = DB2
    UniqueTag   = DATE+TIME
< DBTable:Catalog >
    DBHandler   = DB2
    UniqueTag   = CatalogID
< DB2_FileConvert >
    LBYI        = ZD.DMS1_LBYI
    LBYD        = ZD.DMS1_LBYD
    LBYLOG      = ZD.DMS1_LBYLOG
    Catalog     = ZD.DMS1_CAT
< Library:LBYI >
    DBTable    = LBYD
    LBYLogFile = LBYLOG
< LibraryManager >
    Class           = GA;Georgia resource
    Class           = TX;Texas resource
    Class           = MD;Maryland resource
    Library         = LBYI
    Mode            = DEV;Development
    Mode            = TEST;Test
    Mode            = PROD;Production
    Project         = P001;Project 001
    Project         = P002;Project 002
    Project         = P003;Project 003
    Status          = TEST;Needs to be tested
    Status          = FAILED;Failed testing
    Status          = PASSED;Passed testing
    Status          = PROMOTED;Has been promoted
< MasterResource >
    BaseDef         = <CONFIG:DMS1> BaseDef =
```

```
BDFFile          = <CONFIG:DMS1> BDFFile =
BDFLib           = <CONFIG:DMS1> BDFLib =
COMPLIB          = <CONFIG:DMS1> CompLib =
DALFile          = <CONFIG:DMS1> DalFile =
DDTFile          = <CONFIG:DMS1> DDTFile =
DDTLib           = <CONFIG:DMS1> DDTLib =
DefLib           = <CONFIG:DMS1> DefLib =
DictionaryFile   = <CONFIG:DMS1> DictionaryFile =
FieldBaseFile    = <CONFIG:DMS1> FieldBaseFile =
FNTFile          = <CONFIG:DMS1> FntFile =
FontLib          = <CONFIG:DMS1> FontLib =
FORFile          = <CONFIG:DMS1> FORFile =
FORLib           = <CONFIG:DMS1> FORLib =
Form7x           = <CONFIG:DMS1> Form7x =
FormDef          = <CONFIG:DMS1> FormDef =
FormFile         = <CONFIG:DMS1> FormFile =
FormLib          = <CONFIG:DMS1> FormLib =
GRPFile          = <CONFIG:DMS1> GRPFile =
GRPLib           = <CONFIG:DMS1> GRPLib =
HelpLib          = <CONFIG:DMS1> HelpLib =
LbyLib           = <CONFIG:DMS1> LbyLib =
LogoFile         = <CONFIG:DMS1> LogoFile =
LogoLib          = <CONFIG:DMS1> LogoLib =
TableLib         = <CONFIG:DMS1> TableLib =
TablePath        = <MASTERRESOURCE> TableLib =
XDDFile          = <CONFIG:DMS1> XDDFile =
XRFFile          = <CONFIG:DMS1> XrfFile =
```

Checking the JCL

When you run the system with the DMS1 sample resources, use the DMRUNLM job which invokes the DMLM procedure in PROCLIB. Make sure DMRUNLM has a valid jobcard and that the PROCLIB name is correct.

Also in DMRUNLM job, check the override statement for GenTrn, GenData, and GenPrint to make sure you have the correct DB2 SUBSYSTEM, PLAN, and LINKLIB from your system installation.

See Binding the Documaker DB2 Plan to Your DB2 Subsystem on page 35 for more information on the DB2 SUBSYSTEM and PLAN. For a description of the DMLM procedure, see the description of the DAP procedure above.

Here is an example of the DMRUNLM job:

```
//*  COPY JOBCARD HERE
//
*********************************************************************
//*                          - DMRUNLM -
//*
//
*********************************************************************
//*
//DMRUNLM  EXEC DMLM
//GENTRN.SYSTSIN  DD  *
 DSN SYSTEM(DB8G)
 RUN  PROGRAM(GENTRN) PLAN(DAP113) -
      LIB('FSI.V113.LINKLIB')
```

```
 END
//*
//GENDATA.SYSTSIN  DD  *
 DSN SYSTEM(DB8G)
 RUN  PROGRAM(GENDATA) PLAN(DAP113) -
      LIB('FSI.V113.LINKLIB')
 END
//*
//GENPRT.SYSTSIN  DD  *
 DSN SYSTEM(DB8G)
 RUN  PROGRAM(GENPRINT) PLAN(DAP113) -
      LIB('FSI.V113.LINKLIB')
 END
//*
```

RUNNING GENARCDX

This task is optional.

The JCLLIB member GenArcDX (GenArcX prior to version 10.3) executes the GenArc program in two steps. The first step, called *GenArcD*, executes the IEFBR14 program and contains DD statements that delete various files which will be created in the following GenArc step. The second step, called *GenArc*, executes the TSO Terminal Monitor Program and starts the DB2 Command Processor program (DSN) which runs the GenArc program and DB2 plan.

At the bottom of the GenArcDX job, the SYSTSIN DD statement has some in-stream control cards, one of which indicates the LINKLIB you loaded from tape. Change &HLQ..LINKLIB to match the name you gave the LINKLIB you loaded from the tape. Also, you'll need to change the subsystem name from *TDB1* to match your DB2 subsystem name.

```
//SYSTSIN   DD  *
 DSN SYSTEM(TDB1)
 RUN  PROGRAM(GENARC) PLAN(DAP110) -
      LIB('&HLQ..LINKLIB')
 END
```

To test, submit the GenArcDX job. The GenArcDX job should receive a return code of *00* or *04*. The output for the GenArcDX job should include logging messages (sent to SYSPRINT) similar to the following messages.

RPEX1 messages

```
--- GenArc ---
==> Processing: TransactionId-GroupName1-GroupName2-GroupName3-
TransactionType
==> Processing: 1234567-SAMPCO-LB1--T1
==> Processing: 2234567-SAMPCO-LB1--T1
==> Processing: 5SAMPCO-SAMPCO-LB2--T1
==> Processing: 6SAMPCO-SAMPCO-LB2--T1
==> Processing: 7SAMPCO-SAMPCO-LB2--T1
==> Processing: 8SAMPCO-SAMPCO-LB2--T1
==> Processing: 9SAMPCO-SAMPCO-LB2--T1
==> Processing: 4234567-FSI-CPP--T1
==> Processing: 5234567-FSI-GL--T1

==> Transactions Read       :      9
==> Transactions Archived   :      9
==> Transactions In Error   :      0
==> Transactions Rolled Back:      0

==> Warning count:    1
==> Error   count:    0

Elapsed Time: 44 seconds
--- GenArc Completed ---
```

UTEX1 messages

```
--- GenArc ---

==> Processing: TransactionId-GroupName1-GroupName2-GroupName3-
TransactionType
==> Processing: 2222-3445-UTIL-BILL--
==> Processing: 1234-5555-UTIL-BILL--
==> Processing: 3456-7777-UTIL-BILL--
==> Processing: 3999-9999-UTIL-BILL--
==> Processing: 4444-1234-UTIL-BILL--
==> Processing: 4999-R777-UTIL-BILL--
==> Processing: 5111-8888-UTIL-BILL--
==> Processing: 5555-7010-UTIL-BILL--
==> Processing: 6750-R758-UTIL-BILL--
==> Processing: 7000-7000-UTIL-BILL--
==> Processing: 2428-1927-UTIL-BILL--
==> Processing: 8234-5555-UTIL-BILL--

==> Transactions Read       :     12
==> Transactions Archived   :     12
==> Transactions In Error   :      0
==> Transactions Rolled Back:      0

==> Warning count:    0
==> Error   count:    0

Elapsed Time: 34 seconds
--- GenArc Completed ---
```

Running GenArcX

This task is optional. The JCLLIB member GenArcX executes the GenArc program in two steps. The first step, called GenArcD, executes the IEFBR14 program and contains DD statements that delete various files which will be created in the following GenArc step. The second step, called GenArc, executes the GenArc program to archive to Documanage or to a CARDATA file if you are using stacked DPA.

To test, submit the GenArcX job. The GenArcX job should receive a return code of *00* or *04*. The output for the GenArcX job should include logging messages (sent to SYSPRINT) similar to the following messages if you are using stacked DPA.

RPEX1 messages

```
--- GenArc ---
==> Processing: TransactionId-GroupName1-GroupName2-GroupName3-
TransactionType
==> Processing: 1234567-SAMPCO-LB1--T1
==> Processing: 2234567-SAMPCO-LB1--T1
==> Processing: 5SAMPCO-SAMPCO-LB2--T1
==> Processing: 6SAMPCO-SAMPCO-LB2--T1
==> Processing: 7SAMPCO-SAMPCO-LB2--T1
==> Processing: 8SAMPCO-SAMPCO-LB2--T1
==> Processing: 9SAMPCO-SAMPCO-LB2--T1
==> Processing: 4234567-FSI-CPP--T1
==> Processing: 5234567-FSI-GL--T1
==> Transactions Read       :      9
==> Transactions Archived   :      7
==> Transactions In Error   :      0
==> Transactions Rolled Back:      0
==> Warning count:    1
==> Error   count:    0
Elapsed Time: 27 seconds
--- GenArc Completed ---
```

UTEX1 messages

```
--- GenArc ---
==> Processing: TransactionId-GroupName1-GroupName2-GroupName3-
TransactionType
==> Processing: -UTIL-BILL--
==> Processing: -UTIL-BILL--
==> Processing: -UTIL-BILL--
==> Processing: -UTIL-BILL--
==> Processing: -UTIL-BILL--
==> Processing: -UTIL-BILL--
==> Processing: -UTIL-BILL--
==> Processing: -UTIL-BILL--
==> Processing: -UTIL-BILL--
==> Processing: -UTIL-BILL--
==> Processing: -UTIL-BILL--
==> Processing: -UTIL-BILL--
==> Transactions Read       :     12
==> Transactions Archived   :     12
==> Transactions In Error   :      0
==> Transactions Rolled Back:      0
==> Warning count:    1
==> Error   count:    0
Elapsed Time: 66 seconds
--- GenArc Completed ---
```

Printing the Output

When you performed the tasks discussed in the topic, Running DAPRUN (or GenTrnX, GenDataX, and GenPrtX) and DMRUNLM on page 71, DAPRUN should have produced three print-ready files. These files would be referenced as shown in the following table:

If you configured your system for...	You will see these default DD names...
Xerox Metacode	PRTBAT1, PRTBAT2, and PRTBAT3
IBM AFP	PRTBAT1, PRTBAT2, and PRTBAT3
PostScript	PRTBAT1, PRTBAT2, and PRTBAT3
PDF	PDFLIB(BAT10000), PDFLIB(BAT20000), and PDFLIB(BAT30000)

Consult the appropriate topic below to send the output to the type of printer you are configured for.

Xerox Metacode printer

If you configured the system to create Xerox Metacode, use the GENERXER member of JCLLIB to *IEBGENER* the print files to JES. Make any changes necessary in this member to match your z/OS environment.

BARR-attached Metacode printer

If you have a Metacode printer attached to your local area network (LAN) and you are using the BARR interface, you must first *BARRWRAP* the output print stream, then send this print stream to the printer's spool area.

To BARRWRAP the output print stream, run the BARRWRPX utility. You'll find an example of this utility in JCLLIB. Then download the resulting file to your LAN. Be sure to download without carriage returns or line feeds (CRLF) and without ASCII translation. Once the file is downloaded, copy it to the BARR SPOOL area.

IBM AFP printer

If you have configured the system to create an IBM AFP data stream, use the GENERAFP member of JCLLIB to *IEBGENER* the print files to JES. Make any changes necessary in this member to match your z/OS environment.

Optimizing Performance

This topic will help you configure your system for optimum performance. To gather the following recommendations, we first created benchmarks on a test system. Then, by changing different parameters of that system, we measured performance gains or losses. Here are some of the terms we used during this exercise:

CPU Time. The amount of time that a program, such as GenTrn, GenData, or GenPrint, uses the CPU. In a multi-tasking operating system like z/OS, there are many tasks competing for use of the CPU.

EXCPs. Execute Channel Programs. We have used these counts as basic measurements of I/O activity.

Wall Clock Time. The elapsed time, as measured from the time a program begins to the time that the program ends. This *wall clock time* can vary significantly from one run to another.

Batch Window. Most installations have specific times of the day or night when large batch processes, like this system, are scheduled to run. The time frame in which these processes run is sometimes referred to as the *batch window*. A batch window is measured in *wall clock time*, such as from 10:00 pm to 5:00 am. Your system installation should run fast enough to complete its processing within the *batch window*.

Most, but not all, of the following recommendations are the result of many tests and subsequent improvements designed for a hypothetical user. The characteristics of Documaker Server implemented for this hypothetical user are as follows:

- Extract file with large record length (approximately 25,000 bytes/record).
- Form sets composed with large number of individual images.
- Large number of different recipients (approximately 300).
- Moderate number of transactions (approximately 4,000)

Compile Options

If your license included source code, use the following information when compiling your system for best results.

OPTIMIZE

Use the highest level of C optimization by passing the following option to the compiler:

Use	If you are using
OPT(1)	C/C++ For MVS/ESA, 1 is the highest level
OPT(2)	either of the other C/370 compilers

NOVALIDATEHANDLE

Specify NOVALIDATEHANDLE by passing the following option to the compiler:

DEF(NOVALIDATEHANDLE)

INLINE

The INLINE option can produce performance gains but there are other activities required for it to work.

Language Environment (LE) or C/370 Runtime Options

Use the following runtime options:

REPORT

You can use the REPORT option to gather information on various values reached when you run a particular program, such as GenData or GenPrint. There is overhead involved in running with this option.

Do not use the REPORT option in production runs. Use the REPORT option during testing to report runtime values, then use these values to assess the need to run with additional runtime options such as HEAP.

HEAP

Based on the values received from running with the REPORT option, you may choose to specify the HEAP run-time option, modifying the initial and incremental heap allocations as the values from REPORT would recommend. You would then want to remove the REPORT option.

If time permits, set HEAP values based on the values received from running with the REPORT option.

File Types and Characteristics

Block sizes

Best results have been achieved by specifying large block sizes. In general, try to use half-track block size—that is, if you are creating files on a 3380-type device (which has a track size of 47,476 bytes), specify a block size around 23,500. A half-track block size for a 3390-type device (which has a track size of 56,664 bytes) would be around 28,000.

Specify half-track block sizes for all files.

Record formats

FB stands for the *fixed block* record format and *FBS* stands for the Fixed Block Standard record format. The record format is specified on the JCL DD statement when a particular file is defined. The C function, *fseek*, appears to be able to seek more quickly to a location in a file if that file is defined with a record format of FBS instead of FB.

The *standard* keyword guarantees that all the blocks of a file will be of the stated block size length and that there will be no *short* blocks.

If you are choosing between record formats of FB and FBS for any of these files, choose FBS: EXTRACT, TRNFILE.DAT, NEWTRN.DAT, NAFILE.DAT, and POLFILE.DAT.

As for the size of record blocks, the larger the record length, the fewer records per block. For FBS, this results in additional EXCPs and slower performance.

Number of files

As mentioned earlier, the total run-time, or *wall clock time* of a job is increased as more files are created, opened, and closed. An additional 10 or 20 files may have no significant affect on the wall time but if your system has to create, open and close hundreds of files, the wall time will lengthen significantly.

Design your system with a minimal number of recipient batch and print files. For instance, try to keep the number of recipient files to less than 20 or so. See also the AliasPrintBatches option on page 90.

Extract Files

The standard extract file is usually defined with a record format of fixed block standard (FBS) with a half-track block size.

Defining the extract file as a VSAM ESDS

You may choose to define the extract file to be a VSAM Entry Sequenced Data Set (ESDS). See Defining the Extract File as a VSAM ESDS on page 91 for more information.

DEFLIB

The standard DEFLIB is created as a Partitioned Data Set (PDS), or as a Partitioned Data Set Extended (PDS/E). The best performance results have been achieved with DEFLIB defined with a Variable Blocked (VB) record format.

Placing DDT files into a VSAM KSDS

DEFLIB contains DDT files, DFD files, SETRCPTB, FORMDAT, various tables, and so on. You can move the DDT files into a VSAM KSDS by running the program PDS2VSAM. See Moving DDT Files into a VSAM KSDS on page 93 for more information.

SETRCPTB

Moving the SETRCPTB member into a VSAM KSDS

You can move the set recipient table member, SETRCPTB, into a VSAM KSDS by running the RCP2VSAM utility. See Moving SETRCPTB to a VSAM KSDS on page 99 for more information.

FAPLIB

Define the FAPLIB file as variable blocked (VB) with a half-track block size.

TRNFILE

Define the transaction trigger file (TRNFILE) file as fixed block standard (FBS) with a half-track block size.

NEWTRN

Define the updated transaction trigger file (NEWTRN) file as fixed block standard (FBS) with a half-track block size.

NAFILE

Define the NAFILE file as fixed block standard (FBS) with a half-track block size or use VSAM to create and read from the NAFILE as outlined below.

Creating a NAFILE as a VSAM file

The NAFILE can be created in the GenData step as a VSAM key sequenced data set (KSDS), then read in the GenPrint step. Actually, NAFILE and POLFILE function as a pair so that they must both be either VSAM or non-VSAM. See Creating NAFILEs and POLFILEs as VSAM KSDSs on page 101 for the steps required to create the NAFILE/POLFILE files as VSAM files.

POLFILE

Define the POLFILE file as fixed block standard (FBS) with a half-track block size or use VSAM to create and read from the POLFILE as outlined in the section below.

Creating a POLFILE as a VSAM file

The POLFILE can be created in the GenData step as a VSAM key sequenced data set (KSDS), then read in the GenPrint step. See Creating NAFILEs and POLFILEs as VSAM KSDSs on page 101 for the steps required to create the NAFILE.DAT and POLFILE.DAT files as VSAM files.

Recipient Batch Files

As mentioned earlier, the fewer recipient batch files you have the better the performance. Define the recipient batch files as *fixed block standard* (FBS) with a half-track block size. A rule of thumb is to try to keep the number of recipient batch files under 20. If it is necessary to design the system as having many recipient batch files you may consolidate these logical groupings into a single *physical* recipient batch file. See AliasPrintBatches option on page 90 for more information.

Pre-compiled Xerox Metacode Files (PMETLIB)

The standard PMETLIB is created as a PDS or PDS/E. If you are using a PDS or PDS/E, define PMETLIB with a record format of VB and with a half-track block size.

Placing PMETs into a VSAM KSDS

You can move the PMETs into a VSAM KSDS by running the program PDS2VSAM. See Moving PMETs into a VSAM KSDS on page 97 for the steps to accomplish this task.

Print Files

Define the print files as Variable Blocked Machine (VBM) control codes, with a record length near 255 for Xerox or 8205 for AFP, and a half-track block size.

Setting Your FSISYS INI Options

Caching Options

The following options attempt to minimize the repeated opening and closing of frequently used files (actually, PDS members in MVS) by retaining, or caching, file handles and file data. In many cases the defaults are sufficient but for specific cases, where many different Images are used, these caching values may be increased to improve performance.

Caching FAP files

In some cases, FAP files (image definitions) are loaded from the FORMLIB or FAPLIB PDS during GenData processing. The cache feature keeps frequently used FAP members available for re-use. Here is an example of how you set up the cache feature:

```
< Control >
    CacheFAPFiles = 100
```

The default is 100. Accept the default value *unless* you are loading FAP files in GenData using the CompileInStream FSISYS INI option (set to Yes) and you are using more than 100 FAP files and/or logo files.

DDT files

Data definition table (DDT) members are loaded from the DDTLIB or DEFLIB PDS during GenData processing. The cache feature keeps frequently used DDT members available for re-use. Here is an example of how you set up the cache feature:

```
< Control >
    RuleFilePool = 100
```

The default is 100. Accept the default value unless you are using more than 100 DDT files.

Using/Caching Xerox PMET files

MET files contain pre-compiled Xerox Metacode information produced by the FAP2MET utility. The GenPrint program loads MET members from the PMETLIB PDS. The cache option keeps frequently used MET members available for re-use. Here is an example of how you set up the cache feature:

```
< PrtType:XER >
    CacheFiles     = 100
    CompileInstream= No
```

The default is 100. Accept the default *unless* you are using pre-compiled FAP files you are using more than 100 FAP files and/or logo files.

Using AFP Overlays

For best performance, you should run the FAP2OVL utility, compiling FAPs into AFP overlays. Tell the system to use the overlays by specifying:

```
< PrtType:AFP >
    SendOverlays= Yes
```

Logging Options

LogTransactions option

The GenTrn, GenData, and GenPrint programs optionally place transaction information into a LOG file. In most situations, this information is not needed. The LogTransactions option is specified in:

```
< Control >
    LogTransactions = No
```

The default is Yes.

For optimum performance, specify **No**.

LogToConsole option

The GenTrn, GenData, and GenPrint programs optionally place transaction information in the SYSPRINT DD statement you specified in the job's JCL. In most situations, this information is not needed. The LogToConsole option is specified in:

```
< Control >
    LogToConsole = No
```

The default is Yes.

For optimum performance, specify **No**.

Debug Options

If_Rule control group

```
< If_Rule >
    Debug_if = No
```

The default is Yes.

For optimum performance, specify **No**.

RunMode control group

For these run options:

```
< RunMode >
    DownloadFAP  = No
    LoadCordFAP  = No
    CompiledFAP  = Yes
```

The defaults are...

```
DownloadFAP = No
LoadCordFAP = No
CompiledFAP = Yes
```

Set the DownLoadFAP option to **No**, the LoadCordFAP option to **No**, and the CompileFAP option to **Yes** for the best performance.

Other Options

MaxRecsPerTransaction option

```
< ExtractFile >
    MaxRecsPerTransaction  =  nn
```

The default is zero (0) and there is no maximum. Be careful using this option. You might want to use this option if you *know* that each record in the extract file corresponds to a transaction.

AliasPrintBatches option

```
< ExtractFile >
    AliasPrintBatches  =  Yes
```

The default is No. Use the default.

Defining the Extract File as a VSAM ESDS

The extract file is generally defined as a Fixed Blocked Standard (FBS) file, with the record length depending on the particular implementation. There are circumstances when you might want to define the extract file as a VSAM Entry Sequenced Data Set (ESDS) instead of an FBS file.

Follow the steps below to move the extract file into a VSAM ESDS.

1. Execute the job in member EXTESDS in the JCLLIB dataset. Modify the member as appropriate for your site. The EXTESDS job first uses the IBM program IDCAMS to delete and re-define the VSAM extract file and then uses IDCAMS to copy the (non-VSAM) extract file to the VSAM extract file. Set the RecordSize parameter to the exact record length of the extract file. This job should receive a condition code of oo in each step.

2. Since the name of the VSAM extract file will probably differ from the name of the non-VSAM extract file, change the DAP procedure in PROCLIB (or the GenTrnX and GenDataX JCLLIB members if you are using those jobs) to reference the new name of the extract file. There should be a reference to the extract file in both the GenTrn and GenData steps of the DAP procedure.

3. Modify the FSISYS INI file as follows:

```
< TRN_FILE >
MaxExtReclen=  (the exact record size of the VSAM extract file. The
               maximum LRECL (Logical Record Length) for a dataset
               is 32760—32K minus 8 bytes)
BinaryExt=YES
```

The EXTESDS member of JCLLIB is shown below:

```
//*  COPY JOBCARD HERE
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*        EXTESDS -
//*
//*        THIS JOB PERFORMS 2 STEPS  :
//*
//*        1.  DELETE / DEFINE NEW EXTRACT FILE AS VSAM ESDS.
//*        2.  REPRO EXISTING SEQUENTIAL EXTRACT FILE INTO NEW
//*             ESDS EXTRACT FILE.
//*        3.  &HLQ..&RES.. CANNOT BE USED INSTREAM, PLEASE
//*               REPLACE THE OCCURRENCES IN THE SYSIN DD
//*               STATEMENT WITH THE APPROPRIATE VALUES.
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//DEFINE  EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *

  DEL    &HLQ..&RES..EXTRACT.ESDS

  DEFINE CLUSTER(NAME(&HLQ..&RES..EXTRACT.ESDS)    +
            CYL(25 5)                              +
            NONINDEXED                             +
            REUSE )                                +
```

```
          DATA(NAME(&HLQ..&RES..EXTRACT.ESDS.DATA) +
             RECORDSIZE(??? ???) )

  IF LASTCC = 0 THEN SET MAXCC = 0

//*
//*
//REPRO    EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD  SYSOUT=*
//DD1      DD  DSN=&HLQ..&RES..EXTRACT,DISP=SHR
//DD2      DD  DSN=&HLQ..&RES..EXTRACT.ESDS,DISP=SHR
//SYSIN    DD  *
  REPRO INFILE(DD1) OUTFILE(DD2)
```

Moving DDT Files into a VSAM KSDS

DDT files normally reside as members in the DDTLIB or DEFLIB PDS. The DEFLIB PDS also contains other types of members, such as SETRCPTB, FORMDAT, FXR, FSISYS and various tables. These other types of members should not be copied to the VSAM KSDS, but it probably won't hurt anything if they are copied—so step 1 is optional.

Follow the steps below to move the DDTs in DEFLIB to a VSAM KSDS.

1. You can use the ALLOCDDT job in JCLLIB to define a "DDTLIB PDS" (to contain only DDT files) and to copy the DDT files for the sample resource from DEFLIB into this new PDS. Once you have all the DDT files in a separate (non-VSAM) PDS, proceed to step 2.

   If you are moving your own resources (not the sample resources), copy your DDT files into this DDTLIB.

2. Execute the PDS2VSDX job that resides in the JCLLIB PDS. The job consists of these steps:

   - The IDCAMS step deletes/re-defines the VSAM KSDS that the DDT files will be copied into.
   - The PDS2VSAM step reads the DDTLIB PDS created in step 1 and copies the DDT files to the VSAM KSDS defined in the IDCAMS step.

   The PDS2VSDX job is shown below:

```
//*  COPY JOBCARD HERE
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*  JOB PERFORMS 2 STEPS :
//*
//*        1.  DELETES / RE-DEFINES DDTVSAM KSDS.
//*        2.  RUNS PDS2VSAM PROGRAM TO COPY DDT MEMBERS INTO
//*              DDTVSAM
//*
//*        PRIOR TO RUNNING THIS JOB YOU SHOULD HAVE RUN
//*        THE JOB IN THE "ALLOCDDT" MEMBER OF JCLLIB TO
//*         ALLOCATE A NON-VSAM DDTLIB AND COPY THE DDT'S
//*         FROM DEFLIB INTO IT.
//*
//*  NOTE: THE &HLQ AND &RES WITHIN INSTREAM JCL WILL NOT
//*               WORK.  PLEASE REPLACE THE &HLQ AND &RES WITH
//*               THEIR RESPECTIVE VALUES.
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*
//IDCAMS  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD  *

   DELETE &HLQ..&RES..DDTVSAM CLUSTER

   DEFINE CLUSTER                              -
         ( NAME(&HLQ..&RES..DDTVSAM)       -
           CYL(25 5)                           -
           REUSE                               -
```

```
            SHAREOPTIONS(2 3) )                -
      DATA                                     -
         ( NAME(&HLQ..&RES..DDTVSAM.DATA)  -
           RECORDSIZE(2048 2048)               -
           CONTROLINTERVALSIZE(4096)           -
           FREESPACE(0 10)                     -
           KEYS(16 0) )                        -
      INDEX                                    -
         ( NAME(&HLQ..&RES..DDTVSAM.INDEX))

   IF LASTCC = 00 THEN SET MAXCC = 00

//*
//PDS2VSAM EXEC PGM=PDS2VSAM
//STEPLIB  DD  DISP=SHR,DSN=&HLQ..LINKLIB
//         DD  DISP=SHR,DSN=SYS1.SCEERUN
//SYSOUT   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//PDSIN    DD DSN=&HLQ..&RES..DDTLIB,DISP=SHR
//VSAMOUT  DD DSN=&HLQ..&RES..DDTVSAM,DISP=SHR
```

3 Add this option to the FSISYS INI file:

```
< VSAM >
    DDTVSAM  =  DD:DDTVSAM
```

The DD name you use to associate with the VSAM DDT file is up to you. This INI option tells the system to open and read the DDT file using VSAM.

4 Modify the JCL for the GenData step so the DDTVSAM DD statement is in that step and references the VSAM KSDS created for the DDT files in step 2. If you are using the DAP procedure in PROCLIB, add the appropriate DD statement to that procedure.

If, you are executing each step as a separate job, add the DD statement to the GenDataX job.

MOVING JDTS INTO A VSAM KSDS

The Job Definition Table (JDT) normally resides as member AFGJOB in the DEFLIB PDS. You can move this member (and other JDT members if you have several) into its own VSAM KSDS, similar to the VSAM KSDS used for DDT files.

Follow these steps to move the JDT files in DEFLIB to a VSAM KSDS.

1 You can use the ALLOCJDT job in JCLLIB to define a JDTLIB PDS (to contain only JDT files) and to copy the JDT files for the sample resource from DEFLIB into this new PDS. Once you have all the JDTs in a separate (non-VSAM) PDS proceed to step 2.

If you are moving your own resources (not the sample resources), copy your JDT files into this JDTLIB.

2 Execute the PDS2VSJX job that resides in the JCLLIB PDS. The job consists of these steps:

- The IDCAMS step deletes/re-defines the VSAM KSDS that the JDT files will be copied into.
- The PDS2VSAM step reads the JDTLIB PDS created in step 1 and copies the JDTs to the VSAM KSDS defined in the IDCAMS step.

The PDS2VSJX job is shown below:

```
//*  COPY JOBCARD HERE
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*  JOB PERFORMS 2 STEPS :
//*
//*        1.  DELETES / RE-DEFINES JDTVSAM KSDS.
//*        2.  RUNS PDS2VSAM PROGRAM TO COPY JDT MEMBERS INTO
//*             JDTVSAM
//*
//*        PRIOR TO RUNNING THIS JOB YOU SHOULD HAVE RUN THE
//*        JOB IN MEMBER "ALLOCJDT" OF JCLLIB TO ALLOCATE A
//*        NON-VSAM JDTLIB AND COPY THE JDT MEMBER(S) INTO IT.
//*
//*  NOTE: THE &HLQ AND &RES WITHIN INSTREAM JCL WILL NOT
//*               WORK.  PLEASE REPLACE THE &HLQ AND &RES WITH
//*               THEIR RESPECTIVE VALUES.
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*
//IDCAMS  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD  *

   DELETE &HLQ..&RES..JDTVSAM CLUSTER

   DEFINE CLUSTER                              -
         ( NAME(&HLQ..&RES..JDTVSAM)       -
           CYL(25 5)                           -
           REUSE                               -
           SHAREOPTIONS(2 3) )                 -
      DATA                                     -
         ( NAME(&HLQ..&RES..JDTVSAM.DATA) -
```

```
            RECORDSIZE(2048 2048)              -
            CONTROLINTERVALSIZE(4096)          -
            FREESPACE(0 10)                    -
            KEYS(16 0) )                       -
         INDEX                                 -
            ( NAME(&HLQ..&RES..JDTVSAM.INDEX))

   IF LASTCC = 00 THEN SET MAXCC = 00

//*
//PDS2VSAM EXEC PGM=PDS2VSAM
//STEPLIB  DD  DISP=SHR,DSN=&HLQ..LINKLIB
//         DD  DISP=SHR,DSN=SYS1.SCEERUN
//SYSOUT   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//PDSIN    DD DSN=&HLQ..&RES..JDTLIB,DISP=SHR
//VSAMOUT  DD DSN=&HLQ..&RES..JDTVSAM,DISP=SHR
```

3 Add this option to the FSISYS INI file:

```
< VSAM >
    JDTVSAM  =  DD:JDTVSAM
```

The DD name you associate with the VSAM JDT file is up to you. This INI option tells the system to open and read the JDT file using VSAM.

4 Modify the JCL for the GenData step so that the JDTVSAM DD statement is in that step and references the VSAM KSDS that was created for the JDT files in step 2. If you are using the DAP procedure in PROCLIB add the appropriate DD statement to that procedure. If, on the other hand, you are executing each step as a separate job, add the DD statement to the GenDataX job.

MOVING PMETS INTO A VSAM KSDS

Pre-compiled Metacode files normally reside as members in the PMETLIB PDS. Follow the steps below to move the PMETs in PMETLIB to a VSAM KSDS.

1 Execute the following PDS2VSPX job, changing any necessary values to fit your environment. This job resides in the JCLLIB PDS and consists of these steps:

- The IDCAMS step deletes/re-defines the VSAM KSDS the PMETs will be copied into.
- The PDS2VSAM step reads the PMETLIB PDS and copies the PMET members to the VSAM KSDS defined in the previous step.

