

IBM FileNet Image Services  
Version 4.2

## *System Reference Guide*





IBM FileNet Image Services  
Version 4.2

## *System Reference Guide*



**Note**

Before using this information and the product it supports, read the information in "Notices" on page 167.

**This edition applies to version 4.2 of IBM FileNet Image Services (product number 5724-R95) and to all subsequent releases and modifications until otherwise indicated in new editions.**

**© Copyright IBM Corporation 1992, 2011.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

## About this manual 17

Overview 17

Document revision history 17

Scope of this guide 18

Accessing IBM FileNet Image Services documentation 20

Software education 20

Comments and suggestions 20

## 1 System design and architecture 21

Server configurations 21

Combined server configuration 21

Dual or multiple storage library server configuration 22

Application server configuration 23

Entry server configuration 25

Directory structure 27

Directory contents 28

/fnsw/bin 28

/fnsw/dev/1 29

/fnsw/diag (UNIX only) 30

/fnsw/etc 30

/fnsw/lib 30

/fnsw/local (or \fnsw_loc)	31
/fnsw/local/sd	31
/fnsw/local/spool	32
/fnsw/local/tmp	32
/fnsw/local/wfl	32
/fnsw/local/<RDBMS>	32
RDBMS (/fnsw/oracle, /fnsw/db2, or \fnsw\mssql)	33
/fnsw/procs	34
/fnsw/spool (UNIX only)	34
\fnsw\setup (Windows only)	34
<b>Subsystem descriptions</b>	<b>35</b>
AT (Release Tools)	35
AR (Archiving)	35
B2 (Enterprise Backup and Restore)	35
BS (Batch Services)	35
CO (Courier Networking)	36
CS (Cache Services)	36
DM (Database Maintenance)	36
DS (Document Services)	37
EV (SystemV Applications Executive)	37
FL (FileNet LAN Analysis Tool)	37
GB (Generic Database)	38
HE (Help Error)	38
IS (Index Services)	38
LM (Software License Management)	38
MK (Multi-Keyed Files)	39
MV (SystemV Miscellaneous)	39
NC (Network Clearing House)	39
NL (National Language Services)	39

NV (System V Networking)	39
OS (OSAR Services)	40
PM (Performance Measurement Tools)	40
PR (Print Services)	40
PR (Print Services II)	40
PV (Server Print)	40
SE (Security Services)	41
SF (System Configuration Tools)	41
SU (System Utilities)	41
WQ (WorkFlo Queue Services)	41
XV (XVT Portable Toolkit)	41

## 2 Configuration Files 42

### Configuration Database 43

### Supplied Configuration Files 46

erm.msg	46
UNIX Files	47
Windows Files	47
Language Directory	47

fninfo.msg 48

trmsg.msg 49

### Generated Configuration Files 50

as\_conf.g 50

as\_conf.s 52

init.ora 53

inx\_conf 54

snmp.conf 55

MKF.ddl(s) 55

nch_dbinit	57
nch_domain	58
print_config	58
rdb.init	59
serverGroup	59
serverConfig	61
Tuning Courier timeout durations	64
Add optional recv() timeout value for worker threads	65
Add new utility "COR_set_recv_timeout"	66
Add optional timeout value for prefetch threads	67
Add new utility "COR_set_prefetch_timeout"	68
Add optional maximum prefetch thread count	69
ssn	69
tapeconfig	69
setup_config	71

## 3 Applications 72

### Application Program Descriptions 73

Backup/Restore	73
backup_daemon	73
BRBs	73
BRRs	73
Batch Entry Services	74
BESs	74
Cache Services	74
CSMs	74
CSM_daemon	74
Document Services	75
bes_commit	75



DLIs	75
DLSs	75
DOCs	76
ds_init	76
ds_notify	76
dbc_commit	76
OSIs	76
OSSs	77
RMKs	77
rmt_commit	77
sas	77
Fax Services	78
FSMs	78
File Services	78
FILs	78
Index Services	78
INXbg	78
INXs	79
INXu	79
MKF Database Services	79
MKF_clean	79
Network Clearinghouse Services	79
NCHs	79
NCH_daemon	80
Print Services	80
PRI_check	80
PRI_daemon	80
PRI_notify	81
PRIs	81
PRI_worker	81

PSMs	81
Security Services	81
SEC_daemon	81
SECs	82
Storage Library Services	82
antcopy	82
del_commit	82
doccopy	82
docimport	83
dsched	83
dtp	83
dtp_tran	83
opendocs	83
OSCs	83
osi_migrate	84
sortod	84
WorkFlo Queue Services	84
WQs	84
Others	84
CDBs	84
CORs	84
COR_Listen	84
dbintf	85
SKFs	85
SQLs	85
TAPs	85
timeoutmonitor	85

## **4      Shared libraries    86**

**Shared library locations 87****Shared library descriptions 88**

## All services 88

COR 88

PPM 88

## Batch Entry Services 89

BES 89

BESI 89

## Cache Services 89

CSM 89

CSMI 89

## Document Services 89

ASH 89

BKG 90

CKS 90

CMT 90

CNF 90

CNT 90

DBL 90

DBP 91

DIA 91

DIG 91

DLI 91

DLII 91

DLIo 91

DLS 92

DLSI 92

DLSr 92

DOC 92

DOCa	92
DOCb	92
DOCI	92
DOCp	93
DSA	93
DT	93
FDT	93
FLT	93
HLT	93
IS	94
MKF	94
OSI	94
OSII	94
OSIr	94
OSS	94
OSSI	95
QMA	95
RFT	95
RMK	95
RMKr	95
SAS	95
SDC	96
SRF	96
Index Services	96
INX	96
INXa	96
INXD	96
INXI	97
INXr	97
Network ClearingHouse Services	97

NCH	97
NCHI	97
NCHr	97
SLMI	97
Print Services	98
PRI	98
PRII	98
Security Services	98
SEC	98
SECI	98
SECr	98
SMM	98
SSD	99
SSU	99
Storage Library Services	99
ARM	99
CLT	99
DIS	99
ODF	99
ODS	99
ODT	100
ODU	100
ODX	100
OFA	100
OPM	100
OSA	100
SLT	100
SNT	101
WRT	101
System Configuration	101

CDB	101
Miscellaneous System V	101
ORA	101
System Messages	101
ERM	101

## **5 Permanent database tables and contents 102**

### **Tables and contents 103**

annot_log	103
annotations	104
cluster_map	105
docs	106
family_disk	110
family_locator	112
lib_surfaces	113
od_stats	114
remote_family	115
scalar_numbers	116
surf_dyn_info	116
surf_info	119
surf_locator	119
surf_stat_info	121
surface_activity	123

## **6 Transient database tables and contents 125**

### **Batch tables and contents 126**

batch_ctl	126
-----------	-----

- batch\_doc 127
- batch\_dyn\_hdr 128
  - Document Phases 131
  - Phase Status 131
- batch\_hdr 132
- batch\_image 132
- batch\_ixdir 134
- batch\_ixval 135
- batch\_stat\_hdr 137
- batch\_data 139
- batch\_folder 140

**Cache Tables and Contents 141**

- csm\_caches 141
- csm\_free\_space 142
- csm\_temp\_id 143
- csm\_used\_space 144

**Print tables and contents 147**

- print\_docs 147
- print\_options 148
- print\_svcs 150
- print\_reqopt 151
- print\_requests 151

**Requests tables and contents 154**

- bkg\_request 154
- write\_request 154

## **7 Security database tables and contents 156**

**Tables and contents 157**

sec_object	157
sec_system	159
sec_deleted	161
sec_groups	162
sec_functions	162
sec_funcmbr	163
sec_namemap	163
sec_dbinfo	164
sec_rm_config	165
sec_map_prin_to_dn	165
sec_ce_dom_to_id	166
sec_map_cn_to_prin	166

## **Notices 167**

**Trademarks 171**

**U.S. Patents Disclosure 171**

## **Index 172**



# About this manual

This *System Reference Guide* is intended for use by support personnel responsible for IBM® FileNet® FileNet Image Services software.

## Overview

This guide discusses the system architecture, directory structure, MKF database structure and contents, and descriptions of shared libraries.

This guide assumes that you have knowledge of the following subjects:

- An understanding of the operating system (AIX®, HP-UX, Solaris, or Windows®) used on your FileNet Image Services servers.
- An understanding of, or prior experience with, the system administration tools available on your operating system (**smit** on AIX, **sam** on HP-UX, **admintool** and **swmtool** on Solaris 8, **Solaris Management Console** tools on Solaris 9 and Solaris 10, and **Control Panel** on Windows).
- An understanding of file system configuration methods and logical volume management concepts.

## Document revision history

Image Services version	Date	Comment
4.2	May 2011	Initial release.

## Scope of this guide

This guide contains a brief discussion of the following FileNet Image Services system items:

- Server configurations
- Directory structure (and content)
- Subsystem descriptions
- Configuration files (some supplied with the FileNet Image Services software others generated at the time of installation)
- Application programs
- Shared libraries
- MKF database tables and their contents (permanent, transient, and security)

The following table details where additional system support information is documented and the type of information you can expect to find in each document

To download IBM FileNet documentation from the IBM support page, see [\*\*“Accessing IBM FileNet Image Services documentation” on page 20.\*\*](#)

Document Name	Document Description
System Administrator's Handbook	A detailed discussion of critical System Administrative tasks, including database management, security management, system management, storage library control, and more.
System Administrator's Companion for UNIX System Administrator's Companion for Windows	A discussion on platform-specific operating system operation. Includes appendices detailing the types of tape drives and media supported. It also discusses backing up the RDBMS database files.
System Messages Handbook	A detailed discussion of system messages and an explanation of the message tools.
System Tools Reference Manual	A description of the system tools used to diagnose and manage a FileNet system. Some tools are used only by support personnel, whereas many others are used by FileNet Image Services System Administrators to analyze, diagnose, update, and repair FileNet Image Services components.
Enterprise Backup/Restore User's Guide	A description of Enterprise Backup/Restore (EBR), EBR concepts and functions, and how to use EBR to backup and restore a FileNet Image Services system.
Third-Party Backup/Restore Guidelines	A listing of general requirements and recommendations when using non-FileNet backup and restore programs or utilities.
Restore Guide for Windows	A discussion of methods used for restoring the permanent, transient, security, and RDBMS databases.

## Accessing IBM FileNet Image Services documentation

To access documentation for IBM FileNet Image Services products:

- 1 On the [www.ibm.com](http://www.ibm.com) website, enter “FileNet Image Services Documentation” in the search box on the menu bar.
- 2 Select **IBM - Product Documentation for FileNet Image Services** from the list of search results.

## Software education

IBM provides various forms of education. Please visit the IBM Training and Certification for IBM software page at:

[www.ibm.com/software/sw-training](http://www.ibm.com/software/sw-training).

## Comments and suggestions

If you have comments or suggestions regarding the product documentation, please send your comments by e-mail to:

[comments@us.ibm.com](mailto:comments@us.ibm.com).

Be sure to include the name of the product, the version number of the product, and the name and part number of the book (if applicable). If you are commenting on specific text, include the location of the text (for example, a chapter and section title, a table number, a page number, or a help topic title).

# System design and architecture

The FileNet Image Services software is designed to operate in a distributed environment. The individual services run in the following configurations:

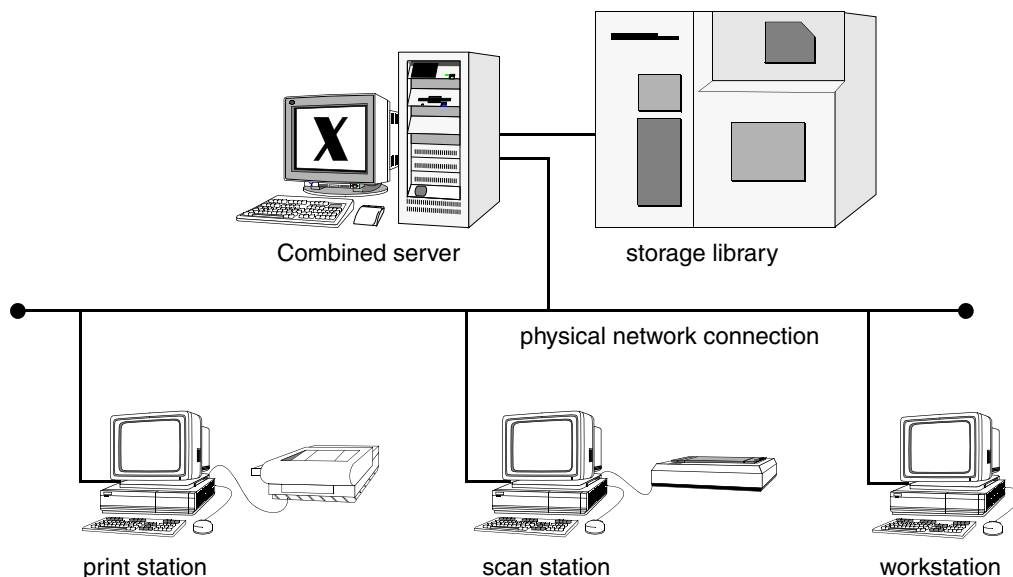
- Root/Index/Storage Library server: Combined server
- Root/Index and Storage Library servers: Dual server or Multiple Storage Library server system
- Application server
- Visual Workflo server
- Entry server (Remote Entry)

## Server configurations

### Combined server configuration

Within the Combined server configuration, only one FileNet Image Services server exists and it hosts all IS services. The Combined server configuration is typically used in development or small production envi-

ronments. The figure below shows the typical configuration for a Combined server system.

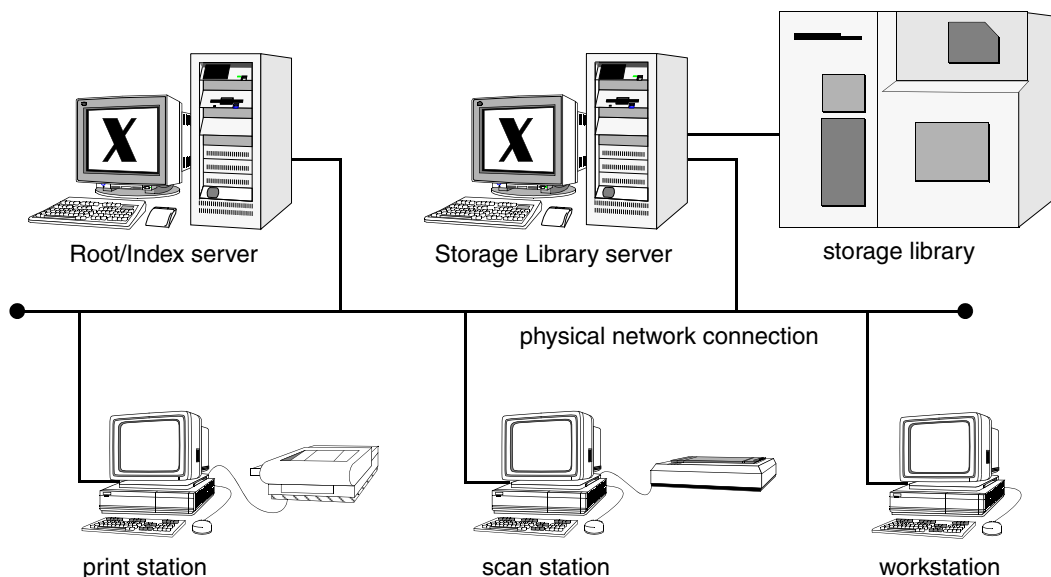


In this configuration the root/index, the Storage Library services, and all additional services reside on the same Combined server system.

## Dual or multiple storage library server configuration

In a Dual or Multiple Storage Library server configuration, at least two FileNet Image Services servers exist. The first server hosts the root/Index services and the other server(s) hosts only the storage library service. The RDBMS software is installed only on the Root/Index server. There can be more than one Storage Library server installed on

a Multiple Storage Library server system. The figure below shows the typical configuration for a Dual server system.

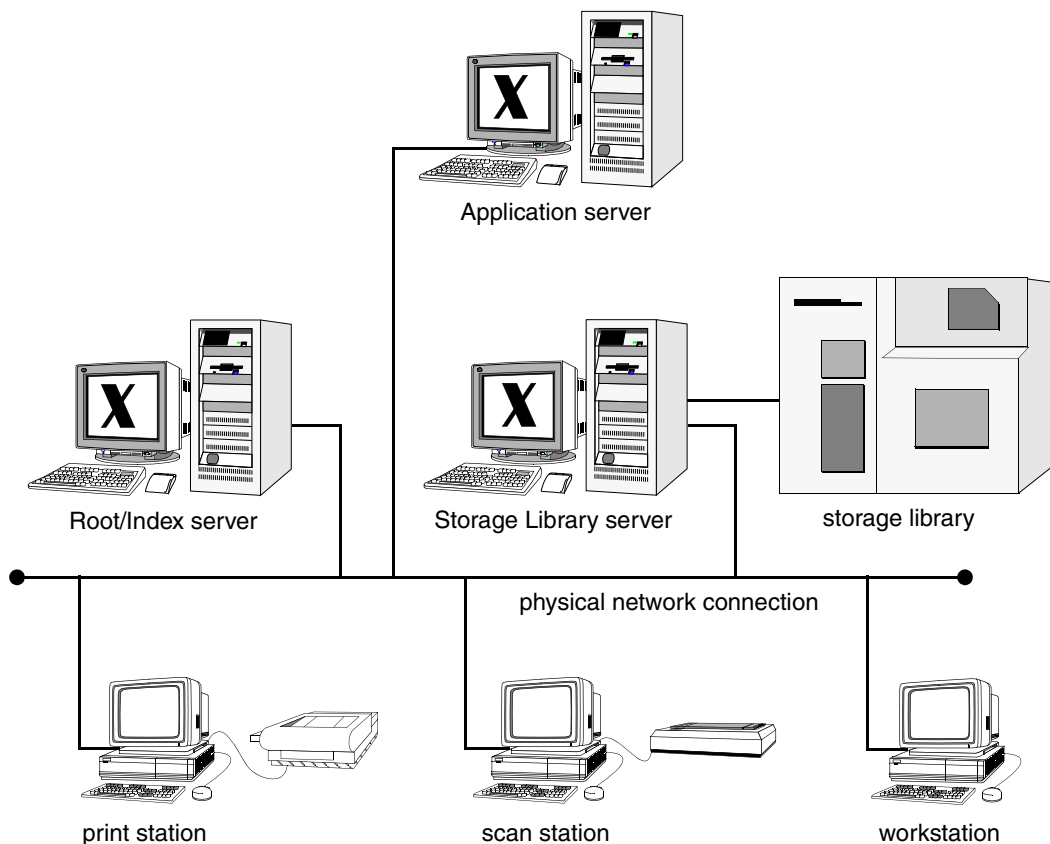


For information concerning the reasons why a Dual server configuration is desirable, consult your service representative.

## Application server configuration

An Application server configuration consists of two or more FileNet Image Services servers. An Application server is a separate, dedicated server that hosts specific services (such as WorkFlo Queue or VW services). The Application server must be attached to a Combined or Dual

server system. The figure below shows an Application server attached to a Dual server system.



If the Application server is hosting either Process services, SQL services or VW services, the RDBMS software is installed on both the Root/Index server and the Application server. Otherwise the RDBMS software is installed only on the Root/Index server.

You can add more than one Application server to the system if the requested resources outstrip the ability of the first Application server.



The following services can exist on more than one Application server in a system:

- Batch services
- Cache services
- Print services
- Structured Query Language services
- Process services
- VW Services

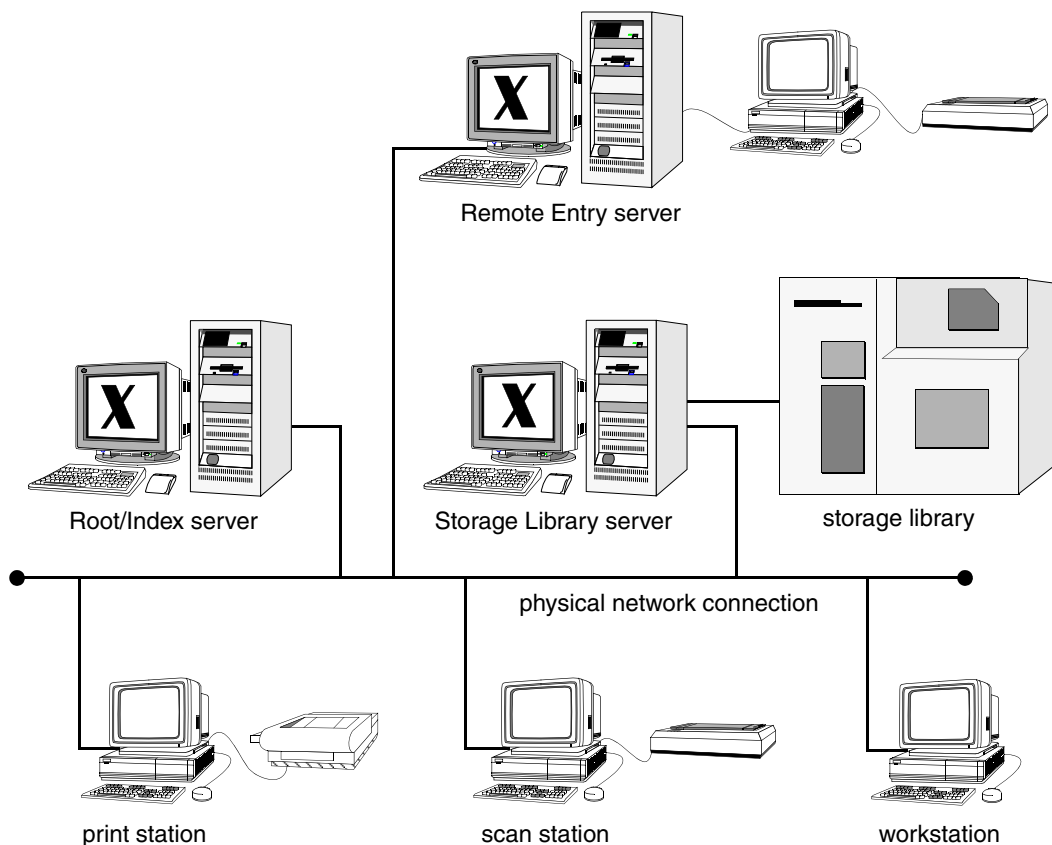
The Application server configuration provides maximum system performance since the Application server can be configured to assume responsibility for services that might otherwise slow the other servers in the system.

## Entry server configuration

An Entry server configuration (a Remote Entry server system) consists of two or more servers.

The Remote Entry server (RES) is configured as a Combined server without any storage library devices attached. Documents are scanned in at the Remote Entry server and sent over the network to a system that has storage capabilities.

The figure below shows a Remote Entry server attached to a Dual server system.



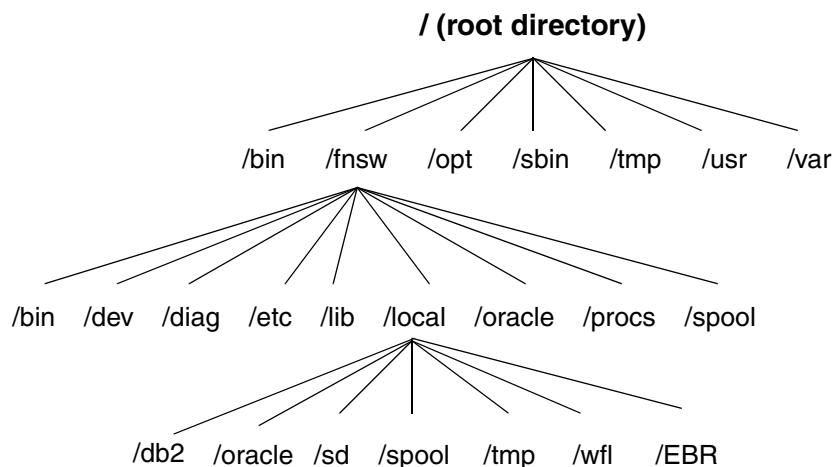
The Entry server's main function is to enter documents into the system. An entry system provides document entry on behalf of a target system.

It functions independently of the target system's status. (For example, the target system can be on-line or off-line.)

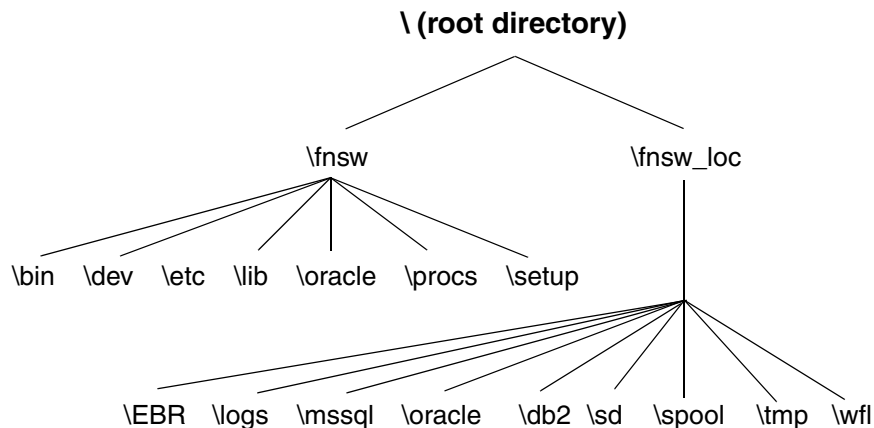
## Directory structure

The FileNet Image Services software resides in unique directories on both the UNIX® and Windows platforms.

The illustration below shows the typical directory structure on UNIX platforms. The directory structure is an approximation for all UNIX platforms. The actual structure may vary depending on platform differences or changes to the intended directory structure introduced during installation.



The illustration below shows the typical directory structure on the Windows platform.



## Directory contents

While the directory structure may be different depending on server usage or platform differences, all FileNet Image Services servers have the programs, shared libraries, tools, etc., residing in the same relative directory locations. The following sections provide a general overview of the contents of FileNet Image Services directories. The actual path designation varies according to the platform.

### /fnsw/bin

The /fnsw/bin directory contains FileNet-specific tools and application programs. For a complete discussion on the tools located in the /fnsw/bin directory, see the *System Tools Reference Manual*.

To download IBM FileNet documentation from the IBM support page, see [“Accessing IBM FileNet Image Services documentation” on page 20.](#)

The default path is:

**UNIX****/fnsw/bin****WIN****\fnsw\bin**

## **/fnsw/dev/1**

On the UNIX platform, the /fnsw/dev/1 directory contains links/device files to FileNet Image Services and FileNet-controlled RDBMS data-base files. This directory contains pointers to raw partitions containing data in MKF databases, the FileNet-controlled RDBMS database, and cache. The actual data is in different physical locations and their directory names depend on the server platform.

On the Windows platform, the data files physically reside in the \fnsw\dev\1 directory. This directory is the default location for MKF database, RDBMS, and cache files. The user determines on which disk drive to place MKF database and cache files. The user can use the FileNet utility, fn\_edit, to identify file locations.

The default path is:

**UNIX****/fnsw/dev/1****WIN****\fnsw\dev\1**

## **/fnsw/diag (UNIX only)**

The /fnsw/diag directory contains hardware diagnostics programs and utilities specifically related to storage library devices.

The /fnsw/diag directory exists on the UNIX servers only. The default path is:

 **UNIX****/fnsw/diag**

## **/fnsw/etc**

The /fnsw/etc directory contains administration files, initialization applications, and scripts. Most of the files located in this directory are used either during the FileNet Image Services software installation process or by your service representative during diagnostic evaluation of the system.

The default path is:

 **UNIX****/fnsw/etc** **WIN****\fnsw\etc**

## **/fnsw/lib**

The /fnsw/lib directory contains library and configuration files. This directory also contains EBR scripts and X11 uid files and help files for Motif.

The /fnsw/lib/conf\_db subdirectory also contains the server templates for configuration. The /fnsw/lib/dev/msg subdirectory contains the .msg and .cat files. The /shobj subdirectory contains FileNet shared libraries

(sometimes called shared objects, abstracts, or loadable libraries). Also in here are the /nls, /nlmaps, /perf, and /translate subdirectories.

The default path is:

**UNIX****/fnsw/lib****WIN****\fnsw\lib**

## **/fnsw/local (or \fnsw\_loc)**

The /fnsw/local (or on Windows: \fnsw\_loc) directory contains subdirectories critical to the operation of the FileNet Image Services software. On most platforms, this directory is a file system that is automatically mounted during system startup.

The default path is:

**UNIX****/fnsw/local****WIN****\fnsw\_loc**

## **/fnsw/local/sd**

The /fnsw/local/sd (on Windows \fnsw\_loc\sd) subdirectory contains the NCH (Network Clearinghouse) database. In addition, this subdirectory contains the server configuration files for the local server. On a Combined server system, the root\_station file resides in this subdirectory. This subdirectory also contains service information and configuration data.

**/fnsw/local/spool**

The /fnsw/local/spool subdirectory contains only the network log files for the TM\_daemon process.

**UNIX****/fnsw/local/spool****WIN****\fnsw\_loc\spool****/fnsw/local/tmp**

The /fnsw/local/tmp subdirectory contains configuration log files. This directory also contains an EBR subdirectory (/fnsw/local/tmp/EBR) which contains sample EBR and shell scripts.

**UNIX****/fnsw/local/tmp****WIN****\fnsw\_loc\tmp****/fnsw/local/wfl**

No information available. Information will be added as it becomes available.

**UNIX****/fnsw/local/wfl****WIN****\fnsw\_loc\wfl****/fnsw/local/<RDBMS>**

The existence of this subdirectory depends upon the platform type and the type of RDBMS software installed on the system.



**\fnsw\_loc\oracle**

**\fnsw\_loc\db2**

**\fnsw\_loc\mssql on Windows**

- On a system with the Oracle RDBMS software installed, the Oracle subdirectory exists. The /fnsw/local/oracle subdirectory contains subdirectories, Oracle parameter files, and SQL scripts for starting and stopping the Oracle software. In addition, this subdirectory contains subdirectories in which the Oracle RDBMS control files reside.
- On a system with the MS SQL Server RDBMS software installed, the \fnsw\local\mssql subdirectory exists. The \fnsw\local\mssql directory contains parameter files for the RDBMS software.
- On a system with the DB2® RDBMS software installed, the \fnsw\local\db2 subdirectory exists.

## **RDBMS (/fnsw/oracle, /fnsw/db2, or \fnsw\mssql)**

The actual path designation varies according to the RDBMS software installed on the system:

### **UNIX**

(Oracle and DB2): **/fnsw/oracle, /fnsw/db2**

### **WIN**

(Oracle, DB2 and Microsoft® SQL Server): **\fnsw\oracle, \fnsw\db2 or \fnsw\mssql**

The /fnsw/oracle directory contains SQL scripts used primarily for Oracle initialization. This directory contains scripts for installing patches.

On the Windows platform:

- if Microsoft SQL Server is installed on the system, the \fnsw\mssql directory exists.
- If Oracle is installed on the system, the \fnsw\oracle directory exists.
- If DB2 is installed on the system, the \fnsw\db2 directory exists.

## **/fnsw/procs**

The path, separated by platform, is shown below:

 **UNIX****/fnsw/procs** **WIN****\fnsw\procs**

The /fnsw/procs directory stores process files for all FileNet Image Services processes. If the directory does not exist the FileNet Image Services software will not start and will report serious errors.

## **/fnsw/spool (UNIX only)**

The /fnsw/spool directory contains a COR subdirectory (/fnsw/spool/COR) to which all Courier process log files are sent.

## **\fnsw\setup (Windows only)**

The FileNet Image Services software installation and setup software resides in this directory. The permission configuration files are also located here. Do not delete this directory or remove the contents after installing the FileNet Image Services software.

## Subsystem descriptions

Each FileNet Image Services subsystem is a specialized portion of the IS software than interacts with other IS subsystems to provide the various system services.

This section provides a summary of each of the subsystems that comprise the FileNet Image Services software.

### AT (Release Tools)

Tools used during the installation and updating of the FileNet Image Services software.

### AR (Archiving)

FileNet Archiving services moves document index and locator information from magnetic disk to optical disk. This frees up space on magnetic disk, allowing more documents to be added to the system without increasing the magnetic disk database size.

### B2 (Enterprise Backup and Restore)

FileNet Enterprise Backup and Restore (EBR) is a set of scripts and programs designed to automate the process for protecting data through backups and restores. Restores are performed infrequently, usually to recover from a disk failure or other disaster.

### BS (Batch Services)

The Batch services are responsible for creating, updating, and deleting batches; keeping track of image and index verification status; accumu-

lating batch totals; creating documents from images; maintaining the batch queues; and maintaining the transient database of batches in progress. A batch service has a three-part name with the batch service name first, followed by the system name followed by the organization: BatchService:pubs:ExampleCo.