```
//*  COPY JOBCARD HERE
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*  JOB PERFORMS 2 STEPS :
//*
//*        1.  DELETES / RE-DEFINES PMETVSAM KSDS.
//*        2.  RUNS PDS2VSAM PROGRAM TO COPY PMET MEMBERS INTO
//*             PMETVSAM
//*        3.  THE DEFAULT INPUT / OUTPUT DD NAMES ARE "PDSIN" AND
//*            "VSAMOUT", RESPECTIVELY.  IF YOU WISH TO OVERRIDE
//*            THESE NAMES PASS THE EXECUTE PARAMETER :
//*            PARM='/ /I=DD:INPUTDD /O=DD:OUTPUTDD'
//*
//*  NOTE: THE &HLQ AND &RES WITHIN INSTREAM JCL WILL NOT
//*                WORK.  PLEASE REPLACE THE &HLQ AND &RES WITH
//*                THEIR RESPECTIVE VALUES.
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*
//IDCAMS  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD  *

   DELETE &HLQ..&RES..PMETVSAM CLUSTER

   DEFINE CLUSTER                              -
         ( NAME(&HLQ..&RES..PMETVSAM)       -
           CYL(25 25)                          -
           REUSE                               -
           SHAREOPTIONS(2 3) )                 -
      DATA                                     -
         ( NAME(&HLQ..&RES..PMETVSAM.DATA) -
           RECORDSIZE(2048 2048)               -
           CONTROLINTERVALSIZE(4096)           -
           FREESPACE(0 10)                     -
           KEYS(16 0) )                        -
      INDEX                                    -
         ( NAME(&HLQ..&RES..PMETVSAM.INDEX) )

   IF LASTCC = 00 THEN SET MAXCC = 00

//*
//*
```

```
//PDS2VSAM EXEC PGM=PDS2VSAM
//STEPLIB  DD DISP=SHR,DSN=&HLQ..LINKLIB
//         DD DISP=SHR,DSN=SYS1.SCEERUN
//SYSOUT   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//PDSIN    DD DSN=&HLQ..&RES..PMETLIB,DISP=SHR
//VSAMOUT  DD DSN=&HLQ..&RES..PMETVSAM,DISP=SHR
```

2 Add this option to the FSISYS INI file:

```
< VSAM >
    METVSAM  =  DD:PMETVSAM
```

The DD name you associate with the VSAM PMET file is up to you. This INI option tells the system to open and read the PMET file using VSAM.

3 Modify the JCL for the GenPrint step so the PMETVSAM DD statement is in that step and it references the VSAM KSDS created for the PMETs in step 1. If you are using the DAP procedure in PROCLIB, add the appropriate DD statement to that procedure. If you are executing each step as a separate job, add the DD statement to the GenPrtX job.

MOVING SETRCPTB TO A VSAM KSDS

Follow the steps below to move the SETRCPTB member from DEFLIB to a VSAM KSDS.

NOTE: Before you run this, please make sure that there are no blank lines at the end of your extract file. Blank lines can cause the following error:

```
DM1002: Error: Required INI definition omitted.
Cannot locate INI group <Key1Table> with value = defined.
```

The error appears when you run Documaker Server.

1 Execute the following RCP2VSMX job, changing any necessary values to fit your installation. This job resides in the JCLLIB PDS and consists of these steps:

- The IDCAMS step deletes/re-defines the VSAM KSDS that the SETRCPTB will be copied into.
- The RCP2VSAM step reads the DEFLIB PDS and copies the SETRCPTB member to the VSAM KSDS defined in the previous step.

```
//*  COPY JOBCARD HERE
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*  JOB PERFORMS 2 STEPS :
//*
//*        1.  DELETES / RE-DEFINES SETRCPVS KSDS.
//*        2.  RUNS RCP2VSAM PROGRAM TO COPY SETRCPTB TO
//*             SETRCPVS KSDS.
//*
//*            DEFAULT INPUT / OUTPUT DD NAMES ARE SETRCPTB AND
//*            SETRCPVS, RESPECTIVELY.  YOU CAN OVERRIDE BY
//*            PASSING THE PARM :  PARM='/ /I=DD:DDIN /O=DD:DDOUT'
//*
//*  NOTE: THE &HLQ AND &RES WITHIN INSTREAM JCL WILL NOT
//*              WORK.  PLEASE REPLACE THE &HLQ AND &RES WITH
//*              THEIR RESPECTIVE VALUES.
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*
//DEFINE  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD  *

   DELETE &HLQ..&RES..SETRCPVS CLUSTER

   DEFINE CLUSTER                              -
         ( NAME(&HLQ..&RES..SETRCPVS)      -
           RECORDSIZE(500 1000)              -
           KEYS(152 0)                       -
           REUSE                             -
           SHAREOPTIONS(2 3) )               -
      DATA                                   -
         ( NAME(&HLQ..&RES..SETRCPVS.DATA) -
           CONTROLINTERVALSIZE(4096)         -
```

```
            RECORDS(1000 1000)                -
            FREESPACE(0 5) )                  -
      INDEX                                   -
         ( NAME(&HLQ..&RES..SETRCPVS.INDEX) -
           RECORDS(100 50) )

   IF LASTCC = 00 THEN SET MAXCC = 00
//*
//RCP2VSAM EXEC PGM=RCP2VSAM
//STEPLIB  DD  DISP=SHR,DSN=&HLQ..LINKLIB
//         DD  DISP=SHR,DSN=SYS1.SCEERUN
//SETRCPTB DD  DISP=SHR,DSN=&HLQ..&RES..DEFLIB(SETRCPTB)
//SETRCPVS DD  DISP=SHR,DSN=&HLQ..&RES..SETRCPVS
//SYSPRINT DD  SYSOUT=*
```

2 Add this option to the FSISYS INI file:

```
< VSAM >
     VSAMRCPTB   =   DD:SETRCPVS
```

The DD name you associate with the VSAM SETRCPTB file is up to you. This INI option tells the system to open and read the SETRCPTB file using VSAM.

3 Modify the JCL for the GenData step so the SETRCPVS DD statement is in that step and it references the VSAM KSDS created for the SETRCPTB in step 1. If you are using the DAP procedure in PROCLIB, add the appropriate DD statement to that procedure. If you are executing each step as a separate job, add the DD statement to the GenDataX job.

Creating NAFILEs and POLFILEs as VSAM KSDSs

Normally, the NAFILE and POLFILE, which are created and written to during the GenData step, are defined as sequential files with a record format of FBS. You can, however, create them as VSAM KSDSs. The member, NAPOLVS, in JCLLIB provides the JCL for the additional IDCAMS job step you will need. This JCL is also listed below:

```
//NAPOLVS  EXEC PGM=IDCAMS
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  *

  DELETE &HLQ..&RES..GENDATA.NAFILE.KSDS PURGE
  DELETE &HLQ..&RES..GENDATA.POLFILE.KSDS PURGE

  DEFINE CLUSTER(NAME(&HLQ..&RES..GENDATA.NAFILE.KSDS) +
           CYL(5 1)      +
           KEY(4 0)      +
           REUSE )       +
         DATA(NAME(&HLQ..&RES..GENDATA.NAFILE.KSDS.DATA) +
           RECORDSIZE(2048 2048) )                      +
         INDEX(NAME(&HLQ..&RES..GENDATA.NAFILE.KSDS.INDEX))

  DEFINE CLUSTER(NAME(&HLQ..&RES..GENDATA.POLFILE.KSDS) +
           CYL(5 1)      +
           KEY(4 0)      +
           REUSE )       +
         DATA(NAME(&HLQ..&RES..GENDATA.POLFILE.KSDS.DATA) +
           RECORDSIZE(2048 2048) )                       +
         INDEX(NAME(&HLQ..&RES..GENDATA.POLFILE.KSDS.INDEX))
```

To use VSAM for the NAFILE and POLFILE follow these steps:

1. Insert the NAPOLVS JCL step listed above into the GenData step prior to the GenData execution step, changing any file names or other values to conform to your environment.
2. Make sure the NAFILE and POLFILE file names in the GenData and GenPrint steps (GenDataX and GenPrtX if you are executing each step as a separate job) reflect the name of the VSAM files created in the IDCAMS step.
3. Add this option to the FSISYS INI file:

   ```
   < VSAM >
       VSAMNA  =  DD:NAFILE
   ```

   The DD name you associate with the VSAM NAFILE file is up to you. This INI option tells the system to open and read both the NAFILE and the POLFILE using VSAM.

The LOADDAP Job

The LOADDAP job in JCLLIB is listed below. This job creates the...

- system datasets
- RPEX1 and UTEX1 sample datasets (for the verification test)
- font datasets

In addition, the LOADDAP job reads from the install tape and populates these datasets.

```
//*  COPY JOBCARD HERE ...
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*       LOADDAP -
//*
//*       COPY THE DAP BATCH SYSTEM FILES, RPEX1 AND UTEX1 RESOURCES
//*       FROM A 3480 CARTRIDGE TO AN MVS SYSTEM.  FILES LOADED ARE :
//*
//*        1.  JCLLIB
//*        2.  PROCLIB
//*        3.  SOURCE
//*        4.  INCLIB
//*        5.  OBJLIB
//*        6.  LINKLIB
//*        7.  DBRMLIB (DB2)
//*        8.  DEFLIB  (RPEX1)
//*        9.  DDTLIB  (RPEX1)
//*       10.  FAPLIB  (RPEX1)
//*       11.  COMPLIB (RPEX1)
//*       12.  EXTRACT (RPEX1)
//*       13.  LOG2IMG (RPEX1 XEROX LOGO IMAGES)
//*       14.  LOG2FNT (RPEX1 XEROX LOGO FONTS)
//*       15.  PFRMLIB (RPEX1 XEROX FRM'S)
//*       16.  DEFLIB  (UTEX1)
//*       17.  DDTLIB  (UTEX1)
//*       18.  FAPLIB  (UTEX1)
//*       19.  COMPLIB (UTEX1)
//*       20.  EXTRACT (UTEX1)
//*       21.  LOG2IMG (UTEX1 XEROX LOGO IMAGES)
//*       22.  LOG2FNT (UTEX1 XEROX LOGO FONTS)
//*       23.  AFP240  (AFP 240-DPI FONTS)
//*       24.  AFP300  (AFP 300-DPI FONTS)
//*       25.  XEROX   (XEROX FONTS)
//*       26.  XEROXR  (XEROX ROTATED FONTS)
//*
//*
//*      =>  GO TO BOTTOM OF JOB AND CHANGE VOLSER & CUSTHLQ TO MATCH
YOUR
//*       =>  SITE REQUIREMENTS.
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*
//*        IN-STREAM PROC FOLLOWS ...
//*
//LOAD    PROC  TUNIT=,
//           TVOLSER=,
//           CUSTHLQ=,
//           DUNIT=,
```

```
//          PDSTYPE=
//*
//JCLLIB  EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&HLQ..JCLLIB,DISP=SHR,UNIT=&TUNIT,
//         VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(1,SL)
//SYSUT2   DD  DSN=&CUSTHLQ..JCLLIB,DISP=(,CATLG),UNIT=&DUNIT,
//         SPACE=(CYL,(2,1,10)),DSNTYPE=&PDSTYPE
//SYSIN    DD DUMMY
//*
//PROCLIB EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&HLQ..PROCLIB,DISP=SHR,UNIT=&TUNIT,
//         VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(2,SL)
//SYSUT2   DD  DSN=&CUSTHLQ..PROCLIB,DISP=(,CATLG),UNIT=&DUNIT,
//         SPACE=(CYL,(2,1,10)),DSNTYPE=&PDSTYPE
//SYSIN    DD DUMMY
//*
//SOURCE  EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&HLQ..SOURCE,DISP=SHR,UNIT=&TUNIT,
//         VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(3,SL)
//SYSUT2   DD  DSN=&CUSTHLQ..SOURCE,DISP=(,CATLG),UNIT=&DUNIT,
//         SPACE=(CYL,(8,10,100)),DSNTYPE=&PDSTYPE
//SYSIN    DD DUMMY
//*
//INCLIB  EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&HLQ..INCLIB,DISP=SHR,UNIT=&TUNIT,
//         VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(4,SL)
//SYSUT2   DD  DSN=&CUSTHLQ..INCLIB,DISP=(,CATLG),UNIT=&DUNIT,
//         SPACE=(CYL,(4,1,50)),DSNTYPE=&PDSTYPE
//SYSIN    DD DUMMY
//*
//OBJLIB  EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&HLQ..OBJLIB,DISP=SHR,UNIT=&TUNIT,
//         VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(5,SL)
//SYSUT2   DD  DSN=&CUSTHLQ..OBJLIB,DISP=(,CATLG),UNIT=&DUNIT,
//         SPACE=(CYL,(35,5,75)),DSNTYPE=&PDSTYPE
//SYSIN    DD DUMMY
//*
//*  DEFINE LINKLIB AS PDS (NOT PDSE) BY DEFAULT
//*
//LINKLIB EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&HLQ..LINKLIB,DISP=SHR,UNIT=&TUNIT,
//         VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(6,SL)
//SYSUT2   DD  DSN=&CUSTHLQ..LINKLIB,DISP=(,CATLG),UNIT=&DUNIT,
//         SPACE=(CYL,(30,5,10)),DSNTYPE=PDS
//SYSIN    DD DUMMY
//*
//*  DON'T GET SYSCPRT FROM TAPE - JUST ALLOCATE SPACE FOR IT
//*    FOR LATER COMPILES.
//*
```