## **CO (Courier Networking)**

The COR\_Listen process listens for incoming RPC requests. When an RPC (Remote Procedure Call) arrives, COR\_Listen immediately finds a suitable service process (stub) and hands over the RPC, then resumes listening for the next RPC. Only one COR\_Listen process is running at any given time

## **CS (Cache Services)**

The Cache services are responsible for moving documents between different caches; the services are also responsible for overall cache management. Cache services provides clients a magnetic disk based storage and retrieval mechanism for objects. Routines are provided to log on to or log off from a cache service, to create, open, close, read, write, delete, move, rename, and copy objects, to get attributes of a cache, and to list objects in a cache.

## **DM (Database Maintenance)**

The database maintenance subsystem is a collection of programs that give the user the ability to set up and maintain database information needed for the storage and retrieval of documents.

## DS (Document Services)

The Document services are responsible for all optical disk activity related to prefetching, assigning document IDs, locating documents (by DOC ID), migrating documents (to and from disk), copying, and importing documents.

## EV (SystemV Applications Executive)

The System V Applications Executive is a set of applications that manages the state of the FileNet Image Services software. The functionality of each application is as follows:

- **Xtaskman** is the graphical user interface application used to control the state of the FileNet Image Services software, list the current FileNet processes, view the error log, and view RPC activity.
- **initfnsw** is the console application used to control the state of the FileNet Image Services software.
- **whatsup** is the console application used to list the current FileNet processes.
- **vl** is the console application used to view the event log.
- **Xapex** is the graphical user interface application used to log into FileNet Image Services. The menu options assist users to initiate other server applications after login.

## FL (FileNet LAN Analysis Tool)

The FileNet LAN Analysis Tool (FLAT) is a support tool that simplifies local area network (LAN) packet trace analysis. The tool can read and decode the output of network analyzer programs and network packet

traces on UNIX operating systems. It provides some FileNet-specific analysis capabilities not found in other commercially-available analyzers and analyzes traces captured from a variety of other tools, such as: LANalyzer and Sniffer, LANwatch, iptrace (AIX only), and snoop (Solaris only).

## **GB (Generic Database)**

The Generic Database subsystem is a set of libraries (GDB, GDBO, GDBM, ORA) that handle the interface between the FileNet software and the supported Relational Databases, Oracle, DB2 and Microsoft SQL Server. All subsystems that want information from the RDBMS will call the GDB libraries with their database queries and updates.

## **HE (Help Error)**

The Help Error subsystem controls the areas of Help text and error messages.

## **IS (Index Services)**

Index Services handles queries and updates to the index database. It stores the document index records, document class and folder information, and the data dictionary.

## **LM (Software License Management)**

FileNet Software License Management (LM) is a set of programs and libraries designed to manage the use of various components of the FileNet Image Services system that are separately licensed. It introduces the concept of a Licensed User and Licensed Configurations.

## **MK (Multi-Keyed Files)**

The multi-keyed files subsystem manages the transient database (Trans\_DB*n*), the permanent database (Perm\_DB*n*), the network clearinghouse database (NCH\_db0), and the security database (Sec\_DB*n*).

## **MV (SystemV Miscellaneous)**

The MV subsystem controls programming utilities, shared memory and killfns<sub>w</sub>.

## **NC (Network Clearing House)**

The Network Clearinghouse (NCH) allows the distributed resources to be located. System elements can locate the services by appropriate calls to the Clearinghouse service. The name of a resource consists of three parts that identify a resource object, domain (system name), and the organization (your company name): object:domain:organization

## **NL (National Language Services)**

The NL subsystem controls the localization tools.

## **NV (System V Networking)**

System V Networking deals with network protocols, Courier, and process management.

## **OS (OSAR Services)**

The OSAR services are responsible for monitoring and controlling all traffic (input/output) between the server and the storage library devices. These services are supported by Document services.

## **PM (Performance Measurement Tools)**

The Performance Measurement Tools (PM) subsystem contains programs designed to measure some aspect of performance on a system. For example, one such program, CPT\_test, measures the Courier network throughput and latency using the CPT's (Courier Performance Testing Tools) server.

## **PR (Print Services)**

Print Services manages print requests for documents or text streams and displays information on printer attributes, printer status, and job queue status.

## **PR (Print Services II)**

Print Services II is a newer version of Print services and performs the same functions as Print services.

## **PV (Server Print)**

The Server Print services are responsible for print documents in cache or in memory. Controls print requests and monitors printing status.



## SE (Security Services)

FileNet Security services (SE) are a set of programs and libraries that provide user authentication and access control for the FileNet Image Services system. Access control is based on hierarchical group membership and distinguishes Read - Write - Execute/Append privileges.

## SF (System Configuration Tools)

The SF subsystem controls the tools used to initialize and update the FileNet Image Services configuration databases, configuration files, and devices.

## SU (System Utilities)

System Utilities are miscellaneous utility programs like **ident**, **stamp**, **less**, and **uncompressdir**.

## WQ (WorkFlo Queue Services)

The WorkFlo Queue services are responsible for monitoring and controlling WorkFlo queues.

## XV (XVT Portable Toolkit)

XVT Portable Toolkit is a vendor-acquired toolkit on which the graphical user interface server applications are built.

## Configuration Files

Each FileNet system must be configured properly before it is used in a production environment. The configuration process includes configuring hardware and software parameters and device files. To manage all configuration settings and maintain consistent performance, the Image Services software uses configuration files to control how each component or device is set up.

A typical FileNet Image Services system contains numerous configuration files. Some files contain parameters for several Image Services components, procedures, and databases. Other files control settings for only a single component.

Configuration files contain numerous parameters determining the configuration for each component installed on the system. Default parameters are established during the Image Services software installation; however, you can change the parameters when necessary.

## Configuration Database

The IMS\_xxx.CDB file is the main configuration database file for an Image Services system. It is created and modified by the fn\_edit program. For a complete description of the fn\_edit utility, see the *System Configuration Overview*. To download IBM FileNet documentation from the IBM support page, see [\*\*“Accessing IBM FileNet Image Services documentation” on page 20.\*\*](#)

The IMS\_xxx.CDB file is located in the following directory (where xxx is a sequentially increasing number assigned automatically by the system):

**UNIX**

`/fnsf/local/sd/conf_db/IMS_xxx.cdb`

**WIN**

`\fnsf_loc\sd\conf_db\IMS_xxx.cdb`

You can use the System Configuration Editor to view current configuration information.

- 3 At the system prompt, enter the following command:

**fn\_edit &**

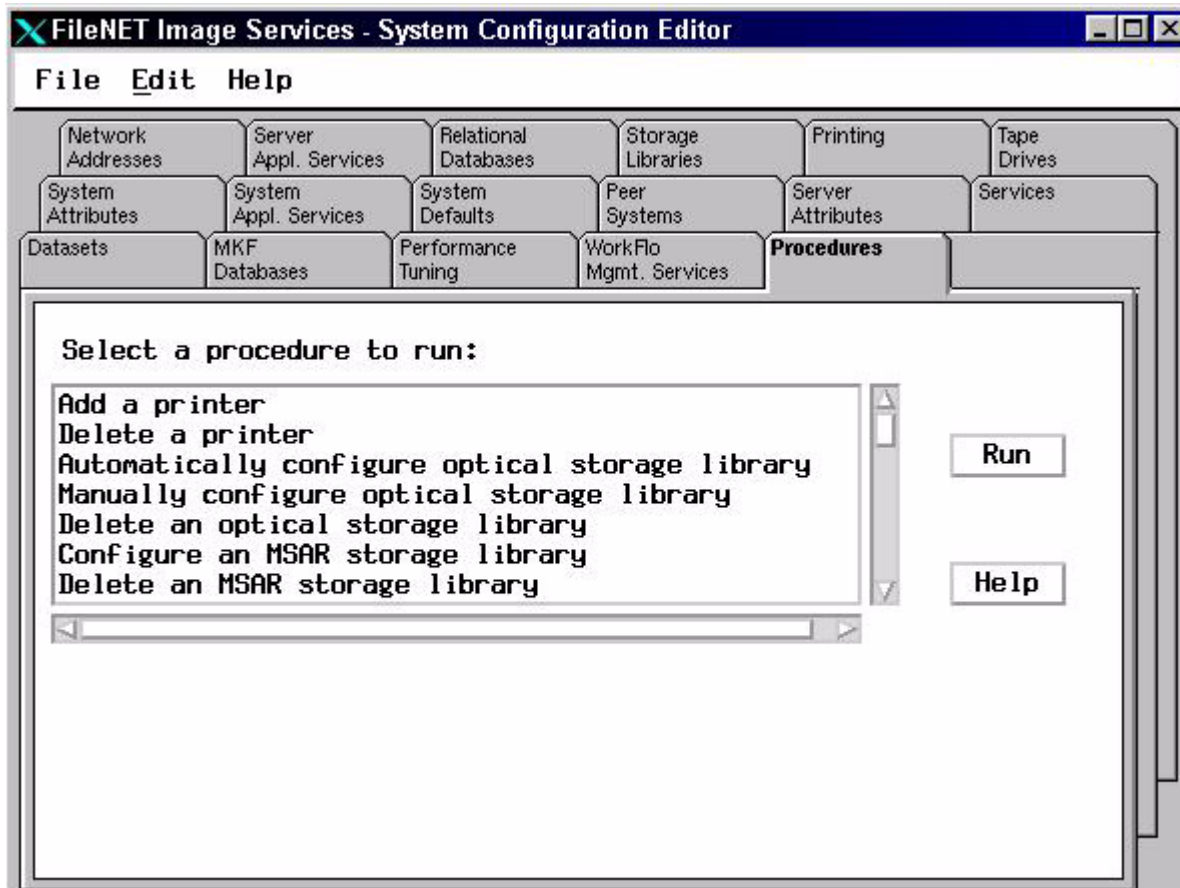
---

**Note** On UNIX platforms you must be running an X Windows/Motif interface to use the System Configuration Editor.

---

**Windows Only:** double-click the Configuration Editor icon.

- 4 Click on any of the available tabs to view the configuration settings.



**Note** Starting with release 3.3.0, the **IMS\_xxx.CDB** database file took the place of the **system** configuration file and the **initparms** file, which was produced by the **es** tool.

When the IS software is initially installed, the `fn_edit` program accesses read-only templates to create the first configuration database file: `IMS_1.CDB`.

However, as you make and save changes to the system configuration (through the `fn_edit` interface), the version number of the `.CDB` file increments accordingly. (The earlier `.CDB` files are left unaffected in the original directory.)

The `IMS_xxx.CDB` file contains information about all aspects of IS software operation including, but not limited to, the following items:

- Cache sizes and locations
- Service configurations and locations
- Dataset sizes and locations
- RDBMS type and version information
- Tablespace names and locations
- Network configuration information
- Storage library configuration information
- Peer system information
- Performance tuning parameters

## Supplied Configuration Files

This section describes configuration files that are placed on the server at the time of the IS software installation.

---

**Note** You can view any of the files discussed in this chapter using a native text editor (Windows - Wordpad, Solaris - textedit, all UNIX platforms - vi, etc.)

---

### erm.msg

The erm file is the main message file on the system containing most of the IS messages. The erm file is located in the following directory (separated according to platform):

- UNIX: `/fnsw/lib/nls/msg/erm.msg`

A partial listing of the contents of the erm.msg file is shown below.

```
00001 <15,255,32>SPP_ErrSuspCreated: Connection suspended
00002 <15,255,4096>ERP_ErrUnspec: Unspecified error was detected at
destination
00003 <15,255,4352>ERP_ErrUnspec2: Unspecified error before reaching
destination
00004 <15,255,4608>ERP_ErrBadConn: Packet send to suspended transport
connection
00005 <15,255,4864>ERP_ErrNoEpds: Unable to allocate epd to send error
pkt
00006 <15,255,8192>COR: Unknown remote service
00007 <15,255,12288>An attempt was made to access a file for a network
service that is either not installed or not configured on this host.
(CONFIGURATION ERROR)
...
```

## **UNIX Files**

The .cat (catalog) and .msg (message) files are provided as part of the standard IS software. Both files contain the same information in different formats. The .msg files are in ASCII and are input files to the .cat files in binary, which are the output files. The UNIX tool, gencat, uses the .msg files to build the .cat files that contain language-specific information for the messages. The .cat files are data files that are accessed by the application programs and cannot be read with any of the conventional viewing commands.

The /fns/lib/msg.category file is referenced by the msg command to map a message tuple number with the name of a category, usually a shared library. The file is provided as part of the standard IS software. It cannot be read with conventional viewing commands.

## **Windows Files**

The .dll files are provided as part of the standard release software and are the equivalent to the .cat (catalog) files on the UNIX platforms.

The .dll files are data files that are accessed by the application programs and cannot be read with any of the conventional viewing commands. The equivalent to the UNIX .msg files are not included in the standard IS software.

## **Language Directory**

For UNIX, this /fns/lib/nls/msg/<lang> directory contains the same three .cat files described above but with messages in a language other than English. To use this directory, you set a language environment variable. Otherwise, Image Services uses the English files in the default path. For example, the de\_DE, for Deutsche, directory contains

the .cat files in German. The .msg files are created manually in German, and then the gencat tool is run to build the .cat files.

See the *Image Services System Messages Handbook* for more information. To download IBM FileNet documentation from the IBM support page, see [\*\*“Accessing IBM FileNet Image Services documentation” on page 20.\*\*](#)

## fninfo.msg

The different platforms use unique message resources to provide information on the IS status. The system information file is located in the following directory:

- UNIX: `/fnsw/lib/nls/msg/fninfo.msg`

A partial listing of the fninfo.msg (for the UNIX platforms only) is shown below.

```
...
05953 <170,12,1>For %s tape %d, please mount tape with serial number %.6s
05954 <170,12,2>For %s tape %d, please mount a labelled tape.
05955 <170,12,3>For %s tape %d, please mount an unlabelled tape.
05956 <170,12,4>Enter 1 (or <CR>) when tape ready, 2 if no tape available
05957 <170,12,5>For %s tape %d (labelled), overwrite unlabelled tape?
(1=yes,2=no)
05958 <170,12,6>For %s tape %d (unlabelled), overwrite labelled tape?
(1=yes,2=no)
...
```

---

**Note** **Windows:** The ERM\_INFO file is used exclusively by the FileNet backup and restore programs.

---



The fninfo.msg also contains SNMP trap messages. SNMP traps are alerts that are pro-actively generated by the agent software and sent to the third party network management software. (See [“snmp.conf” on page 55](#) for more snmp information.)

## tmmsg.msg

The tmmsg file is the task manager message file used by Xtaskman and initfnsw. The tmmsg file contains all of the dynamic task management messages that appear in the *Current Status* pop-up window when you recycle the FileNet software.

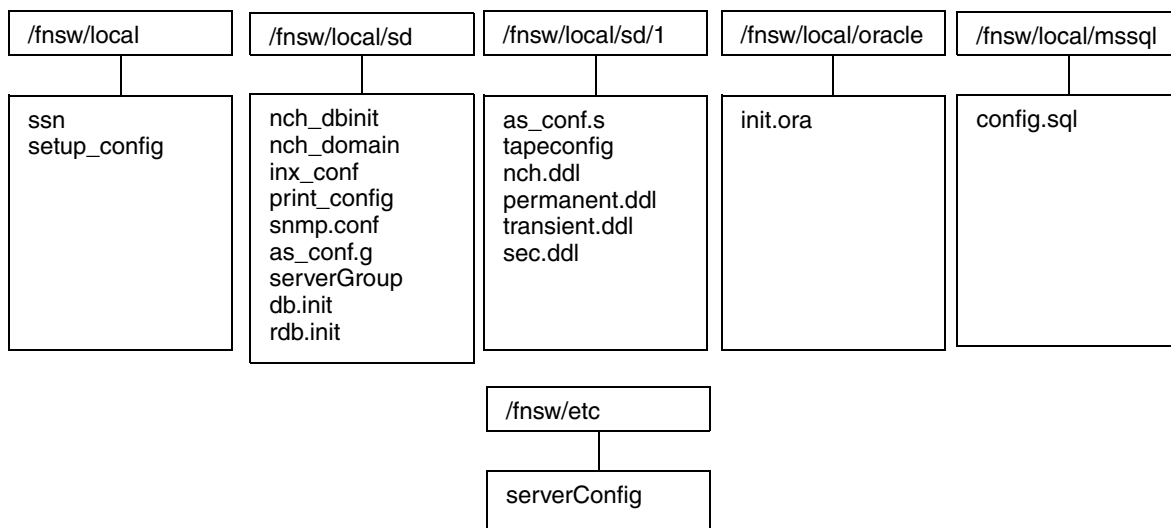
The tmmsg file is located in the following directory:

- UNIX: **/fnsw/lib/nls/msg/tmmsg.msg**

## Generated Configuration Files

All of the files discussed in this section are generated by the `fn_build` program after Image Services software installation. The information contained in the generated files comes from a combination of system information and information entered in the system configuration database file.