```
//SYSCPRT EXEC PGM=IEFBR14
//SYSPRINT DD  SYSOUT=*
//DD1      DD  DSN=&CUSTHLQ..SYSCPRT,DISP=(,CATLG),UNIT=&DUNIT,
//          SPACE=(CYL,(25,25,100)),DSNTYPE=&PDSTYPE
//SYSIN    DD DUMMY
//*
//DBRMLIB EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&HLQ..DBRMLIB,DISP=SHR,UNIT=&TUNIT,
//          VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(7,SL)
//SYSUT2   DD  DSN=&CUSTHLQ..DBRMLIB,DISP=(,CATLG),UNIT=&DUNIT,
//          SPACE=(CYL,(1,1,5)),DSNTYPE=&PDSTYPE
//SYSIN    DD DUMMY
//*
//DEFLIB  EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&HLQ..RPEX1.DEFLIB,DISP=SHR,UNIT=&TUNIT,
//          VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(8,SL)
//SYSUT2   DD  DSN=&CUSTHLQ..RPEX1.DEFLIB,DISP=(,CATLG),UNIT=&DUNIT,
//          SPACE=(CYL,(3,1,15)),DSNTYPE=&PDSTYPE
//SYSIN    DD DUMMY
//*
//DDTLIB  EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&HLQ..RPEX1.DDTLIB,DISP=SHR,UNIT=&TUNIT,
//          VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(9,SL)
//SYSUT2   DD  DSN=&CUSTHLQ..RPEX1.DDTLIB,DISP=(,CATLG),UNIT=&DUNIT,
//          SPACE=(CYL,(2,2,15)),DSNTYPE=&PDSTYPE
//SYSIN    DD DUMMY
//*
//FAPLIB  EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&HLQ..RPEX1.FAPLIB,DISP=SHR,UNIT=&TUNIT,
//          VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(10,SL)
//SYSUT2   DD  DSN=&CUSTHLQ..RPEX1.FAPLIB,DISP=(,CATLG),UNIT=&DUNIT,
//          SPACE=(CYL,(2,2,15)),DSNTYPE=&PDSTYPE
//SYSIN    DD DUMMY
//*
//COMPLIB EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&HLQ..RPEX1.COMPLIB,DISP=SHR,UNIT=&TUNIT,
//          VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(11,SL)
//SYSUT2   DD
DSN=&CUSTHLQ..RPEX1.COMPLIB,DISP=(,CATLG),UNIT=&DUNIT,
//          SPACE=(CYL,(2,2,15)),DSNTYPE=&PDSTYPE
//SYSIN    DD DUMMY
//*
//*       NOTE THAT IEBGENER IS USED TO COPY THE EXTRACT FILE, WHICH
//*         IS A SEQUENTIAL FILE, NOT A PDS.
//*
//EXTRACT EXEC PGM=IEBGENER
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&HLQ..RPEX1.EXTRACT,DISP=SHR,UNIT=&TUNIT,
//          VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(12,SL)
//SYSUT2   DD
DSN=&CUSTHLQ..RPEX1.EXTRACT,DISP=(,CATLG),UNIT=&DUNIT,
```

```
//          SPACE=(CYL,(1,1))
//SYSIN     DD  DUMMY
//*
//LOG2IMG EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1    DD  DSN=&HLQ..RPEX1.LOG2IMG,DISP=SHR,UNIT=&TUNIT,
//          VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(13,SL)
//SYSUT2    DD
DSN=&CUSTHLQ..RPEX1.LOG2IMG,DISP=(,CATLG),UNIT=&DUNIT,
//          SPACE=(CYL,(1,1,5)),DSNTYPE=&PDSTYPE
//SYSIN     DD DUMMY
//*
//LOG2XFNT EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1    DD  DSN=&HLQ..RPEX1.LOG2XFNT,DISP=SHR,UNIT=&TUNIT,
//          VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(14,SL)
//SYSUT2    DD
DSN=&CUSTHLQ..RPEX1.LOG2XFNT,DISP=(,CATLG),UNIT=&DUNIT,
//          SPACE=(CYL,(1,1,5)),DSNTYPE=&PDSTYPE
//SYSIN     DD DUMMY
//*
//PFRMLIB  EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1    DD  DSN=&HLQ..RPEX1.PFRMLIB,DISP=SHR,UNIT=&TUNIT,
//          VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(15,SL)
//SYSUT2    DD
DSN=&CUSTHLQ..RPEX1.PFRMLIB,DISP=(,CATLG),UNIT=&DUNIT,
//          SPACE=(CYL,(1,1,5)),DSNTYPE=&PDSTYPE
//SYSIN     DD DUMMY
//*
//DEFLIBU EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1    DD  DSN=&HLQ..UTEX1.DEFLIB,DISP=SHR,UNIT=&TUNIT,
//          VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(16,SL)
//SYSUT2   DD  DSN=&CUSTHLQ..UTEX1.DEFLIB,DISP=(,CATLG),UNIT=&DUNIT,
//          SPACE=(CYL,(3,1,15)),DSNTYPE=&PDSTYPE
//SYSIN     DD DUMMY
//*
//DDTLIBU EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1    DD  DSN=&HLQ..UTEX1.DDTLIB,DISP=SHR,UNIT=&TUNIT,
//          VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(17,SL)
//SYSUT2   DD  DSN=&CUSTHLQ..UTEX1.DDTLIB,DISP=(,CATLG),UNIT=&DUNIT,
//          SPACE=(CYL,(2,2,15)),DSNTYPE=&PDSTYPE
//SYSIN     DD DUMMY
//*
//FAPLIBU EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1    DD  DSN=&HLQ..UTEX1.FAPLIB,DISP=SHR,UNIT=&TUNIT,
//          VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(18,SL)
//SYSUT2   DD  DSN=&CUSTHLQ..UTEX1.FAPLIB,DISP=(,CATLG),UNIT=&DUNIT,
//          SPACE=(CYL,(2,2,15)),DSNTYPE=&PDSTYPE
//SYSIN     DD DUMMY
//*
//COMPLIBU EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
```

```
//SYSUT1   DD  DSN=&HLQ..UTEX1.COMPLIB,DISP=SHR,UNIT=&TUNIT,
//         VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(19,SL)
//SYSUT2   DD
DSN=&CUSTHLQ..UTEX1.COMPLIB,DISP=(,CATLG),UNIT=&DUNIT,
//         SPACE=(CYL,(2,2,15)),DSNTYPE=&PDSTYPE
//SYSIN    DD DUMMY
//*
//*       NOTE THAT IEBGENER IS USED TO COPY THE EXTRACT FILE, WHICH
//*         IS A SEQUENTIAL FILE, NOT A PDS.
//*
//EXTRACTU EXEC PGM=IEBGENER
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&HLQ..UTEX1.EXTRACT,DISP=SHR,UNIT=&TUNIT,
//         VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(20,SL)
//SYSUT2   DD
DSN=&CUSTHLQ..UTEX1.EXTRACT,DISP=(,CATLG),UNIT=&DUNIT,
//         SPACE=(CYL,(1,1))
//SYSIN    DD  DUMMY
//*
//LOG2IMGU EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&HLQ..UTEX1.LOG2IMG,DISP=SHR,UNIT=&TUNIT,
//         VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(21,SL)
//SYSUT2   DD
DSN=&CUSTHLQ..UTEX1.LOG2IMG,DISP=(,CATLG),UNIT=&DUNIT,
//         SPACE=(CYL,(1,1,5)),DSNTYPE=&PDSTYPE
//SYSIN    DD DUMMY
//*
//LOG2XFTU EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=&HLQ..UTEX1.LOG2XFNT,DISP=SHR,UNIT=&TUNIT,
//         VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(22,SL)
//SYSUT2   DD
DSN=&CUSTHLQ..UTEX1.LOG2XFNT,DISP=(,CATLG),UNIT=&DUNIT,
//         SPACE=(CYL,(1,1,5)),DSNTYPE=&PDSTYPE
//SYSIN    DD DUMMY
//*
//*       FORCE AFP FONTLIB TO BE PDS - PDSE HAS PROBLEMS
//*
//AFP240  EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=FSI.AGFA.AFP240.FONTLIB,DISP=SHR,UNIT=&TUNIT,
//         VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(23,SL)
//SYSUT2   DD  DSN=&CUSTHLQ..AGFA.AFP240.FONTLIB,DISP=(,CATLG),
//         UNIT=&DUNIT,
//         SPACE=(CYL,(35,10,150)),DSNTYPE=PDS
//SYSIN    DD DUMMY
//*
//*       FORCE AFP FONTLIB TO BE PDS - PDSE HAS PROBLEMS
//*
//AFP300  EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=FSI.AGFA.AFP300.FONTLIB,DISP=SHR,UNIT=&TUNIT,
//         VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(24,SL)
//SYSUT2   DD  DSN=&CUSTHLQ..AGFA.AFP300.FONTLIB,DISP=(,CATLG),
//         UNIT=&DUNIT,
```

```
//          SPACE=(CYL,(40,10,150)),DSNTYPE=PDS
//SYSIN     DD DUMMY
//*
//XEROX   EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1    DD  DSN=FSI.AGFA.XEROX.FONTLIB,DISP=SHR,UNIT=&TUNIT,
//          VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(25,SL)
//SYSUT2    DD  DSN=&CUSTHLQ..AGFA.XEROX.FONTLIB,DISP=(,CATLG),
//          UNIT=&DUNIT,
//          SPACE=(CYL,(25,5,75)),DSNTYPE=&PDSTYPE
//SYSIN     DD DUMMY
//XEROXR    EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD DSN=FSI.AGFA.XEROX.ROTATED.FONTLIB,DISP=SHR,UNIT=&TUNIT,
//          VOL=(,RETAIN,,,SER=&TVOLSER),LABEL=(26,SL)
//SYSUT2   DD DSN=&CUSTHLQ..AGFA.XEROX.ROTATED.FONTLIB,DISP=(,CATLG),
//          UNIT=&DUNIT,
//          SPACE=(CYL,(115,10,500)),DSNTYPE=&PDSTYPE
//SYSIN     DD DUMMY
//*
//         PEND
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*
//STEP1 EXEC LOAD,TUNIT=CART,
//          TVOLSER=D10301,           <= CHANGE
//          CUSTHLQ='FSI.v103',       <= CHANGE
//          DUNIT=SYSDA,
//          PDSTYPE=LIBRARY
//*
```

z/OS Libraries

The system begins as many source code modules which are separated into directories or libraries. Each library performs a specific task. For instance, INILIB handles Initialization file tasks and GVMLIB performs global variable manager functions.

Some of these libraries perform functions that are relevant only on the PC platform. You do not need to upload these libraries to your z/OS system. The libraries you must upload to your z/OS system are listed in the table below, along with the various print drivers.

An asterisk (*) indicates the library is needed only to perform a particular function. For instance FAP2MET is required only if you are converting FAPs to PMETs and running with the CompileInstream INI option set to *No*.

Library	Required for the system?	Required for Xerox?	Required for AFP?	Required for Postscript?	Required for PCL?
AFP2MVS			Y		
AFPCOPY			Y		
AFPFMDEF			Y		
AFPLIB			Y		
AFPPRT			Y		
ARCLIB	Y*				
ASCLIB	Y				
BARR2MVS					
BARRWRAP		Y*			
CARLIB	Y				
CFA2FAP	Y				
CPCNV	Y				
CUSLIB	Y				
CUSLIB	Y				
DALLIB	Y				
DB2LIB	Y*				
DBLIB	Y				
DSLIB	Y				
ERRLIB	Y				
FAP2CFA					
FAP2MET		Y*			

Library	Required for the system?	Required for Xerox?	Required for AFP?	Required for Postscript?	Required for PCL?
FAP2OVL			Y*		
FAPLIB	Y				
FDT2CFA	Y				
FIXOFFS					
FNTLIB	Y				
FSILIB	Y				
FSIVER	Y				
GenArc	Y*				
GenData	Y				
GENLIB	Y				
GenPrint	Y				
GenTrn	Y				
GLBLIB	Y				
GRFLIB	Y				
GVMLIB	Y				
INILIB	Y				
LBYLIB	Y				
LOG2PSEG			Y*		
LOGLIB	Y				
METOPT		Y			
PCLLIB					Y
PDS2VSAM	Y*				
PRTLIB	Y				
PSTLIB				Y	
RCBLIB	Y				
RCP2VSAM	Y*				
RPLIB	Y				

Library	Required for the system?	Required for Xerox?	Required for AFP?	Required for Postscript?	Required for PCL?
RULLIB	Y				
SAMPAFP			Y		
TIFLIB	Y				
TRANSLAT	Y				
UTILLIB	Y				
UTLLIB	Y				
VMMLIB	Y				
XDBLIB	Y				
XERDNLD		Y*			
XERLIB		Y			
XMCLIB		Y			

Uploading and Downloading Resources

The standard directory structure for resources on the PC is:

```
C:\RPEX1\
    DEFLIB\
    DDTLIB\
    COMPLIB\
    FORMS\
    EXTRACT\
```

These resources exist on the PC as text files and are sometimes referred to as *ASCII files* (COMPLIB is an exception and will be explained shortly). The resources are generally uploaded into files as follows:

PC directory	z/OS file name	z/OS file type
DEFLIB*.*	&HLQ..&RES..DEFLIB(*)	PDS
DDTLIB*.*	&HLQ..&RES..DDTLIB(*)	PDS
FORMS*.*	&HLQ..&RES..FAPLIB(*)	PDS
EXTRACT*.*	&HLQ..&RES..EXTRACT	Sequential file

Text files on a PC are represented using the ASCII character set. Text files on a z/OS system are represented using the EBCDIC character set. As these text files are uploaded from the PC to a z/OS system, each text character must be translated from ASCII to EBCDIC. The program that provides the communications between your PC and the z/OS system (3270 Emulator) generally includes a file transfer feature. Included in the file transfer feature is an option to translate files from ASCII to EBCDIC as the files are uploaded.

COMPLIB stands for *Compiled Resources Library*. FAP files can be parsed, or *compiled*, before you run the system, and placed into the COMPLIB. The utilities you can use to compile FAP files are: FAP2CFA and FDT2CFA.

This compilation improves performance since the text-oriented FAP files do not have to be parsed again during the print assembly process.

Note: You *must*, however, run the utility using the same version and on the same platform on which you will run the system—you *cannot* compile the FAP files on the PC and upload the resulting CFA files onto a z/OS system. You *must* run the FAP2CFA or FDT2CFA utilities on your z/OS system.

Transferring Files

Your 3270 emulation program may include features which let you transfer files from the PC to the z/OS system (upload) and from the z/OS system to the PC (download). Part of this capability will probably be an option to transfer a *binary* file or a *text* file. Generally, when a 3270 emulator uploads a binary file, it does not translate characters as the file is uploaded—it uploads the file as is. What's more, any carriage return and line feed (CRLF) characters are also uploaded as is.

Conversely, when a 3270 emulation program uploads a file designated as text, it translates each character as it uploads it and removes any CRLF characters from the file. Characters *between* the CRLFs are assumed to be the records and are placed into the file on z/OS. You will almost always upload your resources to the z/OS as *text* files (see the following topic, Handling International Characters on page 113,for an exception).

Your 3270 emulation program may also let you upload a file from the PC to z/OS using the DOS Command Line interface. The upload program is usually called SEND and the download program is called RECEIVE.

For example, to upload the PC file "myfap fap" in the c:\rpex1\forms directory to the z/OS PDS "DOC.FAPLIB" you could enter the following command in a DOS window:

SEND c:\rpex1\forms\myfap.fap 'doc.faplib(myfap)' ASCII CRLF

This SEND command include the exact file name and full path name of the PC file followed by a space and then the name of the host file ('doc.faplib(myfap)'). In this case the host file is a PDS (Partioned Data Set) so you must indicate what member to place the file into.

The *ASCII* parameter tells the system to translate the PC file from ASCII to EBCDIC as it's uploaded. The *CRLF* parameter tells the system to remove the carriage return and line feed characters as it's uploaded.

Using the ASCII and CRLF parameters tells the SEND command to treat the file as a *text* file. If you omitted the ASCII and CRLF parameters, the SEND command would treat the file as a *binary* file. There are some situations where you might want to use just the ASCII parameter or just the CRLF parameter, but these situations are rare (see Handling International Characters on page 113, for an example) and you will generally use either both of them or neither of them.

Handling International Characters

Docucorp uses the Windows ANSI code page (1004) for files residing on the PC and the EBCDIC code page 37 for files residing on a z/OS system. There are, however, characters in code page 1004 which are not in code page 37—mainly desktop publishing characters from code point 128 to 159.

To support these characters, the system uses undefined code points in code page 37 (code points below 64). For maximum portability, avoid using characters which are not defined in code page 37.

If you have resource files, such as extract or FAP files, on the PC which contain international characters you will need to translate those characters into the correct code point when you upload them from the PC to a z/OS system.

Depending on your 3270 emulation program, you might be able to tell it to translate the file during the upload. If your 3270 emulation program does not let you specify the code page to translate from and to during the upload, or if your resource files on the PC use some of the undefined code points in code page 37, use the CPCNV utility.

The CPCNV utility converts text files from one code page to another. You can run the CPCNV utility either on the PC or on a z/OS system. Below are examples of how you would convert a FAP file named *french1.fap*, on a PC or on a z/OS system:

Running CPCNV on a PC

Follow these steps to run the CPCNV utility on a personal computer:

1 Run the CPCNV program on the FAP files as follows:

```
cpcnv  /i=french1  /s=1004  /d=37
```

2 Upload the FAP files as follows:

```
SEND french1.fap  'doc.faplib(french1)' CRLF
```

Note that this is one of the cases in which you use the CRLF parameter but do not use the ASCII parameter. You do not use the ASCII parameter because you have already translated the text characters in french1.fap from ANSI code page 1004 (ASCII) to EBCDIC code page 37. You still use CRLF however, because you want the upload process to remove the carriage return and line feed characters from the file as it is uploaded.

Running CPCNV on z/OS

Follow these steps to run the CPCNV utility on z/OS:

1 Upload the FAP file as follows:

```
SEND french1.fap  'doc.faplib.unconv(french1)' CRLF
```

Again, note that you do not use ASCII, but you do use CRLF. In this case, you do not want the 3270 Emulator program to translate from ASCII to EBCDIC because you are going to let the CPCNV utility on a z/OS system perform the translation. Also, note that the file name on the z/OS system of *'doc.faplib.unconv'* denotes that the resources in this PDS *are not* converted.

2 Run the CPCNV utility on the FAP file using, as an example, the JCL provided in member CPCNVX of JCLLIB. Just as on the PC, the source code page will be 1004 and the destination code page will be 37.

Xerox Image, Font, and Form Files

You can use the XERDNLD utility to send a resource to the Xerox printer and save it on the printer's disk drive. Some of the Xerox resources you might want to save to the printer are Xerox images (IMG), fonts (FNT) and forms (FRM). Each of these resources must first be created on the PC and then uploaded to the z/OS system *without* ASCII or CRLF.

To get a Xerox resource from the PC up to the z/OS system and then to a Xerox printer, follow these steps:

1. Define a PDS on the z/OS system to upload into. Define the PDS with the DCB characteristics as shown in the XERLOGOS job in JCLLIB, such as RECFM=VBM, LRECL=512, BLKSIZE=23000.
2. Upload the Xerox resource from the PC to the z/OS system *without* ASCII or CRLF.
3. Run the XERDNLD utility against this resource. There is a sample job in JCLLIB called XERDNLDX which runs XERDNLD. Note the output PDS is again defined with the characteristics RECFM=VBM, LRECL=512, and BLKSIZE=23000.
4. Run the GENERXER job in JCLLIB. Change the SYSUT1 DD statement so it points to the output from the XERDNLD utility, as discussed in step 3.

   Change the SYSUT2 DD statement's SYSOUT class to the correct class for the Xerox printer. The GENERXER job will copy the resource into this SYSOUT class and the resource will be saved onto the Xerox printer's disk drive. Before you send the resource to the printer, start the Xerox printer in the same manner you currently start it (or intend to start) to print Documaker Server output.

Xerox Pre-compiled Metacode (PMET) Files

Currently, you cannot upload Xerox pre-compiled Metacode (PMET) files generated on the PC onto a z/OS system. You must run the FAP2MET utility on a z/OS system and convert the FAPs to PMETs on your z/OS system.

z/OS Programs and Utilities

The following is a list and a short description of the programs and utilities you can run on a z/OS system. For more information about these utilities, see the Docutoolbox Reference.

AFPCOPY utility

This utility takes an AFP print stream which may contain multiple BEGINDOC and ENDDOC statements and generates an AFP print stream with only one BEGINDOC and one ENDDOC statement—in other words, it converts the print stream from *cut-sheet* to *continuous-form*. No parameters are required. The input AFP print stream is read from the INFILE DD statement and the output AFP print stream is written to the OUTFILE DD statement. Look in member AFPCOPYX of JCLLIB to find an example of this utility.

AFPOPT utility

This utility optimizes an AFP print stream. The AFPOPT utility reads an AFP print stream produced by Documaker and outputs a smaller, optimized AFP print stream. The input AFP print stream is read from the INFILE DD statement. The output AFP print stream is written to the OUTFILE DD statement. Look in member AFPOPTX of JCLLIB to find an example of this utility.

AFPFMDEF utility

This utility accepts as input a source form definition file and generates an AFP FormDef. Look in member AFPFMDFX of JCLLIB to find an example of this utility.

AFP2MVS utility

This utility accepts as input an AFP resource, such as a font, page segment, or print-ready file, that's been uploaded to the z/OS system from the PC with no ASCII and no CRLF. It also generates that resource to another file such that the record-oriented format of the resource will be restored. Look in the AFP2MVSX member of JCLLIB to find an example of this utility.

BARR2MVS utility

This utility program accepts as input a print-ready Metacode file produced by Documaker Server running on the PC (using these INI options: Environment=OS22, OutMode=BARR, or BARRWORD), strips that file of the BARR-encoded information and produces a Metacode print-ready file acceptable in a z/OS environment. Look in the BARR2MVX member of JCLLIB to find an example of this utility.

BARRWRAP utility

This utility accepts as input a print-ready Metacode file produced by Documaker Server on z/OS and produces a Metacode print-ready file *wrapped* with control information the BARR printer-attachment system understands. Look in the BARRWRPX member of JCLLIB to find an example of this utility.

CPCNV utility

This utility converts a source file from one code page to another. The utility reads an input source file, such as an extract file, a FAP file, and so on, which corresponds to a specific code page. You can convert either a single member in the PDS (use /I=member) or all members of the PDS (use /I=*). The utility then generates a target file which corresponds to a different code page. Look in member CPCNVX of JCLLIB to find an example of this utility.

FAP2AFP utility

This utility generates an AFP print-ready file from a FAP file. You can convert a single FAP file in the PDS (/I=FAPNAME) or all FAP files in the PDS (/I=*). Look in the FAP2AFPX member of JCLLIB to find an example of this utility.

You can also convert a FAP file in the library (use /LIB=LIBNAME). You can specify the optional parameter /VER and /REV parameters to specify the version and/or revision of a FAP file in the library you want to convert. Keep in mind that you cannot use wildcards (/I=*) when you include the /LIB parameter.

Look in the FAP2AFPL member of JCLLIB to find an example of this utility.

FAP2CFA utility

This utility accepts as input a FAP file and generates a compiled FAP file. Look in member FAP2CFAX of JCLLIB to find an example of this utility.

FAP2MET utility

This utility accepts as input a FAP file and generates a pre-compiled Metacode member, usually placing the *PMET* into the PMETLIB dataset. You can convert a single FAP in the PDS (use /I=fapname) or all FAPs in the PDS (use /I=*). There are several options you can select when you run the FAP2MET utility. For example, you can save the PMET in the PMETLIB dataset so it is ready to be included by the GenPrint program into a print stream or you can generate the PMET so it can be sent directly to the Metacode printer. You can also select the HMI, Short Edge Bind, or Long Edge Bind options. Look in member FAP2METX of JCLLIB to find an example of this utility.

FAP2OVL utility

This utility generates an AFP overlay from a FAP file, usually placing it into an overlay library PDS, which is generally referred to as *OVERLIB*. You can convert a single FAP in the PDS (use /I=fapname) or all FAPs in the PDS (use /I=*). Look in the FAP2OVLX member of JCLLIB to find an example of this utility.

FDT2CFA utility

This utility accepts as input a FSISYS INI file (member of DEFLIB), a FORM file (member of DEFLIB) and the FAPLIB PDS (Partitioned Data Set) and produces a COMPLIB PDS containing compiled FAP files.

This utility is similar to the FAP2CFA utility, except the FAP2CFA utility reads a single FAP and compiles it into a compiled FAP, while the FDT2CFA utility reads the FSISYS INI file, gets the name of the FORM member, opens the FORM member of DEFLIB to get a list of all of the FAP files that might be used, compiles all of these FAP files into compiled FAPs, and places them into the COMPLIB. Look in member FDT2CFAX of JCLLIB to find an example of this utility.

FIXOFFS utility

Use the FIXOFFS utility to correct field offsets after you have uploaded or downloaded a file from an environment where physical offset values differ. For example, you must use this utility if you are running the GenArc program in a z/OS environment using a local area network (LAN) archive of z/OS-generated data.

The utility uses the offsets in the NA and POL files to determine how to update other output files, such as recipient batch files and the NEWTRN file.

FSIVER utility

This utility accepts as input a LINKLIB PDS and generates a report that lists which patches have been applied to the programs in that LINKLIB. You can generate the report for either a single program (use /I=program name) or for all programs in the PDS (use /I=*). Look in member FSIVERX of JCLLIB to find an example of this utility.

SEQ2KSDS utility

Use the SEQ2KSDS utility to convert a non-VSAM NAFILE or POLFILE dataset into a VSAM copy of that dataset. A 4-byte key is prefixed to each record of the VSAM dataset as it is created. You can find sample JCL for running the SEQ2KSDS utility in the SEQ2KSDX member of JCLLIB.

GenArc program

The GenArc program is an optional part of the system which you run after the GenData program. The GenArc program reads the NEWTRN, the NAFILE, and POLFILE and writes (or archives) this information to a repository (DB2). You can retrieve and print these archived form sets. The GenArcDX member of JCLLIB runs GenArc.

GenData program The GenData program uses a processing engine referred to as the *Rules Processor* to step through the transaction trigger file (TRNFILE) created by the GenTrn program. This creates an NAFILE which contains image and variable data information, a POLFILE which contains recipient information, and recipient batch files (BATCH1, BATCH2, and so on) which contain trigger records for the different recipient groups. These files are sent to the GenPrint (or GenArc) program. The GenData program is part of the DAP procedure in PROCLIB. There is also an example of GenData in the GenDataX member of JCLLIB.

GenPrint program The GenPrint program reads the recipient batch files, NAFILEs, and POLFILEs produced by the GenData program and produces print-ready streams for specific printer platforms such as Xerox Metacode or IBM AFP. The print streams can be sent to datasets which you can copy to the JES print SPOOL or directly to JES SPOOL as SYSOUT files. The GenPrint program is part of the DAP procedure in PROCLIB. There is also an example of GenPrint in the GenPrtX member of JCLLIB.

GenTrn program The GenTrn program performs the first step of the processing cycle. The GenTrn program accepts as input an extract file and produces a transaction trigger file or *TRNFILE*. Each record in the TRNFILE represents a single transaction. The TRNFILE is then passed to the GenData program. The GenTrn program is part of the DAP procedure in PROCLIB. There is also an example of GenTrn in the GenTrnX member of JCLLIB.

LOG2PSEG utility This utility accepts as input a logo file, which is similar to a bitmap file, and produces an AFP page segment. The page segment is generally placed into a page segment library PDS referred to as PSEGLIB. This PSEGLIB must be available to your AFP printer when you print. You can convert a single logo in the PDS (use /I=logoname) or all logos in the PDS (use /I=*). Look in member LOG2PSGX of JCLLIB to find an example of this utility.

PDFKEY utility Use the PDFKEY utility to generate the encrypted passwords used in the PDF security control group.

PDS2VSAM utility This utility accepts as input a partitioned dataset, such as the DDTLIB or PMETLIB PDSs, and copies those members into a VSAM KSDS. Look in members PDS2VSDX, PDS2VSJX, and PDS2VSPX of JCLLIB for examples of this utility.

RCP2VSAM utility This utility accepts as input a set recipient table, usually the SETRCPTB member in the DEFLIB PDS, and copies this table into a VSAM KSDS. When you have large set recipient tables, this conversion can speed the reading of the table. Look in member RCP2VSMX of JCLLIB to find an example of this utility.

SAMPAFP utility This utility accepts as input the name of either an AFP overlay or page segment and generates a SYSOUT file that, together with the correct USERLIB reference to this AFP resource, will print this overlay or page segment resource. Look in member SAMPAFPX of JCLLIB to find an example of this utility.

TRANSLAT utility This utility accepts as input a tokenized message file (MSGFILE) that is output from the GenTrn, GenData, GenPrint, or GenArc program. This utility then produces a readable error file. The TRANSLAT utility lets you translate the tokens in the MSGFILE into various languages or simply change the contents to meet your needs. Look in member TRANSLTX of JCLLIB to find an example of this utility.

XERDNLD utility

This utility lets you get resources to a Xerox printer if you do not have other software programs to accomplish this task.

The utility accepts as input a Xerox Metacode resource, such as a font, image, or FRM file, encapsulates it with location-specific DJDE information and generally writes the resource back out to a PDS. You can then send the contents of this PDS to the JES SPOOL and target those contents to a specific Xerox printer where the resource will be stored on the printer's local disk device. The resources will then be available when needed at print time. Look in the XERDNLDX member of JCLLIB to find an example of this utility.

CHAPTER 3

Installing Documaker Server on UNIX/Linux

This guide provides detailed information on how to install and configure the system on an IBM AIX, Sun Solaris, HP-UX, and Linux environments.

You will find information on these topics:

- Overview on page 120
- What's on the Installation CDs on page 121
- Installing the System on page 122
- Checking Your Installation on page 124
- Using the Documentation on page 127
- Start-Up Tips on page 128
- Optimizing Performance on page 129
- Uploading and Downloading Resources on page 132
- Directory Structures on page 133

Overview

This chapter provides you with steps to install and test the system on your UNIX/Linux system. This overview discusses the contents of this chapter.

NOTE: See System Requirements on page 2 for information about requirements.

What's on the Installation CD

The system is available on CD-ROM for IBM AIX, Sun Solaris, HP-UX, and Intel-based Linux systems. This topic describes the contents of the Installation CD you received. For more information, see What's on the Installation CDs on page 121.

Installing the System

This topic explains how to install the system. For more information see Installing the System on page 122.

Checking Your Installation

After you install your system, use the sample resources to check your installation. For detailed instructions, see Checking Your Installation on page 124.

Using the Documentation

This topic describes the on-line documentation included on the Installation CD and explains how to use it. For more information, see Using the Documentation on page 127.

Start-Up Tips

This topic provides additional information on using the system in a UNIX/Linux environment. For more information, see Start-Up Tips on page 128.

Optimizing Performance

Once you have installed and tested your system, review this topic to improve system performance. For more information, see Optimizing Performance on page 129.

Uploading and Downloading Resources

Sharing resources between the different platforms is the subject of this topic. Once you have installed and tested your system and you are ready to take developed master resource libraries from Windows to UNIX/Linux, you can review this topic to get details transfer methods and formatting. For more information, see Uploading and Downloading Resources on page 132.

Directory Structures

This topic describes the layout of the files in a typical installation. For more information, see Directory Structures on page 133.

What's on the Installation CDs

The Installation set includes these CD-ROMs:

CD	Description
CD-1	Installation routines, program files, documentation, and sample MRLs
CD-2	Fonts

NOTE: Registered users can go to the Support site where you can logon, review current patch reports, and download installation files which you can transfer to UNIX systems for installation.

Installation routines

Depending on your license, you will find one of the following installation programs on CD-1 along with the MRL installation program:

	AIX	Solaris	HP-UX	Linux
Run time environment	CD1DocumakerFullSystemAIX113Pxx.exe	CD1DocumakerFullSystemSPA113Pxx.exe	CD1DocumakerFullSystemHPX113Pxx.exe	CD1DocumakerFullSystemLNX113Pxx.exe
Master resource libraries (MRLs)	setuprp1.aix setuprpex1.aix	setuprp1.spa setuprpex1.spa	setuprp1.hpx setuprpex1.hpx	setuprp1.lnx setuprpex1.lnx
SDK	setupsdk.aix	setupsdk.spa	setupsdk.hpx	setupsdk.lnx

Where *Pxx* indicates the patch number.

See Installing the System on page 122 for instructions on how to install the system.

On-line documentation

On CD-1 you will find directories which contain on-line manuals and other documentation. Use Adobe's Acrobat Reader to view and print the on-line manuals. Here is a list of the directories:

Directory	Contains
\doc\api	Technical documentation about system libraries and application program interfaces (APIs). You can view this information using your browser.
\doc\dm-rp	The user documentation in PDF format.
\doc\release	The Features and Enhancements Guide for this release.

See Using the Documentation on page 127 for information on how to use the documentation.

Installing the System

Follow these steps to install the following on a computer running AIX, Solaris, HP-UX, or Linux (Intel-based):

- Documaker
- AGFA, PCL, and PostScript fonts
- AFP fonts
- Xerox fonts

A separate and optional installation program is provided that contains a sample resource library which includes test results and sample data. The RPEX1 resources are designed for the insurance market.

NOTE: Re-installation of the Run Time Environment *does not* overwrite your INI files.

Before You Begin

Make sure you have full directory access permissions (read/write/execute) to the directory into which you will be installing Documaker Server.

Setting up libraries

If you install the system into an existing directory, the current library setup (INI files) is retained. If you install the system into a new directory, you must set up all of your libraries—just as you would for a new customer.

Setting up users

Follow these steps to install the system:

1. Log on to the UNIX/Linux system using the account you want to use for the installation. By default the installation process installs the system into a subdirectory in your home directory with a name based on the release number.

   For example, if you are installing the system using an account named *fap* that has a home directory of */home/fap*. the installation installs the system in the */home/fap/rel110* directory. You can, however, override the default.

2. Insert CD-1 in your CD-ROM drive on the UNIX/Linux server. Some systems may be set up to require system administrator's access to access the CD-ROM drive and mount the CD. Check with your system administrator if necessary.

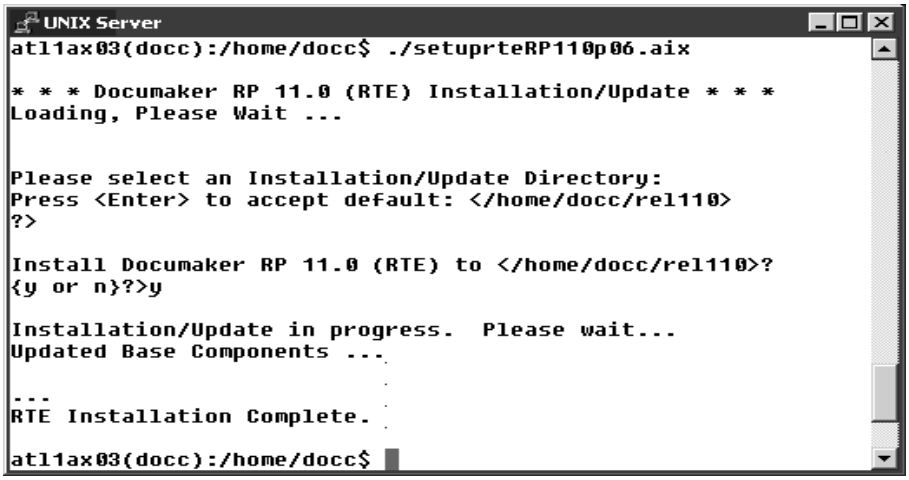
NOTE: Optionally, you can insert the CD into the CD-ROM drive of a Windows workstation and transfer the installation files from the CD to the UNIX or Linux server using FTP in binary mode or with a network drive mapping using NFS (Network File System) or SaMBa (SMB/CIFS file sharing protocol).

You may need a knowledge of file permissions so you can make the installation executable if the transfer method changes the permissions.

3. Open the readme.txt file on the CD using a text editor and review it. This file contains any additional information that could not be added to this document before publication.

4 Run the installation program from the CD or copy the installation files to a different location and run the installation program from there using either a Korn or a Bash shell prompt.

5 The initial installation screen describes the type of installation, displays the default directory the installation will use, and lets you override the installation location. Here is an example of the RTE installation screen with an override directory of */home/docc/rel110* being entered:



Checking Your Installation

The system includes a sample resource library. The RPEX1 resources are designed for the insurance market. You can check your installation by processing the resources in the sample resource library you received. The sample library includes an example directory which you can use to test against the results you get when you run the following tests.

These examples use scripts such as *run.sh* that pre-set the PATH and library PATH environment variables so the Dynamically Shared Objects (DSOs) and executables can be located at run time.

To manually run the programs or utilities (such as gentrn, gendata, genprint, ovlcomp, fap2cfa,and so on) at a shell command prompt, set these environment variables by prepending the PATH and the appropriate library PATH environment variables with the location of these files (such as LD_LIBRARY_PATH for Solaris and Linux, SHLIB_PATH for HP-UX, and LIBPATH for AIX).

Some implementations may set these environment variables using a .profile for the shell so they are pre-set automatically when a Korn shell is run. Please check with your system administrator for options.

This is an example of the command-line shell prompt commands to enter if you installed the system into the /home/fap/rel110 directory:

Solaris and Linux