The `fn_build` program automatically creates the new files in the directories shown below.



### as\_conf.g

The `as_conf.g` file contains all the system-specific Document services configuration information that applies to the entire domain. It is located in the following directory:

- UNIX: `/fnsw/local/sd/as_conf.g`
- Windows: `\fnsw_loc\sd\as_conf.g`

**Note** Do not manually edit the contents of the `as_conf.g` file. `Fn_build` runs every time you start/restart IS. You can also run `fn_build` after modifying the file. In both cases, modifications will be overwritten. **Manually editing the `as_conf.g` file is NOT recommended.**

---

The content of a typical `as_conf.g` file is shown below.

```
processes {
  notify      ds_notify 2
  scheduler    dsched
  dtp          dtp
  dtp_tran     dtp_tran 1
  rmt_commit   rmt_commit 1
  fbc_commit   fbc_commit
  del_commit   del_commit
  osi_migrate   osi_migrate
}

compatible_systems {
}

delete_on_write f
fbc_breakup f
fbc_breakup_count 400
conversion_system f
pages_per_retrieval 8
page_cache_doc_duration 300 300 120
docnum_range 100000 3999999999
surfaceid_range 3000 4294967290
doc_batch_bufsize 80
ims_name DefaultIMS:earth:FileNet
```

## as\_conf.s

The as\_conf.s file is located in the following directory:

- UNIX: **/fnsw/local/sd/1/as\_conf.s**
- Windows: **\fnsw\_loc\sd\1\as\_conf.s**

The as\_conf.s file handles all the server-specific Document services configuration information. The content of a typical as\_conf.s file is shown below.

```
cache_partitions {
    /fnsw/dev/1/cache0 0 102400
}
special_caches {
    FN_print_cache print_cache1:earth:FileNet
}
scheduling_options {
    minimum_transfers 50
    critical_read_time 60
    write_delays 50 60 600
    io_active_time 1800
    spindown_delay 6
    prefetch_delays 0 0
}
data_base_names {
    permanent /fnsw/dev/1/permanent_db0 DocServer
    transient /fnsw/dev/1/transient_db0 DocServer
}
document_bufcnt 16
document_bufsize 64
csm_io_process_cnt 2
read_lookahead_cnt 10
dir_bufsize 16
dir_bufcnt 4
cache_threshold 85 80 90
page_cache_doc_duration 300 300 120...
```

## init.ora

The init.ora is a text file containing a list of instance configuration parameters. This file contains Oracle database parameters and is built by fn\_build. In FileNet-controlled Oracle installations, the Oracle software reads the init.ora file each time the RDBM software starts. The init.ora file is located in the following directory:

- UNIX: **/fnsw/local/oracle/init.ora**
- Windows: **\\fnsw\_loc\\oracle\\init.ora**

The Oracle software must be able to read the init.ora file to start an instance. If the init.ora file is not present on the server or is corrupted, no Oracle instance will start. The content of a typical init.ora file is shown below.

```
control_files = /fnsw/local/oracle/control0/ctl.ora,/fnsw/local/oracle/control1/
ctl.ora
db_name = indexdb
nls_territory = AMERICA
db_block_buffers = 1000
db_block_size = 2048
db_files = 200
db_file_multiblock_read_count = 8
dml_locks = 100
log_archive_start = FALSE
log_checkpoint_interval = 50000
log_buffer = 163840
max_dump_file_size = 10240
processes = 64
shared_pool_size = 64M
compatible = 9.2.0.1.0
sort_area_size = 131072
rollback_segments = rs0,rs1,rs2,rs3
# WARNING: lines above this line will be replaced by FileNet at startup.
```

You can edit parameter values in a parameter file with a basic text editor; however, editing methods are operating system-specific. Normally, this file is rebuilt each time the software is started (by `fn_build`). Changes should be made using `fn_edit`, except for entries below the **#WARNING** line. Oracle treats string literals defined for National Language Support (NLS) parameters in the file as if they are in the database character set.

---

**Note** Additions to the `init.ora` file below the `# WARNING ...` line will **NOT** be destroyed by `fn_build`.

---

## inx\_conf

The `inx_conf` file identifies if documents will be catalogued to the FileNet-defined Oracle database, and if user indexes will be used in distributing queues. The `inx_conf` file is located in the following directory:

- UNIX: `/fnsw/local/sd/inx_conf`
- Windows: `\fnsw_loc\sd\inx_conf`

The content of a sample `inx_conf` file is shown below.

```
no_catalog = 0
ixwfq_ctl_file = 1
```

## snmp.conf

The snmp.conf file is located in the following directory:

- UNIX: **/fnsw/local/sd/snmp.conf**
- Windows: **\fnsw\_loc\sd\snmp.conf**

The content of a sample snmp.conf file is shown below.

```
OkTrapFlag = 1
WarningTrapFlag = 1
OperatorTrapFlag = 1
SevereTrapFlag = 1
```

## MKF.ddl (s)

The Data Definition Language (ddl) files contain the parameters used when an MKF database is created. The fn\_build program runs a different script to create each MKF database. The data definition files are located in the following directory:

- UNIX: **/fnsw/local/sd/1/\*.ddl**
- Windows: **\fnsw\_loc\sd\1\\*.ddl**

Once an MKF database is created, your service representative may need to update it due to a software upgrade or changes to performance tuning parameters.

The table below outlines the scripts used with the ddl files. The fn\_build program uses a different script for the Permanent and Transient data-

bases following some upgrades. The table below associates each MKF database with a ddl file and the script that creates it or updates the file.

*.ddl (MKF) File	Database	Created by	Updated by
nch.ddl	NCH	fn_util initnch	fn_util initnch
permanent.ddl	Permanent	fn_util initperm	fn_util updateperm
sec.ddl	Security	fn_util initsec	fn_util updatesec
transient.ddl	Transient	fn_util inittrans	fn_util updatetrans

The contents of the ddl files vary slightly from one file to another. However, they all contain default parameters for each MKF database including the size of the database and recovery log partitions. The content of a typical permanent.ddl file is shown below:

```
--permanent Document Server Database DDL
PARAMETERS
(
  number_of_buffers = 256,
  max_concurrent_transactions = 3,
  max_concurrent_long_transactions = 1,
  read_after_write = set,
  max_record_types = 32,
  max_items_per_record = 32,
  overwrite_rl_action = warning_message,
  rl_update_frequency = 230
);
FILES
(
  target_station "DocServer";
  base data partition "/fnsf/dev/1/permanent_db0" (blocks =
102400);
  recovery_log partition "/fnsf/dev/1/permanent_rl0" (start
= 0, blocks = 40960);
);
#include "/fnsf/lib/perm_db.desc"
```



## nch\_dbinit

This configuration file is the Network Clearinghouse database initialization file. The fn\_util initnch program builds the NCH database using parameters stored in this file. The fn\_build program is responsible for transferring modified information from IMS\_xxx.CDB to nch\_dbinit.

The nch\_dbinit file is located in the following directory:

- UNIX: /fnsw/local/sd/nch\_dbinit
- Windows: \fnsw\_loc\sd\nch\_dbinit

The content of a typical nch\_dbinit file is shown below.

```
# nch_header
init hint:defaults:FileNet
#
primarydomain glacrpt:FileNet
set domain glacrpt
set organization FileNet
removeremotes
...
# nch_ims
createobj DefaultIMS
additem DefaultIMS ims "System IMS Defaults"
additem DefaultIMS imsDesc 256 IndexServer DocServer 10337
FormsServer

# nch_systemDefault

createobj fn_system
additem fn_system system "System Defaults"
additem fn_system defServiceDesc 2 Bes1 \ DefaultIMS
PrintServer SKFserver WfLS
erver \ SQLServer
additem fn_system defService1Desc 1 \ RFSServer
SecurityService \
additem fn_system defDeviceDesc 0 \ \
additem fn_system defCacheDesc 2 \ batch_cache1 revise_cache1
\ \ \
additem fn_system attributesDesc 1 1 11 11
...
```

## nch\_domain

This file is created by the `fn_build` utility. It contains the domain:organization names of the system. The `fn_build` utility reads the system file and creates `nch_domain`. The `fn_build` utility then calls the program `nch_tool` to update the `NCH_db0` with the contents of the `nch_domain` file. The `nch_domain` file is located in the following directory (separated according to platform):

- UNIX: `/fnsw/local/sd/nch_domain`
- Windows: `\fnsw_loc\sd\nch_domain`

The content of a sample `nch_domain` file is shown below.

```
earth:YourCorp
```

## print\_config

The `print_config` file contains both global and specific Print services parameters. This configuration file contains a description of each printer for all Print services on the system.

---

**Note** If a `print_config` file exists, it is renamed `print_config.old` when the `fn_build` program rewrites it each time `fn_build` is run.

---

The `print_config` file is located in the following directory (separated according to platform):

- UNIX: `/fnsw/local/sd/print_config`
- Windows: `\fnsw_loc\sd\print_config`

The partial content of a sample print\_config file is shown below.

```
print_servicePrintServer:earth:FileNet {  
print_cachesys_print_cache1:earth:FileNet  
cache_fullness_threshold 80  
print_page_bytes50000  
...
```

## rdb.init

The rdb.init file contains a start\_stop parameter at a value of 1 or 0. This file is located in the following directory (separated according to platform):

- UNIX: **/fnsw/local/sd/rdb.init**
- Windows: **\fnsw\_loc\sd\rdb.init**

The content of a typical rdb.init file is shown below.

```
start_stop 1
```

## serverGroup

The serverGroup file is used by the PPM shared library to enforce the number of licensed users accessing the Image Services software created by fn\_build.

The serverGroup file is located in the following directory (separated according to platform):

- UNIX: **/fnsw/local/sd/serverGroup**

- Windows: `\fnsw_loc\sd\serverGroup`

The content of a typical `serverGroup` file is shown below.

```
#CULS, RDB_TYPE, INX, SQI, WQS  
63 1 134231042 1 134231063 1 134231066
```

When the maximum number of users is reached, RPCs for any program in that group will be rejected, and a reject count displayed incrementally. You can track the reject count using PPMOI. The `serverGroup` file contains at least one line, and each line is structured similar to the example shown below:

**max RDB\_Type program1 version1 program2 version2 ...**

where **max** is the CUL limit, **RDB\_Type** is an integer, **program** is the Courier remote program number, and **version** is the Courier program version number.

The example shown specifies a maximum of 50 users for INX, SQI and WQS.

**50 1 0x8003402 1 0x8003417 1 0x800341a 1**

All numbers are separated by spaces and specified according to C conventions, i.e., 0x# for hex, 0# for octal and # for decimal representations. Comment lines start with either a # or / character.

## serverConfig

Each time the Image Services software starts, the serverConfig file is loaded into active memory on the server. It is actually part of the Process Manager Software and is static. The file contains the maximum number of request handlers that can be loaded into memory for each service or procedure. The PPM shared library checks the in-memory serverConfig file to determine the maximum number of request handlers that can be started for each request type.

The serverConfig file is located in one of these directories (separated according to platform):

- UNIX: **/fnsw/etc/serverConfig**
- Windows: **\\fnsw\\etc\\serverConfig**

The table lists the maximum number of request handlers stored in the serverConfig file illustrated later in this section.

System Component	Request Handler	Maximum
Batch services	BESs	8
Cache services	CSMs	12
Index services	INXs	12
Document services	DOCs	12
	RMKs	1
	DLIs	5
NCH services	NCHs	12
OSAR services	OSIs	3
	OSSs	3
	OSCs	3
Print services	PRIs	12

System Component	Request Handler	Maximum
Backup	BRBs	1
Security services	SECs	12
Restore	BRRs	1

If you have been trained to support your FileNet system, use the PPMOI tool to change the maximum number of request handlers for the in-memory version of the serverConfig file. (See the *System Tools Reference Manual* for more information on using the PPMOI tool.)

The serverConfig file has a unique structure that allows you to easily interpret the current request handler status for each service type. Each request handler entry appears on its own line and consists of seven parameter fields separated by spaces. The example request handler entry below illustrates the typical structure of the serverConfig file entries.

An optional eighth column is available for the purpose of tuning Courier timeout durations. See page 64 for details.

```

BESS      0134231044  1  8  1  0  64
|          |         |  |  |  |  |
1          2         3  4  5  6  7

```

- The first (1) parameter field is the name of the request handler process.
- The second (2) parameter field is the remote program number.
- The third (3) parameter field is the remote program version number.
- The fourth (4) parameter field is the maximum number of request handler processes of this type.

- The fifth (5) parameter field is a flag. If the flag is set to 1, the process will be monitored by the FileNet SNMP proxy agent.
- The sixth (6) parameter field is an inactivity interval. If the value is other than zero, the number is the number of seconds after which the request handler process will free certain resources (for example, log off of Oracle).
- The seventh (7) parameter field is the queue connection size for request handler processes.
- The eighth (8) parameter is optional. See **Tuning Courier timeout durations on page 64** for details.

The content of a typical serverConfig file is shown below.

```
...
NCHs      02          2 12 1 0 64
CSMs      0134231040 1 12 1 0 64
DOCs      0134231041 1 12 1 0 64
INXs      0134231042 1 12 1 0 64
PRIs      0134231043 1 12 1 0 64
BESs      0134231044 1 8   1 0 64
PSMs      0134231050 2 4   0 0 64 # Do not increase number of PSMs
RMKs      0134231051 1 1   0 0 64 # Do not increase number of request handlers
OSIs      0134231052 1 3   0 0 64
DLIs      0134231053 1 5   0 0 64
OSSs      0134231054 1 3   1 0 64
DLSs      0134231055 1 8   0 0 64
BRBs      0134231056 1 1   0 0 64
BRRs      0134231057 1 1   0 0 64
TAPs      0134231058 1 1   0 0 64
dbintf    0134231059 1 12  0 0 64
SECs      0134231060 1 12  1 0 64
SKFs      0134231061 1 12  0 0 64
SQIs      0134231063 1 10  1 0 64
FILs      0134231064 1 6   1 0 64
WQSs      0134231066 1 12  1 0 64
ARIs      0134231068 1 1   0 0 64
CPTs      0134231069 1 32  0 0 64
RJEs      0134231070 1 3   0 0 64
OSCs      0134231072 1 3   0 0 64
FSMs      0134231073 1 6   0 0 64
CDBs      0134231074 1 6   0 0 64
BRDs      0134231075 1 12  0 0 64
BRTs      0134231076 1 12  0 0 64
BRRMKs    0134231077 1 12  0 0 64
VWSs      0134231078 1 64  0 0 64
...
```

## Tuning Courier timeout durations

As of Image Services 4.2.0, an optional parameter, Tuning courier timeout durations, is available.

The values for this parameter are:



- Add optional `recv()` timeout value for worker threads.
- Add new utility `"COR_set_recv_timeout"`
- Add optional timeout value for prefetch threads.
- Add new utility `"COR_set_prefetch_timeout"`
- Add optional maximum prefetch thread count to be between 75 and 200.

### Add optional `recv()` timeout value for worker threads

An optional new column appended to `/fnsw/etc/serverConfig`. This new column can contain nothing, a zero, or a positive timeout value in seconds which is used as the timeout value for the particular request handler type.

For example:

```
...  
NCHs      02          2 24 1 0 30  
CSMs      0134231040 1 24 1 0  
DOCs      0134231041 1 24 1 0 100  
INXs      0134231042 1 24 1 0  
PRIs      0134231043 1 24 1 0 0  
BESs      0134231044 1 16 1 0 555  
...
```

In this example:

NCHs has a timeout value of 30

CSMs has a timeout value of 0 (no entry in column 7)

DOCs has a timeout value of 100

INXs has a timeout value of 0 (no entry in column 7)

PRIs has a timeout value of 0 (0 = no timeout)  
BESs has a timeout value of 555

The timeout value for each incoming RPC is set during prefetch period. Just before the RPC goes into the service queue, the prefetch thread sets the timeout value in a new structure field in COR handle. The COR abstract loads the serverConfig information once at startup into shared memory and the information is available for COR\_Listen(s) through a new COR entry point.

A value of 0 in serverConfig means no timeout.  
If timeout value is not specified, it means no timeout.  
If a timeout is reached, a syslog entry is generated:

```
2010/10/15 12:35:35.023 155,20,4 <fns> COR_Listen -pt -s32769 -  
t3600 -d100 (991476.515.39 0xf20f4.203) ...
```

Courier error: recv() timed out (15 secs) from ip address 9.65.17.89  
[1850]. request handler type = 134231040

### **Add new utility "COR\_set\_recv\_timeout"**

This new tool takes zero, one or two parameters. If run with no parameter, it displays the current value of the timeout for all request handlers. If run with one parameter, it displays the timeout value for a request handler type. If run with two parameters, it dynamically sets the new timeout value in seconds for a request handler type.

For example:

COR_set_recv_timeout	display timeout values for all stubs
COR_set_recv_timeout CSMs	display timeout value for CSM

only

COR\_set\_rcv\_timeout DOCs 100          set timeout value for DOCs  
to 100

The range of the timeout value must be in between 0 and MAXINT. if the value specified is out of the range, then it's a no-op and the application exits with a value of 1. Having a discipline on the exit return value of the utility makes it scriptable.

If run with -h or illegal parameters, it displays this usage message.

Usage: COR\_set\_rcv\_timeout [stub type] [timeout in seconds]

### **Add optional timeout value for prefetch threads**

In all previous Image Services releases, COR\_Listen launched with a -d3600 options. This is the timeout value used for prefetch threads. 3600 seconds is one hour. In Image Services 4.2.0, this number can be reconfigured to be in the range between 90 and 3600. The optional prefetch timeout value (in seconds) and max instance number can be specified in the configuration file: /fnsw/local/trigger/COR\_prefetch.txt

MAX value must be between 75 and 200

TIMEOUT value must be between 90 and 3600

For example:

```
MAX      100
TIMEOUT  300
```

Under rare occasions where COR\_Listen is launched without the -d option and not optionally configured, the timeout value is 7200 (two hours).

If a timeout is reached, a syslog entry is generated:

```
2010/10/25 11:06:29.096 155,18,4 <fns> COR_Listen -pt -s32769 -  
t3600 -d100 (1474810.42663.241 0x1680fa.a6a7) ...
```

COR\_Listen Warning: client @ 9.39.40.29 [62784] did not send RPC.

Connection terminated by server due to timeout (180 secs). error = 0,  
socket = 8

### **Add new utility "COR\_set\_prefetch\_timeout"**

This new tool takes zero or one parameter. If run with no parameters, the tool displays the current value of the prefetch timeout. When run with one parameter, the tool dynamically sets the new timeout value in seconds for prefetch threads.

For example:

COR\_set\_prefetch\_timeout                      display current timeout value

COR\_set\_prefetch\_timeout 300                set timeout value to 300

The range of the timeout value must be between 90 and 3600. If the value specified is out of range, then it is not operative and the application exits with a value of 1. Having a discipline on the exit return value of the utility makes it scriptable.

If run with -h or illegal parameters, it displays this usage message:

Usage: COR\_set\_prefetch\_timeout [time in seconds]

### Add optional maximum prefetch thread count

The maximum count can be between 75 and 200.

See **Add optional timeout value for prefetch threads** for configuration details.

## ssn

The ssn file contains the current system serial number. The valid range is 1000 - 2147483646.

### Important

---

Never modify the contents of the ssn file. The ssn is written onto all storage media, and the Image Services software will not operate properly if the incorrect system serial number is entered into the ssn file.

---

The ssn file is located in the following directory (separated according to platform):

- UNIX: **/fnsw/local/ssn**
- Windows: **\fnsw\_loc\ssn**

Enter **ssn** at the system/command prompt to view the system serial number output. The content of a typical ssn file is shown below.

12620417

## tapeconfig

The tapeconfig file is read by tape services software. The tapeconfig file is located in the following directory (separated according to platform):

- UNIX: `/fnsw/local/sd/1/tapeconfig`
- Windows: `\fnsw_loc\sd\1\tapeconfig`

This file contains the tape device configuration, specifying the default backup tape device. A separate tapeconfig file exists on each server that has a tape drive. The content of a typical tapeconfig file is shown below:

```
TAPE_CONFIG 10
backup tape1:earth:FileNet
tape1:earth:FileNet,5,0,(1,/dev/rmt/0m,/dev/rmt/0mn),\
(2,/dev/rmt/0m,/dev/rmt/0mn)
```

The file has the following structure for the parameters of the defined tape drive:

- One line containing the format level
- One line containing the name of the default backup tape drive
- One or more lines describing the name, type, capacity, densities, and device names of the various drives. (Type 1 is reel-to-reel, type 2 is cartridge. Low and high densities for cartridge tapes are specified with the values 1 [low] and 2 [high].)

---

**Note** Backslashes (\) indicate continued lines.

---

## setup\_config

Created by the fn\_setup program, the setup\_config file is used to determine the name of the DBA group.

The setup\_config file is located in the following directory (separated according to platform):

- UNIX: **/fnsw/local/setup\_config**
- Windows: **\fnsw\_loc\setup\_config**

The content of a typical setup\_config is shown below.

```
RDBMS_TYPE=1  
RDBMS_UID=oracle  
RDBMS_GID=dba  
RDBMS_HOME=/usr/oracle
```

## Applications

This chapter lists a number of FileNet application programs used by various components of the Image Services software. All programs that belong to the IS software are stored in the bin directory under the fnsw file system/directory for both the UNIX and Windows platforms:

**UNIX****/fnsw/bin****WIN****\fnsw\bin****Attention**

---

Do not install non-FileNet application programs in the /fnsw/bin subdirectory.

---

An IS server receives many different types of requests from its clients. A client may request to scan a batch of documents or store index values in a database. In order to efficiently manage the task of satisfying client requests, the IS software is broken down into separate tasks or services. Each IS service has a corresponding request handler. When a request arrives at a server, COR\_Listen deciphers part of the message to find out which service will handle the request. It passes the request to the appropriate request handler (stub).



## Application Program Descriptions

An application program with a name including a small “s” is a stub or request handler. For example BESs stands for Batch Entry services stub. A stub and a request handler are the same thing.

### Backup/Restore

#### **backup\_daemon**

The backup daemon program monitors backup partitions and makes periodic copies to avoid an overflow of data (Backup).

#### **BRBs**

The BRBs program handles I/O to a remote tape drive when backup is performed (Backup). The number of BRBs request handlers that can be loaded into memory is dictated by the BRBs entry in the server-Config file.

#### **BRRs**

The BRRs program handles the I/O to a remote tape drive when a restore is being performed (Restore). The number of BRRs request handlers that can be loaded into memory is dictated by the BRRs entry in the serverConfig file.

## Batch Entry Services

### BESs

The BESs program is the request handler that commences I/O with the batch tables of the Transient database on behalf of a request from a remote server. It is started up by the COR\_Listen program when a request for batch table information is received over the network from a client. The number of BESs request handlers that can be loaded into memory is dictated by the BESs entry in the serverConfig file.

## Cache Services

### CSMs

The CSMs program is the request handler that commences I/O with the cache tables of the Transient database on behalf of a request from a remote server. It is started as a result of client requests for I/O with the csm tables. The number of CSMs request handlers that can be loaded into memory is dictated by the CSMs entry in the serverConfig file.

### CSM\_daemon

The CSM\_daemon program is started by ds\_init at FileNet Image Services startup. This program also runs on Application servers with Cache services. Objects (documents) remain in the page (retrieval) cache for a configurable period of time. They are then eligible for deletion if the space is needed for another object. The daemon manages the aging process of objects. It also monitors the free space and performs the deletions of eligible objects.

## Document Services

### **bes\_commit**

The `bes_commit` program is the batch committal daemon. It is started up when the FileNet Image Services software is started. It checks the queue field of the `batch_stat_hdr` table every sixty seconds to see if a batch is ready for committal.

Having found a batch that is ready for committal, the daemon uses BES and BESI shared libraries to open the batch and perform an asynchronous committal.

If the cataloging is successful, the batch records are then deleted from the Transient database. If not, an error is posted, the `next_phase` field of the `batch_dyn_hdr` table is set to “recommit”, and the `phase_status` field is set to “has errors”.

### **DLIs**

The DLIs program begins I/O to the `docs`, `scalar_numbers`, and `cluster_map` tables of a DocLocator database on behalf of a request from a remote server. The number of DLIs request handlers that can be loaded into memory is dictated by the DLIs entry in the server-Config file.

### **DLSs**

The DLSs program begins I/O to the `docs` and `cluster_map` tables of a DocLocator database on behalf of a background job request from a remote server (optical disk to disk copy, disk import, etc.).

## **DOCs**

The DOCs program is the request handler that runs on the DocLocator server and implements entry points to Document services on behalf of a request from a remote server. The number of DOCs request handlers that can be loaded into memory is dictated by the DOCs entry in the serverConfig file.

## **ds\_init**

The ds\_init (Document services initialization process) program links to all the abstracts in Document services, builds the surface records, rebuilds all the demand queues and the committal queue, then starts all the processes listed in the as\_conf.g file.

## **ds\_notify**

The ds\_notify program notifies the client when a retrieval has been completed.

## **fbc\_commit**

The fbc\_commit program processes fast batch committals. This program doesn't need to run if the server has a storage library.

## **OSIs**

The OSIs program is the server process for OSI (receives requests from OSIr). It performs Permanent database I/O on an Storage Library server on behalf of a request from a DocLocator server. The number of OSIs request handlers that can be loaded into memory is dictated by the OSIs entry in the serverConfig file.

**OSSs**

The OSSs program performs permanent database I/O on a Storage Library server for background jobs (disk to disk copy, disk import, etc.) on behalf of a request from a DocLocator server. The number of OSSs request handlers that can be loaded into memory is dictated by the OSSs entry in the serverConfig file.

**RMKs**

The RMKs program is the server process for Remote MKF interfaces or RMK (receives requests from RMKr). It updates the Permanent database on behalf of a remote server when very low volume transactions are being performed. The number of RMKs request handlers that can be loaded into memory is dictated by the RMKs entry in the serverConfig file.

**rmt\_commit**

The rmt\_commit program is responsible for committing documents. The program is started by the ds\_init process. Multiple instances of this program can run as determined by the number of remote domains that are configured for remote committal.

**sas**

Support and development personnel use the sas program to determine if session timeouts are working properly.

**Important**

---

The sas tool should only be run for diagnostics and troubleshooting. Do not run this tool without the advice and consent of your IBM FileNet technical support representative.

---

The sas tool uses the following syntax:

```
sas { -d | -n | -s | -a }
```

- d Set session timeout to 0 (debug)
- n Set session timeout to 600 (normal)
- s Print session statistics
- p Print sessions active
- a Print all sessions

Stop the IS software before setting the session timeout value with the -d or -n options.

## Fax Services

### FSMs

Handles requests for Fax servers.

## File Services

### FILs

FILs is the File services server process.

## Index Services

### INXbg

The INXbg program is a background process that keeps INXI and INXD in memory and handles data dictionary updates. This program normally runs only on the server(s) where index services and bes\_commit are running and where dictionary builds and abstract linking are particularly time-consuming.

## **INXs**

The INXs program is a server stub program. This program receives remote procedure calls, does server-side session handling, and calls INXI on behalf of a remote client. The number of INXs request handlers that can be loaded into memory is dictated by the INXs entry in the serverConfig file.

## **INXu**

The INXu program runs the server update process. To avoid interference with on-going queries, a separate process is sometimes needed to perform updates and commit transactions.

# **MKF Database Services**

## **MKF\_clean**

The MKF\_clean program preserves MKF database integrity in case of abnormal termination.

# **Network Clearinghouse Services**

## **NCHs**

The NCHs program handles I/O to the NCH database on behalf of requests from remote servers. The number of NCHs request handlers that can be loaded into memory is dictated by the NCHs entry in the serverConfig file.

## **NCH\_daemon**

The NCH\_daemon is a program that only runs on the Root server and that transmits the Root server's protocol ID as each remote server is booted. There will be an NCH\_daemon running on the Root server for each type of network protocol in use (for example, TCP/IP).

The NCH\_daemon process listens for requests for NCH services and responds to these requests with a packet containing the network address at which the NCH database server can be reached.

## **Print Services**

### **PRI\_check**

The PRI\_check program checks to see if any printers that were marked as down are now responding. This function is done separately from the PRI\_daemon because it takes about a minute for the network software to time out.

### **PRI\_daemon**

The PRI\_daemon program handles time-related tasks. This program activates print requests which have been set to print at a certain time and deletes old print requests from the Transient database when these requests have been completed.

PRI\_daemon also de-queues records from the status queue and updates Print services data structures as appropriate. The status queue contains responses from the dtp process indicating that a document retrieval is complete, and responses from the print server indicating that a page has been printed.



**PRI\_notify**

The PRI\_notify program sends the print complete status from Print services back to client.

**PRIs**

The PRIs program receives the client request for Print services and puts requests into the Transient database on behalf of a request from a remote server. This program modifies and cancels print requests, and returns status on the activities of Print services. The number of PRIs request handlers that can be loaded into memory is dictated by the PRIs entry in the serverConfig file.

**PRI\_worker**

The PRI\_worker program sends the pages to be printed into the appropriate print cache and then sends a request to the printer imaging software. Up to 12 PRI\_worker programs can be started to handle system printing needs. PRI\_worker is not dedicated to any one printer; each PRI\_worker can handle requests for any printer.

**PSMs**

The PSMs program runs on the Print server and handles incoming documents to be printed

**Security Services****SEC\_daemon**

The SEC\_daemon program removes security logging information from the QMA queue and writes that information to the security log file. This

program is started by the FileNet software initialization script and runs only on the Root server.

## **SECs**

The SECs program services remote procedure calls (RPCs) to Security services, formats a response network message, sends a message back to the requesting client, and passes the request to the Security services target abstract, SECI (which provides the actual functionality). This program decrypts information received during user logon. The number of SECs request handlers that can be loaded into memory is dictated by the SECs entry in the serverConfig file.

## **Storage Library Services**

### **antcopy**

The antcopy program performs the document copy background process. This program copies a document's annotations from one optical disk to another.

### **del\_commit**

The del\_commit program migrates documents to optical disk on a delayed basis. This program only runs on the document locator server.

### **doccopy**

The doccopy program copies a document from one optical disk to another.

**docimport**

The docimport program imports document information and indexes from optical disk to MKF/RDBMS databases.

**dsched**

the dsched program schedules/controls the optical disk drives and the movement of the optical disks within each storage library.

**dtp**

The dtp program performs the data transfer process. This program handles the transfer of documents to and from optical disk.

**dtp\_tran**

The dtp\_tran program calls Cache services to update the Transient database when the writing of a document completes.

**opendocs**

The opendocs program determines statistics about optical disks. This program reports on the type, surface, open, closed, deleted, total, and percent of open statistics for an optical disk's documents.

**OSCs**

The OSCs program handles requests for Storage Library services from Document services. The number of OSCs request handlers that can be loaded into memory is dictated by the OSCs entry in the server-Config file.

**osi\_migrate**

The osi\_migrate program handles interaction with the Permanent database in regard to remote committal.

**sortod**

The sortod program is used to sort binary files. This program is used in such functions as an optical disk to disk copy.

**WorkFlo Queue Services****WQSs**

WQSs is the server process for WorkFlo Queue services. It receives requests from WorkFlo Queue services.

**Others****CDBs**

The CDB program handles requests to the configuration database files.

**CORs**

The CORs program handles requests to transmit/receive database information and images between servers.

**COR\_Listen**

COR\_Listen is the daemon that must be running at the server to accept an incoming request from the client. COR\_Listen transfers the

connection that it opened when it received the request from the network to the available server stub. COR\_Listen does not start the request handler itself but rather calls PPM to do the job.

For more information on PPM, go to [page 88](#).

### **dbintf**

The dbintf program performs I/O with the index database on behalf of a remote server when creating an index field or document class.

### **SKFs**

The SKFs program performs I/O with the doctaba table of the Index database when the dbverify utility is run from a server other than the Index server.

### **SQIs**

The SQIs program performs SKF database I/O for requests from remote servers.

### **TAPs**

The TAPs program performs I/O on a server that is interfacing with a remote tape drive.

### **timeoutmonitor**

The timeoutmonitor program monitors timeout waits on interlocks that are used by FileNet Image Services processes. The timeoutmonitor program, which is started automatically, must be running on all servers running FileNet Image Services and ISTK software.

## Shared libraries

This chapter describes a number of different shared libraries used by various components of the FileNet software. The entries in this chapter include shared library paths and naming conventions, followed by shared library definitions.

---

**Tip** Support personnel use the terms **abstract** or **shared object** when referring to a shared library. These are older terms but still appear in many tools.

---

A shared library is a collection of functions (for example, open file, close file, read from file, etc.). A running process calls the shared library when it needs to perform a function stored in the shared library. The system loads the functions into memory and the process accesses the code from there.