```
PATH=/home/fap/rel110/bin:$PATH
LD_LIBRARY_PATH=/home/fap/rel110/bin:$LD_LIBRARY_PATH
export PATH LD_LIBRARY_PATH
```

AIX

```
PATH=/home/fap/rel110/bin:$PATH
LIBPATH=/home/fap/rel110/bin:$LIBPATH
export PATH LIBPATH
```

HP-UX

```
PATH=/home/fap/rel110/bin:$PATH
SHLIB_PATH=/home/fap/rel110/bin:${SHLIB_PATH:-""}:
export PATH SHLIB_PATH
```

Note: Carefully follow these steps, otherwise, the results may be unpredictable.

Testing with the RPEX1 Sample Resources

The system includes a shell script file (run.sh) which lets you process the RPEX1 sample resources. There is are specific options and arguments you can use to select the Documaker Server modules (gentrn, gendata, genprint, and genarc) in single- or multi-step mode and select the type of printer output, such as PCL, PostScript, Xerox, Metacode and AFP.

The shell script file executes the Documaker Server modules (gentrn, gendata, genprint, and genarc) in the appropriate mode for the specified printer type and copies the appropriate INI files and printer overlays into their proper directories (see the Documaker Server System Reference for more information).

Processing the RPEX1 Sample Resources

To run the system through a processing cycle, follow these steps:

1 Go to the mstrres/rpex1 directory where you installed the MRL examples.

2 At the Korn or Bash shell prompt, enter the appropriate command:

To produce output for	For multi-step, enter	For single-step, enter
all printer types	**run.sh a m**	**run.sh a s**
AFP printers	**run.sh f m**	**run.sh f s**
Xerox Metacode printers	**run.sh x m**	**run.sh x s**
PostScript printers	**run.sh s m**	**run.sh s s**
PCL printers	**run.sh p m**	**run.sh p s**

NOTE: You can print the usage for the run.sh script to get all the options. Also, you can run the run.sh script without arguments to get a menu system.

Printing the RPEX1 Output

After running the system, the printer ready files can be found in the ../rpex1/data directory.

NOTE: See the Documaker Server System Reference for detailed instructions on printing to AFP, Metacode, PCL, or Postscript printers.

Testing WIP with RPEX1

The RPEX1 library is set up to place two transactions into WIP. Follow these steps to perform the test:

NOTE: If you are licensed for WIP the following steps apply. If, however, you *are not* licensed for WIP, do not perform these steps.

To process the WIP transactions on UNIX/Linux and view and edit the WIP transactions with the Entry module you must have a drive letter on a Windows workstation mapped to the UNIX/Linux directory where the system was installed using NFS or SaMBa. The Windows workstation installation of the Entry system can then directly access the WIP transactions.

1. At a UNIX/Linux Korn or Bash shell prompt, enter this command:

   genwip

2. If using a NFS or SaMBa mapped drive on Windows, start the Entry module on Windows by entering:

   afemnw32 /mode=wip

3. Choose the WIP, WIP List option.

4. Select transaction *2234567* and click Ok.

Viewing Archived RPEX1 Forms

Optionally, you can archive the bills previously printed by following these steps:

NOTE: If you are licensed for Archive Retrieval the following steps apply. If, however, you *are not* licensed for Archive Retrieval, do not perform these steps.

---

1. At a UNIX/Linux Korn or Bash shell prompt, enter this command:

   genarc

2. If using a NFS or SaMBa mapped drive on Windows, start the Entry module on Windows by entering this command:

   afemnw32 /mode=retrieve

3. Choose the Retrieve, Formset option to view the archived bills.

Using the Documentation

The \doc directory contains the on-line documentation available for this release. In this directory on the CD, you will find Portable Document Format (PDF) files which you can view or print using the Adobe® Acrobat™ Reader™.

In the \doc\dm-rp directory you will find the various manuals, such as:

- Docucreate Supervisor Guide
- Docucreate User Guide
- Documaker Supervisor Guide
- Documaker User Guide
- Documaker Server System Reference
- Docutoolbox Reference
- Rules Reference
- DAL Reference
- product release information
- Documaker Server Installation Guide

For best results, make sure you have the latest version of Acrobat Reader. You can download the free reader at Adobe's web site: www.adobe.com.

NOTE: If you have Acrobat Reader installed on your computer, uninstall the current version before you install an updated version.

You can use the Acrobat Reader to view the documentation on the Installation CD or you can copy the various files onto your hard disk and view them from there.

Start-Up Tips

Here are some additional tips to help you get your system up and running:

Using long file names

File names in UNIX and Linux can be up to 256 characters long or longer. You should, however, limit the names of resource files, such as FAP and LOG files, to 100 characters or less, excluding the path.

Documaker Server software imposes no limits on the length of the names use for the output files, such as print streams, so you can name those files however you like.

Naming conventions for printers in Windows

Windows does not use the *LPT* naming convention, such as LPT1, LPT2, and so on, for printers. Instead, it uses the *server name**printer* convention. For Documaker, you may need to modify the FSISYS.INI and FSIUSER.INI files. You must log on as the administrator and establish your printer connections to set up the naming conventions for your printers.

Optimizing Performance

This topic will help you configure your system for optimum performance. To gather the following recommendations, we first created benchmarks on a test system. Then, by changing different parameters of that system, we measured performance gains or losses. in our benchmark testing. Here are some of the terms we used during this exercise:

CPU Time The amount of time that a program, such as GenTrn, GenData, or GenPrint, uses the CPU.

EXCPs Execute Channel Programs. We have used these counts as basic measurements of I/O activity.

Wall Clock Time The elapsed time, as measured from the time a program begins to the time that the program ends. This wall clock time can vary significantly from one run to another.

Batch Window Most installations have specific times of the day or night when large batch processes, like this system, are scheduled to run, such as through *cron*. The time frame in which these processes run is sometimes referred to as the *batch window*.

A batch window is measured in wall clock time, such as from 10:00 pm to 5:00 am. Your system installation should run fast enough to complete its processing within the batch window.

Most, but not all, of the following recommendations are the result of many tests and subsequent improvements designed for a hypothetical user. The characteristics of Documaker Server implemented for this hypothetical user are as follows:

- Extract file with large record length (approximately 25,000 bytes/record).
- Form sets composed with large number of individual images.
- Large number of different recipients (approximately 300).
- Moderate number of transactions (approximately 4,000)

NOTE: You can find additional information on optimizing performance in the Documaker Server System Reference.

Setting Your FSISYS INI Options

The following options attempt to minimize the repeated opening and closing of frequently used files by retaining, or caching, file handles and file data. In many cases the defaults are sufficient but for specific cases, where many different images are used, these caching values may be increased to improve performance.

Caching FAP files In some cases, FAP files (images) are loaded as the GenData program runs. The cache feature keeps frequently used FAP files available for re-use. The CacheFAPFiles option is specified in:

```
< Control >
    CacheFAPFiles = 100
```

The default is 100.

Accept the default value *unless* you are loading FAPs in GenData using the CompileInstream option, you are using more than 100 FAP files or logos, or both.

DDT files

Data definition table (DDT) files are loaded during as the GenData program runs. The cache feature keeps frequently used DDT file available for re-use. The RuleFilePool option is specified in:

```
< Control >
    RuleFilePool = 100
```

The default is 100.

Accept the default value unless you are using more than 100 DDTs.

Caching Xerox PMET files

MET files contain pre-compiled Xerox Metacode information produced by the FAP2MET utility. The GenPrint program loads MET files as necessary. The cache option keeps frequently used MET files available for re-use. The CacheFiles option is specified in:

```
< PrtType:XER >
    CacheFiles = 100
    CompileInstream = No
```

The default is 100.

Accept the default value *unless* you are using pre-compiled FAP files, more than 100 FAP files or logos, or both.

Using AFP Overlays

For best performance, you should run the FAP2OVL utility program to compile FAPs into AFP overlays. Tell the system to use the overlays by specifying:

```
< PrtType:AFP >
    SendOverlays   =   Yes
```

Use the PSF librarian to add printer resources to the printer.

Logging Options

LogTransactions option

The GenTrn, GenData, and GenPrint programs optionally place transaction information into a LOG file. In most situations, this information is not needed. The LogTransactions option is specified in:

```
< Control >
    LogTransactions = No
```

The default is Yes. For optimum performance, specify No.

LogToConsole option

The GenTrn, GenData, and GenPrint programs optionally store transaction information. In most situations, this information is not needed. The LogToConsole option is specified in:

```
< Control >
    LogToConsole = No
```

The default is Yes. For optimum performance, specify No.

Debug Options

In the If_Rule control group, the Debug_If option helps you solve problems when using the IF rule:

```
< If_Rule >
    Debug_if = No
```

The default is Yes. For optimum performance, specify No.

Run Options

In the RunMode control group, you have these run time options:

```
< RunMode >
    Download FAP = No
    LoadCordFAP  = No
    CompiledFAP  = No
```

The defaults are shown above.

For optimal performance, set the DownloadFAP option to No, the LoadCordFAP option to No, and the CompileFAP option to Yes.

You can learn more about these options in the Documaker Server System Reference.

Other Options

MaxRecsPerTransaction option

```
< ExtractFile >
    MaxRecsPerTransaction  =  nn
```

The default is zero (0) and there is no maximum. Be careful using this option. You might want to use this option if you are sure each record in the extract file corresponds to a transaction.

AliasPrintBatches option