---

**Tip** The Windows platform uses shared libraries called DLLs (dynamically linked library).

---

The programs used by the FileNet Image Services software rely on shared libraries to perform their functions. If an error occurs, the shared library usually reports the problem. The shared libraries detailed in this chapter are separated according to the services.

## Shared library locations

All shared libraries that belong to the FileNet software are stored in the shobj directory under the fnsw filesystem. All shared libraries used by the FileNet Image Services software are located in the following path:

- UNIX Platforms: **/fnsw/lib/shobj <shared library name>**
- Windows: **\fnsw\lib\shobj <shared library name>**

(where **<shared library name>** is the name of the shared library being called.)

Each platform uses a unique convention for shared library names. The table illustrates an example of the shared library naming conventions, according to platform, for the BES shared library.

Platform	Name
AIX/6000	BES
HP-UX	libBES.sl
Solaris	libBES.so
Windows	BES.dll

## Shared library descriptions

Shared libraries are used by distributed applications and are often separated into remote (client) and local (server) objects.

### Important

---

Some shared library names include r or l (for example, BESr or BESl). The r indicates that the shared library is used remotely, and the l indicates the shared library is used locally. (Other letters appended to the shared library names are more or less arbitrary and do not indicate the location of the shared library execution.)

---

## All services

### COR

COR stands for Courier. The shared library is executed by a program to transport an image or database information across the network from the server to the client. A request handler calls COR and hands off the server response. COR is involved in getting the server response packed up to travel over the network back to the client.

### PPM

PPM is the Protocol Process Manager which reads the serverConfig file to find out the maximum number of request handlers that can be loaded into memory. PPM is the shared library that reports an error if a request handler cannot be started. PPM starts a request handler if a server stub is not available in memory. Once PPM either locates an available request handler or starts a new one, it advises COR\_Listen that the correct stub is now available. PPM is only a shared library or set of functions. It requires a program to call it. For more information on COR\_Listen, go to [page 84](#).



## **Batch Entry Services**

### **BES**

BES is used by the `bes_commit` program to interface with the batch tables. For instance, the BES code executes the functions needed to create, open, close, enqueue, dequeue, commit, delete, update, and find a batch.

### **BESI**

BESI is used by the BES shared library to perform I/O on the fields of the batch tables. Most batch service errors are reported by BESI.

## **Cache Services**

### **CSM**

CSM performs I/O with the csm tables of the Transient database.

### **CSMI**

CSMI assists CSM in handling I/O with the csm tables of the Transient database and all caches.

## **Document Services**

### **ASH**

ASH assists in the handle of caching which is used to associate an FileNet Image Services handle with a service handle and vice versa.

**BKG**

BKG manages the background requests. It handles the MKF background requests table (bkg\_requests), which is in the Transient database, and also starts and stops processes which are performing background requests (e.g., an optical disk to disk copy).

**CKS**

CKS computes a checksum on the data indicated.

**CMT**

CMT is the Document services shared library to commit documents on the document server.

**CNF**

CNF is the configuration shared library for Document services. It accesses the as\_conf.g and as\_conf.s files. Its entry points provide things like the number and names of the retrieval cache partitions, optical drives, storage libraries, etc.

**CNT**

CNT compiles performance statistics for Document Services. It includes counts for surface records, migrates from optical disk, pages committed, documents committed, fast batches committed, etc.

**DBL**

DBL is the Document services database logon shared library for MKF. It contains entry points for Document services to log on or off Permanent and Transient databases.

**DBP**

DBP controls memory buffers which are used for reading document headers and document pages. The size and number of buffers is controlled by the configuration file (see CNF). Manages images in memory being transferred to/from cache.

**DIA**

DIA contains routines which import document headers from optical disk into the MKF and RDBMS databases.

**DIG**

DIG is a diagnostic interface. It reserves a storage library for use by an instance of Document services.

**DLI**

DLI is the document locator interface shared library. It is used by the Storage Library servers to request actions by the document locator.

**DLII**

DLII performs the actual Permanent database I/O on the Storage Library server.

**DLIo**

DLIo is the locking shared library for DLI. It requests information for a Storage Library server from a DocLocator server.

**DLS**

DLS is the document locator service shared library. It provides document locator functions for the background tasks: document copy, document import, and annotation copy.

**DLSI**

DLSI performs the actual Permanent database I/O on the DocLocator server for the document locator service.

**DLSr**

DLSr requests a background job from a DocLocator server.

**DOC**

DOC is the shared library interface component for Document services. It processes all calls to the rest of the service. It is used to initiate I/O with the docs table of a Permanent database.

**DOCa**

DOCa implements Document services on an archive FileNet Image Services system.

**DOCb**

DOCb is Document services fast batch commit.

**DOCI**

DOCI is the target shared library. It implements the actual service—making DBMS calls. DOCI is the heart of Document services. It performs the actual I/O with the docs table of a Permanent database with the help of the DT shared library.

## **DOCp**

The DOCp module implements Document services on a portable database cache (PDB).

## **DSA**

DSA is the Document services shared library to interface with the annotations table.

## **DT**

DT manages insertions, deletions, and updates to the docs table on the Storage Library server.

## **FDT**

FDT accesses the family\_disk table of the Permanent database.

## **FLT**

FLT manages the family\_locator table which is used to find the proper Storage Library server ID given a family ID. Callers of this shared library must log on to the Document Locator Permanent database using RMK, and use RMK for all database transactions using this shared library.

## **HLT**

HLT handles high level tasks. It reassigns read/write requests, enables/disables disks, and updates optical disk families and SRF and ODT records.

**IS**

The IS module contains routines to facilitate access to an Image Service. It contains handle management routines, and also some miscellaneous functions.

**MKF**

MKF establishes contact for any program needing access to an MKF database.

**OSI**

OSI is the Storage Library server interface. It is used by the DocLocator and the Storage Library servers to request actions of a specified Storage Library server.

**OSII**

OSII is called by the OSIs request handler to perform Permanent database I/O on the Storage Library server.

**OSIr**

OSIr is called by the OSI shared library to request information from a Storage Library server.

**OSS**

OSS, the Storage Library server service shared library, provides Storage Library server functions for the background tasks: document copy, document import, and annotation copy.

## **OSSI**

OSSI is called by the OSSs request handler to perform Permanent database I/O on the Storage Library server.

## **QMA**

QMA, the Queue Manager shared library, manages memory queues for requesting Document services shared libraries.

## **RFT**

RFT is the Document services shared library to interface to the remote\_family table.

## **RMK**

RMK is the remote MKF interface. It is used by the DocLocator and the Storage Library servers to request low volume MKF operations remotely.

## **RMKr**

RMKr is used by remote servers to request a low-volume MKF data-base exchange.

## **SAS**

SAS, the Session Application Service shared library, assists the various services (DOC, INX, CSM, PRI, etc.) in manipulating sessions. A "session" is called active when a process has logged on to a service, and SAS has stored information about that logon on a remote system. SAS opens, closes, and deletes sessions for all of the services.

**SDC**

SDC is the interface for Single Document Committal (SDC) which provides for the committal, deletion, and security update of documents. All these functions would normally require calls to multiple services (e.g. INX and DOC) and/or calls to BES (which would require a series of calls to manage a batch). These calls are intended to simplify the interface to application services for clients which require simple functionality.

**SRF**

SRF, the Surface Record Format shared library, handles manipulation of surface records. There is one surface record for each optical disk surface either in the storage library, or which has pending demands. The surface records are memory structures which parallel the Optical Disk Table (ODT) records.

**Index Services****INX**

INX decides whether to make local or remote calls to the rest of the service.

**INXa**

INXa implements the actual index service in the archive database.

**INXD**

INXD reports if the RDD is empty or corrupted.



**INXI**

INXI implements the actual service—making RDBMS calls.

**INXr**

INXr performs RDD caching, Courier RPCs, and client-side session handling.

**Network ClearingHouse Services**

**NCH**

NCH initiates contact with the NCH database.

**NCHI**

NCHI performs I/O with the NCH database.

**NCHr**

NCHr requests access from the NCH database for non-root servers.

**SLMI**

SLMI directly accesses encrypted license key information in the NCH database.

## **Print Services**

### **PRI**

PRI initiates a validation of print options and the processing of fax requests.

### **PRII**

PRII performs a print option validation and processes fax requests.

## **Security Services**

### **SEC**

SEC accesses information in the security database.

### **SECI**

SECI performs security DB searches, updates, deletes, adds, licensing checks.

### **SECr**

SECr processes requests for Security services for a client (non-root) server.

### **SMM**

SMM allocates/de-allocates chunks of shared memory on a per table basis.

**SSD**

SSD provides interface between SECI and the security database.

**SSU**

SSU encrypts/decrypts data, performs object name conversion.

**Storage Library Services**

**ARM**

ARM contains routines that control mechanical functions of a storage library.

**CLT**

CLT is the shared library for the DS III Cluster Map Table.

**DIS**

DIS assigns requests to optical disk surfaces.

**ODF**

ODF inspects/modifies optical disk descriptor without using record structure.

**ODS**

ODS provides functions that access the family\_disk table on the Storage Library server.

**ODT**

ODT manages optical disk table surf\_dyn\_info and surf\_stat\_info.

**ODU**

ODU contains routines to access the optical disk in a formatted manner.

**ODX**

ODX contains routines to access the optical disk in an unformatted manner.

**OFA**

OFA writes COLD documents to optical disk families.

**OPM**

OPM transmits and receives messages for the Storage Library Control program.

**OSA**

OSA allows program to get/set status about storage libraries' slots, drives, etc.

**SLT**

SLT manages the surf\_locator table.

## **SNT**

SNT maintains the scalar\_numbers table of the Permanent database.

## **WRT**

WRT manages the write requests table.

# **System Configuration**

## **CDB**

CDB accesses IMS\_xxx.cdb file. It creates the initial configuration database file.

# **Miscellaneous System V**

## **ORA**

ORA performs I/O with Oracle database used by GDB Oracle compatibility layer.

# **System Messages**

## **ERM**

ERM is used by any program that requires access to the error message file.

## Permanent database tables and contents

Document and Storage Library services access the information stored in the tables of the Permanent database. The Permanent database contains information about each document that is written to optical disk as well as about the optical disks themselves. The Permanent database acts as a directory for the storage library. Many tables in this database are referenced during the document retrieval process.

This chapter discusses the following Permanent database tables:

- annot\_log
- annotations
- cluster\_map
- docs
- family\_disk
- family\_locator
- lib\_surfaces
- od\_stats
- remote\_family
- scalar\_numbers
- surf\_dyn\_info
- surf\_info
- surf\_locator
- surf\_stat\_info
- surface\_activity

## Tables and contents

Each table (and its contents) in the Permanent database are discussed in this section; some of the tables contain examples of the contents. You can use the MKF\_tool to view these tables and contents directly.

### annot\_log

The **annot\_log** (annotation export log) table maintains a record of each annotation that is exported to the Content Engine of a P8 system in a Content Federation Services for FileNet Image Services (CFS-IS) environment. The annotation text is also stored in the annotations table.

Field Name	Field Description
seq_key cd_os_id seq1 seq2	Unique search key composed of ce_os_id (Content Engine Object Store ID), seq1, and seq2 keys - (group).
doc_id	The identification number of the document that contains the annotation.
page	The page number on which the annotation occurs.
ssn	The serial number of the FileNet Image Services system on which the document originated.
annot_id	A system-assigned identification number for the annotation, based on the document identification number, page number, and ssn of the document. The first annotation on a page always has a value of 1 in this column.
action	The action performed on the annotation: 1 = insert 2 = export 3 = delete 4 = update
sec_rd	The id of the group that has read access to the annotation.

Field Name	Field Description, Continued
sec_wr	The id of the group that has write access to the annotation.
sec_ax	The id of the group that has append/execute access to the annotation.
data_len	The number of bytes in the data column.
data	The annotation (in hexadecimal). Can be up to 800 bytes long.
create_date	The date that the annotation was created. (Used for CFS-IS bidirectional committal support.)
modify_date	The date that the annotation was last modified. (Used for CFS-IS bidirectional committal support.)

## annotations

The **annotations** table maintains a record of each annotation and its associated document and page. The actual annotation text is stored in a hexadecimal format.

Field Name	Field Description
annot_key	Unique search key composed of doc_id, page, ssn, and annot_id keys - (group).
doc_id	The identification number of the document that contains the annotation.
page	The page number of the document.
ssn	The serial number of the FileNet Image Services system on which the document originated.
annot_id	A system-assigned identification number for the annotation, based on the document identification number, page number, and ssn of the document. The first annotation on a page always has a value of 1 in this column.
create_date	The date and time that the annotation was created.



Field Name	Field Description, Continued
modify_date	The date and time that the annotation was last modified.
security_len	The number of bytes in the security_info column.
security_info	Encoded access restrictions - read, write, execute/append.
capability	Capability of user updating the annotation. Null if no update is in progress.
data_len	The number of bytes in the data column.
data	The annotation (in hexadecimal). Can be up to 800 bytes long.
ce_annot_guid	The GUID for the annotation on Content Engine. (Used for CFS-IS bidirectional committal support.)

The following example illustrates a typical record in the **annotations** table:

```

doc_id.....8639401  page..... 1
ssn.....2620417
annot_id.....1  data_len.....27
create_date....752962296 => 99/11/10 12:11:36
modify_date....752962296 => 99/11/10 12:11:36
data          [0] ....0x0400020001010004000000000020004000000000c0001020300

```

## cluster\_map

The **cluster\_map** table maintains a record of each cluster identifier and the optical disk surface identifier associated with it. The table is referenced during the process of committal. Data I/O with this table is primarily accomplished through the use of the CLT shared library.

Field Name	Field Description
cluster_key	Unique search key composed of cluster_space and cluster_id keys - (group).
cluster_space	0,1,... Identifies the cluster number of the surface.

Field Name	Field Description, Continued
cluster_id	The id number of the cluster. It is a 6 byte id number derived from the index_cluster table of the index database.
surface_id	The current optical disk surface that is involved in the cluster operation.

The following example shows a typical record from the **cluster\_map** table:

cluster_space	cluster_id	surface_id
-----	-----	-----
1	0xe0cae8cae400	3008

## docs

The **docs** table maintains a record for each document id number and its associated optical disk surface. The table is updated during committal and also when writing to optical disk. The table is referenced whenever a document is retrieved.

Field Name	Field Description
doc_id	Unique search key - document id number.
status_1	NULL or 0 = normal 1 = bad (unreadable) 2 = deleted 3 = not migrated to optical disk
status_2	NULL or 0 = normal 1 = bad (unreadable)
pages	The number of pages in the document.
contig_sectors	The number of contiguous optical disk sectors in the last page of the document.

Field Name	Field Description, Continued
surface_id_1	The surface number of the “family” disk for this document.
offset_1	The sector of the “family” disk containing this documents long descriptor file.
surface_id_2	The surface number of the “tranlog” disk for this document.
offset_2	The sector of the “tranlog” disk containing this documents long descriptor file.
security_len	The number of bytes in the security_info column.
security_info	Encoded access restrictions - read, write, execute/append.
orig_doc	Unique search key composed of orig_ssn and orig_doc_id keys - (group). This number is NULL if the documents are local.
orig_ssn	System serial number of the system the document was originally from. This number is NULL if the documents are local.
orig_doc_id	Document id number on original system. This number is NULL if the documents are local.
cache_id	Cache id when not on optical disk.
back_contig	Number of contiguous sectors (page 0 last).
duration	Time at which the object is to be written to optical disk. This field was added to accommodate delayed migration.
ce_os_id	Content Engine Object Store id
sds_id and date	SDS object id and date composed of sds_id and sds_obj_date (group)
sds_id	SDS unit number
sds_obj_date	Time at which the SDS object was archived to the ISDS device.
sds_obj_id_len	Length of the SDS object id number.
sds_obj_id	SDS object id number. (Centera or DR550 CLIP ID)
sds_obj_blob_sz	Size of an SDS object blob (binary large object).
sds_orig_doc_id	SDS document id number on original system. Indicates a foreign document. This number is NULL if the document is local.

Field Name	Field Description, Continued
sds_retent_typ	SDS retention type 1 = none 2 = infinite 3 = chronological (f_deletedate or f_archivedate is set) 4 = only on Centera with ARM (EBR + EBR supported device) 5 = virtual EBR (EBR + non-EBR supported device)
sds_retent_value	SDS retention expiration date and time. Not set for none (1), infinite (2), and EBR + EBR supported device (4). Always set for chronological (3) and set initially to default setting for virtual EBR (5).
sds_held	If set to 1, document is held.
sds_migrating	If set to 1, document is migrating to SDS.
last_annot_id	If non-zero, tracks the last annotation id.

The following example shows a typical record from the **docs** table.

doc_id.....101200	pages.....1	surface_id_1.....3004
offset_1.....101706	surface_id_2.....3006	offset_2.....234756
security_len.....3	orig_ssn.....13476	orig_doc_id.....114537
security_info [0]....0x8181d8		

The following example shows a record from the **docs** table on a system with an Hitachi Content Archive Platform (HCAP) SDS device.

doc_id.....122177	status_1.....3	pages.... ..5
surface_id_1.....1002	offset_1.....0	surface_id_2. . .0
offset_2.....0	cache_id.....1	sds_id.....3
sds_obj_blob_sz.....7348	sds_retent_typ.....1	
sds_obj_date...1223676236	=> 10/10/2008 15:03:56	

The following examples show records from the **docs** table on a system with an EMC Centera SDS device with Advanced Retention Management (ARM).

```
doc_id.....104845  status_1.....3  pages.....1
contig_sectors.....2  surface_id_1.....1002  offset_1.....0
surface_id_2.....0  offset_2.....0  cache_id.....1
sds_id.....2  sds_obj_id_len.....53  sds_obj_blob_sz...1952
sds_retent_typ. ....3
sds_obj_date...1215541164 => 07/08/2008 11:19:24
sds_obj_id [0]...0x3856544b4c4f4a52434a33384f653335534f41494a304b4855
           [25]...0x564a4734313354355639424a4e30503941494e533634453930
           [50]...0x4b3434 [50]...0x4b3434
sds_obj_id sds_retent_value.1237138191 => 03/15/2009 10:29:51
```

```
doc_id.....104846  pages.....1  contig_sectors.....2
surface_id_1.....3044  offset_1.....4002  surface_id_2.....0
offset_2.....0  sds_id.....2  sds_obj_id_len.....53
sds_obj_blob_sz.....1952  sds_retent_typ....4
sds_obj_date...1215541164 => 07/08/2008 11:19:24
sds_obj_id [0]...0x435634393738454c4e4338313165444e31484b384938364c50
           [25]...0x414d4734313343314b53435356304c324b32565346524e4544
           [50]...0x394350
```

## family\_disk

The **family\_disk** table keeps a record of each optical disk family on the system. It is referenced during committal. Data I/O with this table is primarily accomplished through the use of the FDT shared library.

Field Name	Field Description
family_name	User-assigned optical disk family name.
family_id	System-assigned optical disk family identification number.
sides	1 = single sided disk 2 = double sided disk
interleaved	Write this number of side A's before side B's.
des_cur_surfs	Desired number of current optical disk surfaces that are available for a write operation.
num_cur_surfs	Actual number of current optical disk surfaces that are available for a write operation.
current_surfs	Current optical disk surface number available for a write operation.
prev_surfs	The optical disk surface number previously written to. (Used when interleaving).
preferred_osar	ID (a through h) of the storage library that contains the optical disk called out in the "current_surfs" column.
num_futr_surfs	The number of future surfaces reserved for this optical disk family.
future_surfs	Future optical disk surface number reserved for this family.
next_ordinal	(not used)
disk_type	0 = unspecified disk type 1 or NULL = Hitachi 12" 2.6 GB WORM 2 = Hitachi 5" 600 MB WORM 3 = Hitachi 5" 600 MB erasable 4 = OSI (LMSI) 12" 2 GB WORM 5 = Hitachi 12" 7 GB high density WORM

Field Name	Field Description, Continued
	6 = OSI (LMSI) 12" 5.5 GB high density WORM 7 = HP 5" erasable 8 = HP 5" WORM 9 = Standard 5" 1.3 GB Erasable 10 = Standard 5" 1.3 GB WORM 11 = LMSI 12" 12 GB LM6000 WORM 12 = Standard 5" 2.6 GB Erasable 13 = Standard 5" 2.6 GB WORM 14 = IBM 5" 2.6 GB permanent WORM 15 = Standard 5" 5.2 GB Erasable 16 = Standard 5" 5.2 GB WORM 17 = IBM 5" 5.2 GB permanent WORM 18 = LMSI 12" 30 GB LM8000 WORM 19 = Standard 5" 9.1 GB Erasable 20 = Standard 5" 9.1 GB WORM
	21 = MSAR 1 GB 22 = MSAR 2 GB 23 = MSAR 4 GB 24 = MSAR 8 GB 25 = MSAR 16 GB 26 = MSAR 32 GB 27 = 5" UDO 30 GB Erasable disk 28 = 5" UDO 30 GB WORM disk 29 = 5" UDO 60 GB Erasable disk 30 = 5" UDO 60 GB WORM disk
is_primary	Is this the primary family? 0 = False (transaction) 1 = True (primary)
prev_0_surfs	Previous surfaces for cur_surf (current surfaces).
tran_families	Family ids of the tranlogs for the family. If all zero, no tran logging.

Field Name	Field Description, Continued
no_migrate_fb	<p>Indicates whether fast batch committal documents will be migrated to storage media or not. The value in this field has no effect on cache-only systems because the documents will always remain in cache.</p> <p>0 = migrate to storage media 1 = DO NOT migrate to storage media</p> <p>This field is set to (0) when at least one storage device is present on the system. After document committal, write requests will be executed to migrate the documents to the storage media.</p> <p>This field is set to (1) when the FileNet Image Services system has no external storage devices and committed documents will not be migrated.</p>
sds_id	SDS id
sds_only	Indicates that the document is stored only on the SDS device.

The following example shows a typical record from the **family\_disk** table.

family_id.....41	interleaved.....1	des_cur_surfs.....0
num_cur_surfs.....0	current_surfs.....0	prev_surfs.....0
preferred_osar....."	num_futr_surfs.....0	future_surfs.....0
disk_type.....2	is_primary.....1	
family_name..."CloseDocs"		
tran_families [0]....0,0,0,0,0,0,0,0		

family\_locator

The **family\_locator** table is used in a multiple Storage Library server environment. The records within the table associate optical disk family id numbers with a Storage Library server. It is referenced during the act



of committal. Data I/O with this table is primarily accomplished through the use of the FLT shared library.

Field Name	Field Description
family_name	Unique search key - user-defined name for an optical disk family.
family_id	System assigned number for an optical disk family.
num_ids	The number of Storage Library servers that use the disks of a given optical disk family.
server_ids	System assigned Storage Library server id number. Used to designate which Storage Library server relates to a given optical disk family.

## lib\_surfaces

The **lib\_surfaces** table is used to keep track of the location of optical drive surfaces.

Field Name	Field Description
surface_id	Optical disk surface id number.
loc_key	The location key of the surface, composed of the library id, unit_indicator, and unit number - (group). This field is NULL when the surface is "out of box."
library	The library identifier
unit_indicator	The unit indicator 1 = lsi_in_slot 2 = lsi_in_drive 3 = lsi_in_gripper
unit_number	The unit number is composed of the slot number, drive number and gripper number.

Field Name	Field Description, Continued
out_of_box	Indicates whether the surface was removed from the storage library or not. NULL or 0 = in box. 2 = out of box because of BKG processing 3 = out of box because it was ejected 5 = out of box, but cannot update MSAR checksums due to an error
name_length	The length of the surf_name file.
surf_file	The surface file name. The name must include the absolute path and have a .dat extension.
disk_type	See the description of the table <a href="#">“family_disk” on page 110</a> for a list of disk types.

## od\_stats

The **od\_stats** table is a statistical record table that maintains a log of optical disk write errors. Data I/O with this table is primarily accomplished through the use of the ODS shared library.

Field Name	Field Description
surface_id	Optical disk surface number that incurred an error.
osardrive	Unique search key composed of osar_num and drive_num keys - (group).
osar_num	The number of the storage library that incurred the error.
drive_num	The optical drive which incurred the error.
num_rewrites	The number of 1k sectors that incurred an error during a write operation.
last_day_sec	The date and time of the last I/O error that occurred.
num_elt	The number of elements expressed in the days_errors and total_errors columns.

Field Name	Field Description, Continued
days_errors	Each number (also known as an element), in this column correlates to a specific type of optical I/O error. For each instance that a particular type of error occurs, the element increments. The value for each element is equal to the total number of errors that occurred since midnight of the day expressed in the last_day_sec column.
total_errors	This column is similar to days_errors except that the error counts shown are for the lifetime of a particular drive or surface, not just since midnight of a given day.

## remote\_family

The presence of records in the **remote\_family** table indicates that committal operations must be done on the remote domain and family name indicated in addition to the local family.

Field Name	Field Description
fam_key	Unique search key composed of family_id and rec_num keys - (group).
family_id	ID of local family.
rec_num	Number of this entry in the table.
rmt_domain	Name of remote domain.
rmt_org	Name of remote organization.
rmt_fam_name	Name of remote family (may be different than local family, but must be a primary family ID).

The following example shows a typical record from the **remote\_family** table:

```
family_id.....52  rec_num.....1
rmt_domain....."costa6"
rmt_fam_name....."Clustering"
```

scalar\_numbers

The **scalar\_numbers** table maintains a record of the next available image id number (for scanning), optical disk surface number (for writing to optical), and background job request number (for optical disk copying).

Field Name	Field Description
rec_num	Unique search key. The value is always 0.
next_doc	The next available image id number to be used for scanning.
next_surface	The next available optical disk surface number that can be assigned.
next_bkg_job	The next available job id number that can be assigned to a background job (optical disk copy, annotation copy, optical disk import).
db_version	Version of the MKF database.
boot_num	Number of times the system has been booted.
char_set	The character set of this database.

The following example shows a typical record from the **scalar\_numbers** table.

rec_num.....0	next_doc.....8752760	next_surface.....3338
next_bkg_job.....132	db_version.....4	boot_num.....1477
char_set.....11		

surf\_dyn\_info

The **surf\_dyn\_info** table keeps a record of available space on each optical disk surface. It is referenced and updated when writing to

optical disk. Data I/O with this table is primarily accomplished through the use of the ODT shared library.

Field Name	Field Description
surface_id	Unique search key - optical disk surface id number.
next_sector	Next available sector on the optical disk surface that can be written to. This is also known as the high-water mark.
num_act_docs	Number of retrievable documents already on optical disk surface. See the important note at the end of this table.
num_del_docs	Number of non-retrievable (deleted) documents on optical disk surface.
num_clusters	Number of clusters involved with this optical disk surface.
last_desc_sect	This is the last sector of the data area, of the last extent, of the short descriptor file on the optical disk surface. The value of this column will be 0 if no short descriptor file has been created yet.
nxt_short_desc	Next available sector in the short descriptor file. The value of this column will be 0 if no short descriptor file has been created yet.
num_unwrt_desc	Number of attempted but unwritten sectors in the short descriptor file. The value of this column should always be 0. Any other number indicates that some type of interruption or crash occurred while a short descriptor file was being written.
short_doc_ids	The document id numbers of the documents that have been written to disk, but have not yet had a short descriptor file written.
short_sects	The sector numbers of the long descriptor files that have not yet had a short descriptor file written.
num_used_sects	The number of 1024 byte sectors occupied on optical disk surface.
num_pages	Total number of document pages on the surface.
next_fid_sect	The next available sector that can be written to in the optical disk directory file.
last_fid_sect	The last available sector that can be written to in the optical disk directory file.

Field Name	Field Description, Continued
cur_fid_off	Offset in the directory of the current entry.
inbox_priority	Priority which keeps the surface in the storage library. This is only set for side 1 surfaces.

The following example shows a typical record from the **surf\_dyn\_info** table.

surface_id.....3001	next_sector.....0	num_act_docs.....0
num_del_docs.....0	num_clusters.....0	last_desc_sect.....0
nxt_short_desc.....0	num_unwrt_desc.....0	num_used_sects.....0
num_pages.....0	next_fid_sect.....0	last_fid_sect.....0
cur_fid_off.....0	inbox_priority.....128	
short_doc_ids		
[0] .0,		
[29] ...0,0,0		
short_sects		
[0] .0,		
[29] ...0,0,0		

**Important** The num\_act\_docs value refers to the number of active documents on a given surface. An active surface is the primary surface and the active tranlog. Images on either of these active surfaces can be directly accessed by IS. The num\_act\_docs value for the primary surface and the active tranlog will be incremented after a successful image committal.

Alternate (or secondary) tranlogs are not considered active. Images stored in these alternate tranlogs must be imported into the system in order to access these images. If your system is set up to use multiple tranlogs, only one of them contains active documents and the last tranlog selected is considered to be the active tranlog. The num\_act\_

docs value for any alternate tranlog(s) is not incremented after a successful image committal.

---

## surf\_info

This is not a database table. It is an alias (view) used by MKF\_tool to retrieve information from the **surf\_dyn\_info** and **surf\_stat\_info** tables. For example, the following command retrieves all of the information from both the surf\_dyn\_info and surf\_stat\_info tables concerning optical disk surface number 3005.

```
select surf_info * where surface_id = 3005
```

## surf\_locator

The **surf\_locator** table is used in a multiple Storage Library server environment. The records within the table associate optical disk surface id numbers with a Storage Library server. This table is referenced during committal and document retrieval. Data I/O with this table is primarily accomplished through the use of the SLT shared library.

Field Name	Field Description
surface_id	Optical disk surface id number.
server_id	System assigned number of the Storage Library server controlling the library that contains this optical disk.
orig_surf	Unique search key composed of orig_ssn and orig_surf_id keys - (group).
orig_ssn	If imported, the original system serial number that this disk came from.
orig_surf_id	ID of the original surface. This column is used in conjunction with "orig_ssn" for certain search operations.
fam_id	Family ID of the surface.

Field Name	Field Description, Continued
disk_type	<p>0 = unspecified disk type  1 or NULL = Hitachi 12" 2.6 GB WORM  2 = Hitachi 5" 600 MB WORM  3 = Hitachi 5" 600 MB erasable  4 = OSI (LMSI) 12" 2 GB WORM  5 = Hitachi 12" 7 GB high density WORM  6 = OSI (LMSI) 12" 5.5 GB high density WORM  7 = HP 5" erasable  8 = HP 5" WORM  9 = Standard 5" 1.3 GB Erasable  10 = Standard 5" 1.3 GB WORM  11 = LMSI 12" 12 GB LM6000 WORM  12 = Standard 5" 2.6 GB Erasable  13 = Standard 5" 2.6 GB WORM  14 = IBM 5" 2.6 GB permanent WORM  15 = Standard 5" 5.2 GB Erasable  16 = Standard 5" 5.2 GB WORM  17 = IBM 5" 5.2 GB permanent WORM  18 = LMSI 12" 30 GB LM8000 WORM  19 = Standard 5" 9.1 GB Erasable  20 = Standard 5" 9.1 GB WORM  21 = MSAR 1 GB  22 = MSAR 2 GB  23 = MSAR 4 GB  24 = MSAR 8 GB  25 = MSAR 16 GB</p>
	<p>26 = MSAR 32 GB  27 = 5" UDO 30 GB Erasable disk  28 = 5" UDO 30 GB WORM disk  29 = 5" UDO 60 GB Erasable disk  30 = 5" UDO 60 GB WORM disk</p>
num_act_docs	Number of documents for each side.



## surf\_stat\_info

The **surf\_stat\_info** table keeps a general record about each optical disk surface. Each record contains information like family id, surface id, disk status, the date it was labeled, etc. The table is referenced when an optical disk retrieval is needed. Data I/O with this table is accomplished primarily through the use of the ODT shared library.

Field Name	Field Description
surface_id	Optical disk surface id number.
family_id	System assigned optical disk family id number for the surface.
series_ordinal	Used to preformat disks.
disk_status	0 = normal 1 = surface terminated
write_protect	0 = write enabled 1 = write protected
sides	1 = single sided disk 2 = double sided disk
label_date	When the volume label was written to the optical disk.
full_date	The date and time the surface became full.
archive_date	The date and time the last active document was deleted.
last_disk_sect	The last usable optical disk sector on the surface.
orig_surf	Unique search key composed of orig_ssn and orig_surfid keys - (group).
orig_ssn	If imported, the original system serial number that this disk came from.
orig_surfid	ID of the original surface. This column is used in conjunction with "orig_ssn" for certain search operations.
old_hw	Value of "next_sector" when the optical disk is converted from 1.8 to 2.2, or NULL.
need_verify	Whether the imported disk was verified.