```
< ExtractFile >
    AliasPrintBatches  =  Yes
```

The default is No. Use the default.

Uploading and Downloading Resources

Transferring Resource Files Between UNIX/Linux and Windows

You can use FTP to transfer files from Windows to UNIX and from UNIX to Windows. The important thing to remember is to use the correct mode (binary or ASCII) for the files.

Other options to transfer files between these platforms are available such as using mapped network drive resources such as NFS and SaMBa. This method lets you map a directory on UNIX directly to a Windows workstation. When using this method, the transfer mode is always binary by default.

Uploading a Library from PC to UNIX

Text files such as INI, DFD, DDT, and FAP should be uploaded in ASCII mode if using FTP. Compiled files should be loaded in binary mode. For example, FRM files for Xerox must be uploaded in binary mode after they are compiled using the FAP2FRM utility on a PC.

Overlays for PCL and PostScript can be compiled by the OVLCOMP utility on a PC and then uploaded to UNIX in binary mode or they can be produced directly on UNIX/Linux with the OVLCOMP executable.

NOTE: MET and CFA files are platform dependent, therefore they must be compiled on UNIX. Be sure to also use the same version of the system to compile and use these files.

Downloading Print Streams from UNIX to PC

All types of print streams (PCL, PST, AFP, and Xerox) from the GenPrint program should be downloaded to PC in binary mode if you are going to print from a Windows workstation.

PCL print stream files, once transferred to a Windows workstation, can be printed to a local or network printer using this command:

copy /b pclbat1 lbt1

PST print stream files, after successful transfer to a Windows workstation, can be printed to a local or network printer using the following command if the DownloadFonts option in the PrtType:PST control group is set to No:

copy /b rel103sm.pst+ pstbat1 lpt1

You can send an AFP print stream to an AFP printer through PSF/2 from an OS/2 workstation.

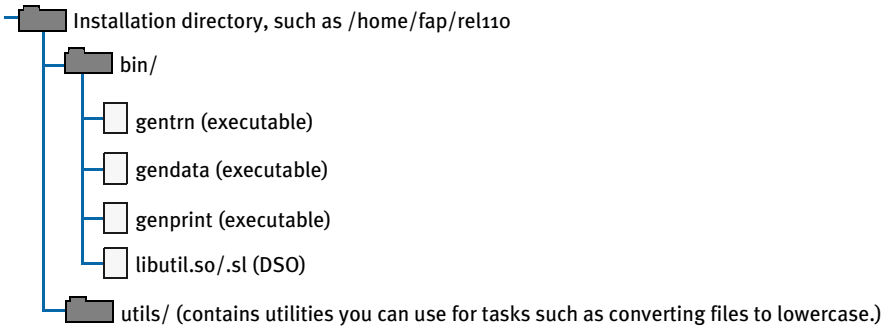
Xerox print stream can be sent to Xerox printer through a connected workstation running BARR software.

Directory Structures

The installation programs create these directory structures:

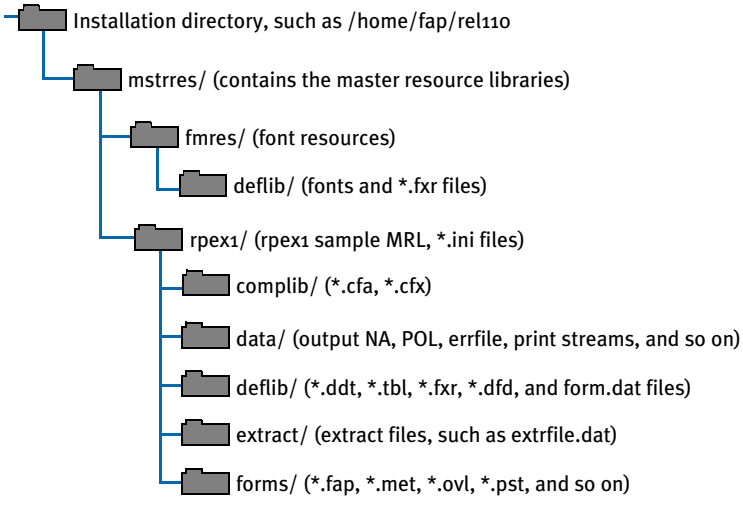
Run Time Environment (RTE)

The run time environment consists of these subdirectories under the directory where the system is installed:



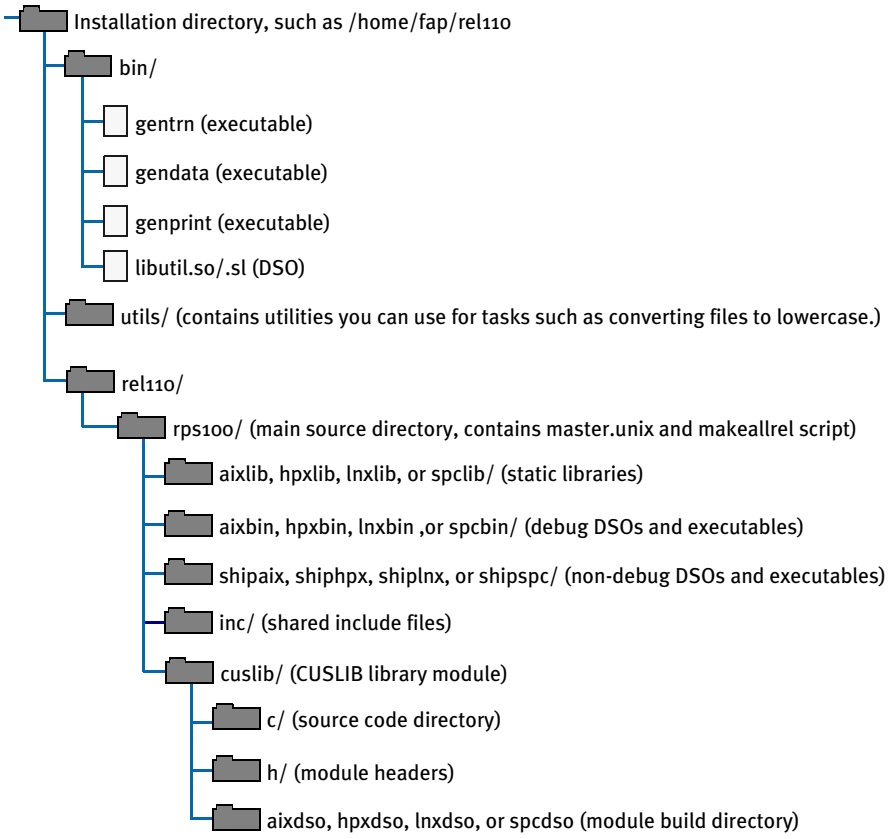
Master Resource Library (MRL)

The MRL installation creates these subdirectories under the directory where the system is installed:



Software Development Kit (SDK)

The SDK installation creates the follow subdirectories under the directory where the system is installed:



CHAPTER 4

Installing Documaker Server on Windows

This guide provides detailed information on how to install and configure the system in a Windows (Windows 2000, Windows XP, or Windows Vista) environment.

You will find information on these topics:

- Overview on page 136
- What's on the Installation CD on page 137
- Installing the System on page 138
- Checking Your Installation on page 139
- Using the Documentation on page 142
- Start-Up Tips on page 143
- Optimizing Performance on page 144

Overview

This chapter provides you with steps to install and test the system on your Windows system. This overview discusses the contents of this chapter.

Note: See System Requirements on page 2 for information about system requirements.

What's on the Installation CD

The system is available on CD-ROM for Windows systems. This topic describes the contents of the Installation CD you received. For more information, see What's on the Installation CD on page 137.

Installing the System

This topic explains how to install the system. For more information see Installing the System on page 138.

Checking Your Installation

After you install your system, use the sample resources to check your installation. For detailed instructions, see Checking Your Installation on page 139.

Using the Documentation

This topic describes the on-line documentation included on the Installation CD and explains how to use it. For more information, see Using the Documentation on page 142.

Start-Up Tips

This topic provides additional information on using the system in a Windows environment. For more information, see Start-Up Tips on page 143.

Optimizing Performance

Once you have installed and tested your system, you can review this topic to find ways to improve system performance. For more information, see Optimizing Performance on page 144.

What's on the Installation CD

The Installation set includes a CD-ROMs with installation routines, program files, documentation, and sample MRLs. Depending on your license, the installation routine installs Documaker Server, Docucreate, Documaker Studio, and/or Documaker Workstation depending on your license type.

See Installing the System on page 138 for instructions on how to install the system.

NOTE: Registered users can go to the Support site where you can logon, review current patch reports, and download installation files which you can transfer to your system for installation.

On-line documentation

On CD-1 you will also find directories which contain on-line manuals and other documentation. Use the readme.htm file as a guide to the various documents included.

See Using the Documentation on page 142 for information on how to use the documentation.

Installing the System

Follow these steps to install the following on a personal computer running Windows:

- Docucreate
- Documaker
- Entry, WIP, and Archive Retrieval systems of Documaker Workstation (optional, depending on your license)
- AGFA, PCL, and PostScript fonts
- AFP fonts
- Xerox fonts

The installation also installs the RPEX1 sample resource library which includes test results and sample data. The RPEX1 resources are designed for the insurance market.

NOTE: Re-installation *does not* overwrite your INI files.

Before You Begin

Be sure to log in as *Administrator* and make sure you have full directory access permissions to the c:\windows\profiles\all users folder, as well as to the registry.

Using NTFS drive formats

When using NTFS drive formats, the logon name must be assigned all privileges and permissions to the Windows operating system folder and all sub-folders within, as well as to make changes to the Windows registry.

Setting up libraries

If you install the system into an existing directory, the current library setup is retained. If you install the system into a new directory, you must set up all of your libraries—just as you would for a new customer.

Installation

Follow these steps to install the system:

1. Insert CD-1 in your CD-ROM drive.
2. If autoplay does not start the installation, select the Run option from the Start menu and use the Browse button to select the CD1DocumakerFullSystemW32113Pxx.exe file, where *Pxx* indicates the patch number. Then click Ok.
3. Follow the instructions that appear on your screen to install the software. The installation creates folders and icons on your Start\Programs menu.

After you install the system, restart your computer so the various system settings can take affect. If you install the software on a network, you will need to set up icons and a program group for each user. Refer to your operating system's documentation for information on how to do this.

Checking Your Installation

The system includes a sample resource library. The RPEX1 resources are designed for the insurance market.

You can check your installation by processing the resources in the sample resource library you received. The sample libraries include an example directory which you can use to test against the results you get when you run the following tests.

NOTE: Carefully follow these steps, otherwise, the results may be unpredictable.

Testing with the RPEX1 Sample Resources

The system includes several batch files which let you process the RPEX1 sample resources. There is a specific batch file that lets you select Documaker Server modules (GenTrn, GenData, GenPrint, and GenArc) in single- or multi-step mode for each type of printer, such as PCL, PostScript, Xerox Metacode, and AFP.

The batch files execute the Documaker Server modules (GenTrn, GenData, GenPrint, and GenArc) in the appropriate mode for the specified printer type and copies the appropriate INI files and printer overlays into their proper directories (see the Documaker Server System Reference for more information).

Processing the RPEX1 Sample Resources

To run the system through a processing cycle, follow these steps:

1. Go to the \rpex1 directory.
2. At the operating system prompt, enter the appropriate command:

To produce output for	For multi-step, enter	For single-step, enter
all printer types	**run a m**	**run a s**
AFP printers	**run f m**	**run f s**
Xerox Metacode printers	**run x m**	**run x s**
PostScript printers	**run s m**	**run s s**
PCL printers	**run p m**	**run p s**

Printing the RPEX1 Output

To print the output you generated when you ran the system, follow the instructions below which match the type of printer.

NOTE: See the Documaker Server System Reference for detailed instructions on printing to AFP, Metacode, PCL, or Postscript printers.

For an AFP printer

Copy to the printer all of the files located in the \rpex1\forms\afpovl directory which have an *afp* extension. (Use the PSF librarian to add printer resources to the printer)

After you have added all of the printer resources to the printer, use the *aprint* command to print the AFPBAT1 file, located in the \rpex1\data directory.

For a Metacode printer

Copy all files located in the \rpex1\forms\xerovl which have a *dat* extension to the \spool directory for the printer.

After you have copied these printer resources to the \spool directory, copy the XERBAT1 file, located in the \rpex1\data directory, to the \spool directory for the Xerox printer.

For a PCL printer

Copy the rel103SM.FNT font file located in \mstrres\fmres\deflib directory to the printer along with the PCLBAT1 file, which is located in the \data directory. You can do this by entering the following command from the \rpex1 directory:

copy /b ..\fmres\deflib\rel103sm.fnt + DATA\pclbat1 lpt1

NOTE: Depending on your printer environment, you may need to substitute *servername**queuename* for *lpt1* in the command shown above.

For a PostScript printer

Copy the rel103SM.PST font file located in \mstrres\fmres\deflib directory to the printer along with the PSTBAT1 file, which is located in the \data directory. You can do this by entering this command from the \rpex1 directory:

copy /b ..\fmres\deflib\rel103sm.pst + DATA\pstbat1 lpt1

NOTE: Depending on your printer environment, you may need to substitute *servername**queuename* for *lpt1* in the command shown above.

Testing WIP with RPEX1

The RPEX1 library is set up to place two transactions into WIP. Follow these steps to perform the test:

NOTE: If you are licensed for WIP the following steps apply. If, however, you *are not* licensed for WIP, do not perform these steps.

1 At an operating system prompt, enter this command:

genwpw32

2 Start the Entry module by entering:

afemnw32 /mode=wip

3 Choose the WIP, WIP List option.

4 Select transaction 2234567 and click Ok.

Viewing Archived RPEX1 Forms

Optionally, you can archive the bills previously printed by following these steps:

NOTE: If you are licensed for Archive Retrieval the following steps apply. If, however, you *are not* licensed for Archive Retrieval, do not perform these steps.

1 At an operating system prompt, enter this command:

genacw32

2 Start the Entry system by entering this command:

afemnw32 /mode=retrieve

3 Choose the Retrieve, Formset option to view the archived bills.

Using the Documentation

The \doc directory contains the on-line documentation available for this release. In this directory on the CD, you will find Portable Document Format (PDF) files which you can view or print using the Adobe® Acrobat™ Reader™.

In the \doc\dm-rp directory you will find the various manuals, such as:

- Docucreate Supervisor Guide
- Docucreate User Guide
- Documaker Supervisor Guide
- Documaker User Guide
- Documaker Server System Reference
- Docutoolbox Reference
- Rules Reference
- DAL Reference
- product release information
- Documaker Server Installation Guide

For best results, make sure you have the latest version of Acrobat Reader. You can download the free reader at Adobe's web site: www.adobe.com.

NOTE: If you have Acrobat Reader installed on your computer, uninstall the current version before you install an updated version.

You can use the Acrobat Reader to view the documentation on the Installation CD or you can copy the various files onto your hard disk and view them from there.

Start-Up Tips

Here are some additional tips to help you get your system up and running:

Using long file names

In Windows, file names, including the path, can be up to 256 characters long. You should, however, limit the names of resource files, such as FAP and LOG files, to 100 characters or less, excluding the path.

Documaker software imposes no limits on the length of the names use for the output files, such as print streams, so you can name those files however you like.

Naming conventions for printers in Windows

Windows does not use the LPT naming convention, such as LPT1, LPT2, and so on, for printers. Instead, it uses the \\server name\printer convention. For Documaker, you may need to modify the FSISYS.INI and FSIUSER.INI files. You must log on as the administrator and establish your printer connections to set up the naming conventions for your printers.

Displaying objects in color in Windows

By default, variable fields use the color red. If, when you open a FAP file in Windows, the first variable field appears in red and the other variable fields appear in dark red, you can change the colors using the Windows Control Panel. To change the color settings, select My Computer, Control Panel, and then Display. The Display Properties window appears. Click the Settings tab. In the Color Palette field, choose High Color (16 bit). Then, reboot your system.

Optimizing Performance

This topic will help you configure your system for optimum performance. To gather the following recommendations, we first created benchmarks on a test system. Then, by changing different parameters of that system, we measured performance gains or losses. in our benchmark testing. Here are some of the terms we used during this exercise:

CPU Time. The amount of time that a program, such as GenTrn, GenData, or GenPrint, uses the CPU.

EXCPs. Execute Channel Programs. We have used these counts as basic measurements of I/O activity.

Wall Clock Time. The elapsed time, as measured from the time a program begins to the time that the program ends. This *wall clock time* can vary significantly from one run to another.

Batch Window. Most installations have specific times of the day or night when large batch processes, like this system, are scheduled to run. The time frame in which these processes run is sometimes referred to as the *batch window*. A batch window is measured in *wall clock time*, such as from 10:00 pm to 5:00 am. Your system installation should run fast enough to complete its processing within the *batch window*.

Most, but not all, of the following recommendations are the result of many tests and subsequent improvements designed for a hypothetical user. The characteristics of Documaker Server implemented for this hypothetical user are as follows:

- Extract file with large record length (approximately 25,000 bytes/record).
- Form sets composed with large number of individual images.
- Large number of different recipients (approximately 300).
- Moderate number of transactions (approximately 4,000)

Note: You can find additional information on optimizing performance in the Documaker Server System Reference.

Setting Your FSISYS INI Options

Caching Options

The following options attempt to minimize the repeated opening and closing of frequently used files by retaining, or caching, file handles and file data. In many cases the defaults are sufficient but for specific cases, where many different images are used, these caching values may be increased to improve performance.

Caching FAP files

In some cases, FAP files (images) are loaded as the GenData program runs. The cache feature keeps frequently used FAP files available for re-use. The CacheFAPFiles option is specified in:

```
< Control >
    CacheFAPFiles = 100
```

The default is 100.

Accept the default value *unless* you are loading FAP files in GenData using the CompileInstream option (set to Yes), you are using more than 100 FAP files or logos, or both.

DDT files

Data definition table (DDT) files are loaded during as the GenData program runs. The cache feature keeps frequently used DDT file available for re-use. The RuleFilePool option is specified in:

```
< Control >
    RuleFilePool = 100
```

The default is 100. Accept the default value unless you are using more than 100 DDT files.

Using/Caching Xerox PMET files

MET files contain pre-compiled Xerox Metacode information produced by the FAP2MET utility. The GenPrint program loads MET files as necessary. The CacheFiles option keeps frequently used MET files available for re-use. The CacheFiles option is specified in:

```
< PrtType:XER >
    CacheFiles     = 100
    CompileInstream= No
```

The default for the CacheFiles option is 100. Accept the default value *unless* you are using pre-compiled FAP files and more than 100 FAP files or logos or both.

Using AFP Overlays

For best performance, you should run the FAP2OVL utility program, compiling FAPs into AFP overlays. Tell the system to use the overlays by specifying:

```
< PrtType:AFP >
    SendOverlays = Yes
```

Use the PSF librarian to add printer resources to the printer.

Logging Options

LogTransactions option

The GenTrn, GenData, and GenPrint programs optionally place transaction information into a LOG file. In most situations, this information is not needed. The LogTransactions option is specified in:

```
< Control >
    LogTransactions = No
```

The default is Yes.

For optimum performance, specify *No*.

LogToConsole option

The GenTrn, GenData, and GenPrint programs optionally store transaction information. In most situations, this information is not needed. The LogToConsole option is specified in:

```
< Control >
    LogToConsole = No
```

The default is Yes.

For optimum performance, specify *No*.

Debug Options

If_Rule control group

```
< If_Rule >
    Debug_if = No
```

The default is Yes.

For optimum performance, specify *No*.

Run Options

RunMode control group

You have these runtime options:

```
< RunMode >
    Download FAP = No
    LoadCordFAP  = No
    CompiledFAP  = Yes
```

The defaults are...

```
DownloadFAP = No
LoadCordFAP = No
CompiledFAP = No
```

Set the DownloadFAP option to *No*, the LoadCordFAP option to *No*, and the CompileFAP option to *Yes* for the best performance.

You can learn more about these options in the Documaker Server System Reference.

Other Options

MaxRecsPerTransaction option

```
< ExtractFile >
    MaxRecsPerTransaction = nn
```

The default is zero (0) and there is no maximum. Be careful using this option. You might want to use this option if you *know* that each record in the extract file corresponds to a transaction.

AliasPrintBatches option

```
< ExtractFile >
    AliasPrintBatches = Yes
```

The default is No. Use the default.

Index

Numerics

3270 emulator program 112

A

access permissions 122
Adobe Acrobat 121
Adobe Acrobat Reader 127, 142
Advanced Function Presentation 50
AFGJOB member 95
AFP
    batch files for RPEX1 139
    configuring 43
    fonts 122
AFP2MVS utility 115
AFPBAT1 file 140
AFPCOPY utility 115
AFPFMDEF utility 115
AFPFMDFX job 50
AGFA
    fonts 122
AIX
    Documaker system requirements 7
    Docupresentment requirements 11
    Docupresentment web server requirements 13
    installation 121
    system requirements 2

AliasPrintBatches option 90, 131, 147
Allocating AFP-related Datasets 49
ALLOCDB2 and COMPDB2 61
ALLOCDDT job 93
ALLOCJDT job 95
and host archive 60
ANSI code page 113
APPIDX table 60, 63
aprint command 140
Archival control group 61
ARCHIVE table 60
archived bills
    viewing from RPEX1 126, 141
ArcRet control group 61
ASCII files 111
awk utility 7

B

BARR interface 83
BARR software 132
BARR2MVS utility 115
BARRWRAP utility 115
BARRWRPX utility 83
batch files
    for processing RPEX1 139
Batch window 84, 129, 144
block sizes 86

C

CacheFAPFiles option 129, 145
CacheFiles option 130, 145
caching options
    for DDT files 88, 130, 145
    for FAP files 88
    summary 145
carriage returns 112
CD1DocumakerFullSystemAIX113Pxx.exe 121
CD1DocumakerFullSystemHPX113Pxx.exe 121
CD1DocumakerFullSystemLNX113Pxx.exe 121
CD1DocumakerFullSystemSPA113xx.exe 121
CDs
    contents 121
changing
    the FSISYS.INI file (AFP) 50
    the JCL 51, 63
    the JCL (Metacode) 49
    the Xerox JSL 43
checking
    the JCL 70
    your FSIUSER and FSISYS settings 67
    your installation 67
code page 1004 113
code page 37 113
colors 143
compile options 85
CompileInStream option 47
compilers
    system requirements 5, 6, 7, 8, 9, 11, 12
COMPLIB 111
configuring
    host archive using DB2 60
    your print environment 43
    your system for AFP printers 49
CPCNV utility 113, 115
CPU time 84, 129, 144
CreateIndex option 62
CreateTable option 62
Creating DB2 Objects 60
Creating NAFILE and POLFILEs as VSAM KSDSs 101

D

DAP procedure 49, 51
    GenPrtX job 51

DAPRUN job 49, 51, 70
Database option 62
DB2
    Command Processor program 80
    database management system 60
    DefaultTag option 62
    table 62
DB2_FileConvert control group 61, 62
DBHANDLER
    DB2 control group 62
DDT files
    caching 88, 130, 145
DDTVSAM DD statement 94
DDTVSAM option 94
debug options 90, 131, 146
DefaultTag option 62
defining the extract file as a VSAM ESDS 91
DEFLIB 86
DEFLIB PDS 93, 95, 99
DJDEIden entry 44
DJDEOffset entry 44
DJDESkip entry 44
Documaker
    system requirements 2
Documaker Server 122
    requirements 4
Documanage
    system requirements 2
documentation 121, 127
Docupresentment
    requirements 10
DOS command line interface 112
DownloadFAP option 47, 90, 131, 146
DownloadFonts option 132
downloading
    print streams 132
downloading resources 111
DSNHC procedure 61

E

EBCDIC
    code page 113
    uploading files 111
encrypted passwords 117
EXCPs 84, 129, 144
extract files 86

F

FAP files
    caching 88
    caching options 129, 145
    compiled 111
    using pre-compiled 89, 130, 145
FAP2AFP utility 115
FAP2CFA utility 111, 116
FAP2MET utility 47, 89, 114, 116, 130, 145
FAP2OVL utility 50, 116
FAPLIB 87
FB 86
FBS 86, 91, 101
FDT2CFA utility 111, 116
file names 128
files
    transferring 111
    types and characteristics 86
FIXOFFS utility 116
FNT files
    for logos 48
FONTLIB PDS 56
fonts
    AFP 50
    Metacode 48

FormDef 50
fseek 86
FSISYS
    changing the JSL for Metacode printers 43
    checking your settings 67
    DDTVSAM option 94, 96
    METVSAM option 98
    options 129, 145
    performance options 88
    VSAMNA option 101
    VSAMRCPTB option 100
FSIUSER
    archive settings 61
    checking your settings 67
FSIVER utility 116

G

GenArc program 60, 124, 139
    and DSN 80
    defined 116
GenArcX member 63
GenData program 117, 124, 139
GENDATAL job 40
GenDataX job 94, 96
GENERAFP job 51
GENERAFP member 83
GENERXER job 49
GENERXER member 83
GENLOGOS job 48
GenPrint program 117, 124, 139
GENPRTL job 40
GenPrtX job 51
GenTrn program 117, 124, 139
graphics card 48

H

handling international characters 113
HEAP option 85
HP-UX 119, 120, 121, 124
    system requirements 2

I

IBM
    DB2 database management system 60
    Print Services Facility 50
IDCAMS job 101
IEBGENER 83
IF_Rule control group 90, 131, 146
ImageOpt entry 44
IMG files
    graphics cards 48
INI files
    installation 122
INLINE option 85
installation
    programs 121

installing the system 122, 138
international characters 113

J

Java
    and web servers 15
JCLLIB PDS 93, 97, 99
JDEName entry 44
JDLCode entry 44
JDLData entry 45
JDLHost entry 44
JDLName entry 44
JDLROffset entry 45
JDLRPage entry 45
JDLRStack entry 45
JDTVSAM DD statement 96
JDTVSAM option 96
Job Description Entry 44
JSL
    changing the Xerox 43
    requirements for Metacode 44
    viewing 43

L

libraries 108
    checking your installation 124
    checking your installation with RPEX1 139
    setting up 122
license
    installation 121
line feeds 112
linearized PDF files 57
Linux
    Docupresentment requirements 12
    Docupresentment web server requirements 13
    installation 121
    system requirements 2, 8
LOADDAP job
    listing 102
loading fonts onto a Metacode printer 48
LOG2PSEG utility 50, 117
logging options 89, 130, 146
logos
    loading 48
    making resources available 48
LogToConsole option 89, 130, 146
LogTransactions option 89, 130, 146
long file names 128, 143

M

making printer resources available
    for AFP printers 49
    for Metacode printers 48
maximum logical record length
    MaxExtReclen 91
MaxRecsPerTransaction option 90, 131, 147
Metacode
    batch files for RPEX1 139
    configuring 43
    moving PMETs 97
METVSAM option 98
modifying the FSISYS.INI file 61
moving
    DDT files into a VSAM KSDS 93
    JDT files into a VSAM KSDS 95
    PMET files into a VSAM KSDS 97
    SETRCPTB files to a VSAM KSDS 99
MultiFileLog option 55
MultiFilePrint rule 55
MVS
    programs 111
    utilities 115

N

NAFILE 87, 101
NAPOLVS member 101
networks
    system requirements 3
NEWTRN 87
NFS 122, 126
NOVALIDATEHANDLE option 85

O

ODBC
    DefaultTag option 62
Optimize option 85
optimizing performance 84, 129, 144
OS//390
    system requirements 2
overlays 50
Overview 120

P

page segments 50
page-at-a-time downloading 57
passwords 117
PCL
    batch files for RPEX1 139
    fonts 122
PCLBAT1 file 140
PDF
    creating PDF files 54
    fonts in print streams 56
    print driver 57
    using DAL scripts 55
    using the MultiFilePrint rule 55
PDFKEY utility 117
PDS2VSAM utility 117
PDS2VSDX job 93
PDS2VSJX job 95
PDS2VSPX job 97
performance
    INI options 129, 145
    optimizing 84, 129, 144
permissions 122
PMETLIB PDS 89, 97
POLFILE 87, 101
Port option 47, 51, 56
PostScript
    batch files for RPEX1 139
    fonts 122
    fonts on z/OS 56
    print streams 53
pre-compiled Metacode files 88, 97, 114
pre-compiling FAP files
    into Metacode files 47
    into overlays 50
print files 88
print streams
    downloading 132
Printer control group 46, 51
printer naming conventions 128, 143
printer resources 48
Printer1 control group 47, 51
PrinterInk entry 45
printers
    system requirements 5, 6, 7
printing
    IMG files 48
    RPEX1 output 125, 140
    with DAPRUN 83
processing
    RPEX1 sample resources 125, 139
PrtType control group
    AFP 50
    XER 43, 46
PSF
    and AFP 50
    librarian 140

PSF/2 132
PSTBAT1 file 140

Q

queue name 140

R

RCP2VSAM utility 87, 117
RCP2VSMX job 99
Receive command 112
recipient batch files 87
record formats 86
REL102SM.FNT font file 140
REL102SM.PST font file 140
Report option 85
resource libraries
    samples 122
RPEX1
    installation 138
    resources 122
    testing with 124, 139
RuleFilePool option 130, 145
run.sh 124, 125
RunMode control group 47, 90, 131, 146
running 61
    DAPRUN 71
    GenArcX 80
    GenTrnX, GenDataX, and GenPrtX 71
    GenTrnX, GenDataX, and GenPrtX separately 70
runtime options 85

S

SaMBa 122, 126
SAMPAFP utility 117
sample resources 122
Send command 112
SEQ2KSDS utility 116
server name 140
SETRCPTB
    member 87, 99
setting
    FSISYS options 88
    FSISYS.INI options 129, 145
Solaris
    Documaker Server system requirements 9
    Docupresentment requirements 11
    Docupresentment web server requirements 13
    installation 121
    system requirements 2
SQL databases
    DefaultTag option 62
start-up tips 128, 143
SYSPRINT
    DD statement 89
    running GenArcX 80
system
    resources on a PC 111
system requirements 2

T

testing
    with RPEX1 124, 139

text files 111
transferring files 112
TRANSLAT utility 117
Trigger2Archive control group 63
TrueType
    fonts on z/OS 56
TSO Terminal Monitor Program 80

U

uncompress utility 7
UNIT names 70
UNIX
    system requirements 2
uploading and downloading resources 111
using
    on-line documentation 127, 142
uudecode utility 7

V

variable fields 143
viewing
    archived RPEX1 forms 126, 141
    viewing the JSL 43
VSAM datasets 116
VSAMNA option 101
VSAMRCPTB option 100

W

wall clock time 84, 129, 144
Windows
    Docupresentment system requirements 10
    Docupresentment web server requirements 13
    system requirements 2

X

XERBAT1 file 140
XERDNLD utility 114, 118
XERLOGOS job 48
Xerox
    batch files for RPEX1 139
    configuring 43
    font names 48
    fonts 122
    image, font, and form files 114
    pre-compiled Metacode files 114

Z

z/OS
    Documaker Server system requirements 6
    system requirements 2