Field Name	Field Description, Continued
disk_type	0 = unspecified disk type 1 or NULL = Hitachi 12" 2.6 GB WORM 2 = Hitachi 5" 600 MB WORM 3 = Hitachi 5" 600 MB erasable 4 = OSI (LMSI) 12" 2 GB WORM 5 = Hitachi 12" 7 GB high density WORM 6 = OSI (LMSI) 12" 5.5 GB high density WORM 7 = HP 5" erasable 8 = HP 5" WORM 9 = Standard 5" 1.3 GB Erasable 10 = Standard 5" 1.3 GB WORM 11 = LMSI 12" 12 GB LM6000 WORM 12 = Standard 5" 2.6 GB Erasable 13 = Standard 5" 2.6 GB WORM 14 = IBM 5" 2.6 GB permanent WORM 15 = Standard 5" 5.2 GB Erasable 16 = Standard 5" 5.2 GB WORM 17 = IBM 5" 5.2 GB permanent WORM 18 = LMSI 12" 30 GB LM8000 WORM 19 = Standard 5" 9.1 GB Erasable 20 = Standard 5" 9.1 GB WORM
	21 = MSAR 1 GB 22 = MSAR 2 GB 23 = MSAR 4 GB 24 = MSAR 8 GB 25 = MSAR 16 GB 26 = MSAR 32 GB 27 = 5" UDO 30 GB Erasable disk 28 = 5" UDO 30 GB WORM disk 29 = 5" UDO 60 GB Erasable disk 30 = 5" UDO 60 GB WORM disk

Field Name	Field Description, Continued
incorporated	0 = not incorporated
is_updated	1 = Short descriptor has been updated.

The following example shows a typical record from the **surf\_stat\_info** table:

surface_id.....3004	family_id.....3	disk_status.....0
write_protect.....0	sides.....2	last_disk_sect..287983
orig_ssn.....13476	orig_surfid.....3004	disk_type.....2
label_date.....638480831	=> 99/3/26 11:47:11	

**surface\_activity**

The **surface\_activity** table stores a surface activity record for each surface. It also records the date when activity occurred on the system. Surface activity logging must be enabled for this information to be recorded.

This table includes special records that track surface activity logging enabling and disabling. These records have a surface id of ASE\_MIN\_SURF\_ID - 1 along with the date on which the record was created. Other fields are NULL for the logging enable records. When the system comes up, if the logging state is not the same as the last one set in the file, a new record is added.

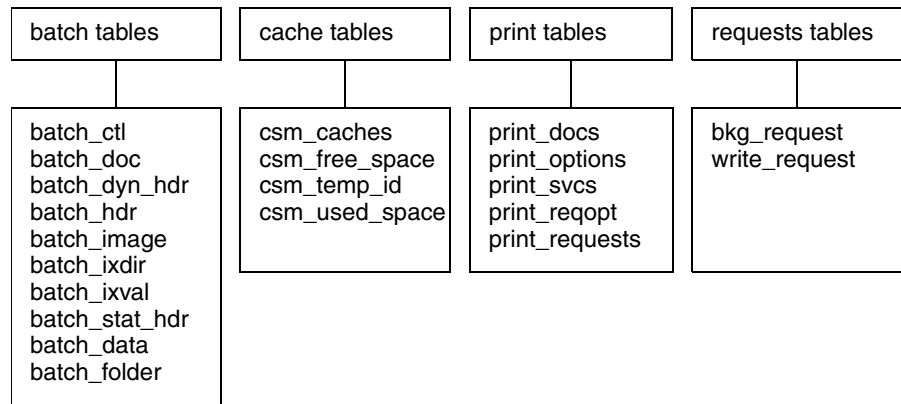
Field Name	Field Description
surf_date	The surface id and activity date. Composed of surface and act_date - (group).
surface	The surface id on the system that the disk is from.
act_date	The date activity occurred on the system.

Field Name	Field Description, Continued
mounts	Number of mounts of this surface on this day.
reads	Number of reads for surface on this day.
writes	Number of writes for surface on this day.

## Transient database tables and contents

Batch, Cache, and Print services use the tables in the Transient database. The Transient database contains information about documents that are in transition during document entry, retrieval cache, and print cache. Due to its temporary nature, this database is generally assigned a small amount of magnetic disk space.

This chapter discusses the following Transient database tables:



The tables are detailed in sections separated according to the FileNet Image Services functions they support.

## Batch tables and contents

Each table (and its contents) in the Transient database used by the batch service is discussed in this section; some of the tables contain examples of the contents. You can use the MKF\_tool to view these tables and contents directly.

### batch\_ctl

The **batch\_ctl** table contains the next available batch\_id number that can be used. Although it is primarily updated during the batch creation phase, anytime a batch is accessed, the batch\_open\_id column of this table is updated.

Field Name	Field Description
ctl_index	Unique search key for the table.
batch_id	Batch identification number used in all of the batch tables of the Transient database.
batch_name_id	Next available batch name that will automatically be assigned to a batch if an operator does not choose one.
batch_open_id	The last time that the "batch_id" column was accessed.
folder_id	Next available folder id.

The following example shows a typical record from the **batch\_ctl** table:

ctl_index	batch_id	batch_name_id	batch_open_id
-----	-----	-----	-----
1	11	6	757791177

## batch\_doc

The **batch\_doc** table keeps a record of multiple documents that have been scanned under a single batch id number. Information such as document type, number of pages, phase status, optical disk family number, etc., is recorded. The record is cleared from this table upon the successful committal and cataloging of the batch.

Field Name	Field Description
doc_index	Unique search key composed of batch_id and doc_id keys - (group).
batch_id	Identification number of the batch.
doc_id	The document id number.
doc_type	The type of document. 0 = image 1 = text 2 = fill in form 3 = mixed 9 = separator sheet
num_pages	Number of pages in the document.
num_indices	Number of indexing fields associated with this document.
num_reqd_indcs	Number of mandatory indexing fields associated with this document.
phase_num	The phase in which the “phase_error” occurred. Will display 65535 if no error has occurred. See the table <b>“Document Phases” on page 131</b> .
phase_error	0 = no error 1 = error occurred while performing this “phase” of the document.
phase_status	The current status for each of the phases that the document could experience. See the table <b>“Phase Status” on page 131</b> .
fam_id	Optical disk family id number to which this document belongs.
cluster_space	Identification number of the cluster operation that is involved. This feature is currently not used.

Field Name	Field Description, Continued
cluster_id	Identification number of the cluster that is involved. Value is 0 if clustering is not involved.
ocr_schema	Reserved for use by OCR.
text_len	The number of bytes in the text_data column.
text_data	Reserved. If the value of text_len = 0, the field is not displayed.

The following example shows a typical record from the **batch\_doc** table:

batch_id.....8	doc_id.....5	doc_type.....0
num_pages.....2	num_indices.....1	num_reqd_indcs.....1
phase_num.....0	phase_error.....0	fam_id.....0
cluster_space.....0	ocr_schema.....0	text_len.....0
phase_status [0] ....0,0,0,0,0,3,0,0,0,0,0		
cluster_id [0] ....0x000000000000		

### batch\_dyn\_hdr

A dynamic record is created within the **batch\_dyn\_hdr** table for each batch that is created. This record is updated as the status of the batch changes, for example, when it is scanned, indexed, verified, or committed. The record is cleared from this table upon the successful committal and cataloging of the batch.

Field Name	Field Description
batch_id	Unique search key for the table - batch identification number.
batch_name	Name of the batch entered by the user at the start of document entry.
exp_pages	Expected number of pages to be scanned in for the batch.
exp_docs	Expected number of documents in the batch.
pages_per_doc	Number of pages/document in batch. 0 = varies per document.



Field Name	Field Description, Continued
double_sided	0 = single sided pages 1 = double sided pages
verify_images	0 = no image verify necessary 1 = image verification needed
verify_indices	0 = no index verify necessary 1 = index verification needed
committal_type	0 = do not migrate 1 = migrate to optical disk
batch_total	0 = batch totaling unnecessary 1 = batch totaling needed
this_phase	Current phase that the document is in. See the <a href="#">“Document Phases” on page 131</a> table for values.
next_phase	Next phase for that document. See the <a href="#">“Document Phases” on page 131</a> table for values.
phase_status	The current status for each of the eleven possible phases that the document could experience. See the table <a href="#">“Phase Status” on page 131</a> for possible values.
phase_1st_time	The last time each of the eleven possible phases for this document was attempted. See the table <a href="#">“Phase Status” on page 131</a> for possible values.
phase_user_id	ID number of operator who last worked on given phase.
sort_order	Sort order for the pages in cache. 0 = Sequential from Page 0 1 = Reverse order
ocr_enable	1 = OCR enabled
auto_index	Used by OCR
text_len	The number of characters in the text_data field.
text_data	Reserved.
migr_dest_name	Name of the cache that an object will be migrated to (other than page cache).
migrate_delay	Number of seconds to delay before migrating an object from the cache to the optical disk. Only meaningful if committal type is COMMIT_NORMAL. A value of 4294967295 = do not migrate.

The following example shows a typical record from the **batch\_dyn\_hdr** table:

```

batch_id.....9   exp_pages.....0   exp_
docs.....0
pages_per_doc.....5   double_sided.....0   verify_
images.....0
verify_indices.....1   committal_type.....1   batch_
total.....0
this_phase.....5   next_phase.....5   sort_
order.....0
ocr_enable.....0   auto_index.....0   text_
len.....54
migrate_delay.....0
batch_name....."pmc2"
phase_status  [0] ....3,3,3,3,3,2,0,4,0,4,4
phase_lst_time [0] .....757619781 => 99/1/3 09:56:21
               [1] .....757619947 => 99/1/3 09:59:07
               [2] .....757787161 => 99/1/5 08:26:01
               [3] ....-2000000000 =>
               [4] .....757790895 => 99/1/5 09:28:15
               [5] .....757791193 => 99/1/5 09:33:13
               [6] .....757789039 => 99/1/5 08:57:19
               [7] ....-2000000000 =>
               [8] ....-2000000000 =>
               [9] ....-2000000000 =>
               [10] ...-2000000000 =>
phase_user_id [0] ....21,11,11,0,11,11,11,0,0,0,0
text_data....." "
migr_dest_name....."::"

```

## Document Phases

The following table shows the document phase for a batch in the **batch\_dyn\_hdr** table.

Number	Phase
0	Batch Creation
1	Scanning
2	Image Verification
3	Rescan
4	Assembly
5	Indexing
6	Index Verification
7	Batch Totaling
8	Committal
9	Cataloging
10	Recommit

## Phase Status

The following table shows the phase status associated with a batch in the **batch\_dyn\_hdr** table.

Number	Phase
0	Not started
1	In progress
2	Interrupted
3	Completed
4	Unnecessary
5	Error occurred
6	Optional

## batch\_hdr

This is not a database table. It is an alias (view) used by MKF\_tool. For example, the following command retrieves all of the information from both the **batch\_dyn\_hdr** and **batch\_stat\_hdr** tables concerning this batch.

```
select batch_hdr * where batch_id = b7
```

## batch\_image

The **batch\_image** table keeps track of each image id number associated with a batch. It is updated during the scanning of a document and also during the act of scan verification. The record is cleared from this table upon the successful committal and cataloging of the batch.

Field Name	Field Description
image_index	Unique search key composed of batch_id and image_id keys - (group).
batch_id	Identification number of the batch. Searches on this column can be done in conjunction with the "image id" column.
image_id	Image id number.
doc_index	Unique search key composed of batch_id2, doc_id, and page_id keys - (group).
batch_id2	Same value as that of the "batch_id" column. However searches on this column can be done in conjunction with the columns "doc_id" and "page_id".
doc_id	This is the document id number of a document relative to a specific batch. For example, if a batch has multiple documents in it, the first document will be document id number 1.
page_id	Page number of a specific document.
image_length	The size of the image expressed in bytes.

Field Name	Field Description, Continued
image_type	Type of image: 0 = scanned image 1 = text 2 = fill in form 3 = mixed 9 = separator sheet 196 = pc form (In the case of a pc form, the columns batch_id2, doc_id, and page_id are not used.)
image_ver_stat	Verification status: 0 = unverified 1 = verified good 2 = failed verification
end_of_doc	A value of 1 indicates that this is the last page of the document. The value is 0 for all other pages.
index_len	The number of bytes in the "index_value" column.
index_value	This column is used in conjunction with images that contain text data. The indexing information is displayed in hexadecimal form.
text_len	The number of bytes in the text_data column.
text_data	Reserved

The following example shows a typical record from the **batch\_image** table:

batch_id.....8	image_id.....100171	batch_id2.....8
doc_id.....12	page_id.....1	image_length.....39640
image_type.....2	image_ver_stat.....0	end_of_doc.....0
index_len.....2	text_len.....8	
index_value	[0] ....0x3b00	

**batch\_ixdir**

The **batch\_ixdir** table keeps a general record for each indexing field that is associated with a specific batch (type of index, name of the index, is verification required, etc.). The record is cleared from this table upon the successful committal and cataloging of the batch.

Field Name	Field Description
ixdir_index	The batch and index ids. Composed of batch_id and index_id - (group).
batch_id	Identification number of the batch.
index_id	Index id number for the field. This number can be cross referenced to the Index database table doctaba. (For example, a value of 36 in this column equals column name A36 of the doctaba table.)
index_name	The name of the indexing field.
index_type	The type of index field. 49 = numeric 50 = string 52 = menu 56 = date
max_string_len	This field only applies if the index_type is 50 which is the maximum number of characters for a string index.
menu_name	The name of an indexing menu field.
verify_flag	0 = no index verify necessary 1 = index verification needed
required_flag	0 = optional 1 = index info must be entered
batch_tot_flag	0 = batch totaling unnecessary 1 = batch totaling needed
up_case_flag	0 = no auto uppercase conversion of string data 1 = Convert string data automatically to uppercase when cataloging.

Field Name	Field Description, Continued
source	How the fields will be indexed. 0 = unknown 1 = manually 2 = ocr 3 = aperture card
exp_total	Expected batch total number expressed in a 16 byte format.
act_total	Actual batch total number expressed in a 16 byte format.
mask	Template being used for a numeric or date index field.

The following example shows a typical record from the **batch\_ixdir** table:

batch_id.....9	index_id.....31	index_type.....50
max_string_len.....6	verify_flag.....1	required_flag.....0
batch_tot_flag.....0	up_case_flag.....1	source.....1
index_name....."I1"		
menu_name....."		
exp_total	[0] ....0x00000000000000000000000000000000	
act_total	[0] ....0x00000000000000000000000000000000	

**batch\_ixval**

The **batch\_ixval** table contains the actual indexing information that a given index field contains. Multiple records can exist for any given batch (one record per index field). The indexing information however is

stored in a hexadecimal format. The record is cleared from this table upon the successful committal and cataloging of the batch.

Field Name	Field Description
ixval_index	Unique search key composed of batch_id, doc_id, and index_id keys - (group).
batch_id	Identification number of the batch.
doc_id	The document id number.
index_id	Index id number from the Index database table doctaba.
index_type	The type of index field. 49 = numeric 50 = string 52 = menu 56 = date
index_len	The number of bytes in the index_value column.
index_value	The index information (in hex).
data_status	Reserved for use by OCR.
text_len	The number of bytes in the text_data column.
text_data	Reserved. If the value of text_len = 0, the field is not displayed.

The following example shows a typical record from the **batch\_ixval** table:

batch_id.....8	doc_id.....5	index_
id.....31		
index_type.....50	index_len.....4	data_
status.....0		
text_len.....0		
index_value	[0] ....0x646f7567	



**batch\_stat\_hdr**

A general record is created within the **batch\_stat\_hdr** table each time a batch is created. The record contains information such as batch id number, name of the document class associated with the batch, and the number of indexing fields to be completed. The record is cleared from the table upon successful committal and cataloging of the batch.

Field Name	Field Description
batch_id	Batch identification number.
queue_index	Non-unique search key composed of queue and qtime keys - (group).
queue	0 = not queued 1 = Uncommit queue 2 = queued for committal 3 = queue in progress
qtime	When the batch was queued for committal.
service_name	Organization/domain/object.
open_time	When the batch was opened.
open_user_id	Identification number of the operator that opened the batch.
open_flag	0 = batch not currently open 1 = batch open
create_time	When the batch was created.
create_user_id	Identification number of the user that created the batch.
tab_out_flag	Equals 1 if the operator can exit the window using the 'TAB' key.
doc_class_name	Document class name under which this batch was created.
num_indices	Number of index fields for the document.
num_reqd_indcs	Number of mandatory index fields for the document.
num_menus	Number of menu indexing fields for the document.
indexing_form	Name of the indexing form used with the document class.

Field Name	Field Description, Continued
access_restrct	Default doc security
cluster_space	Identification number of cluster operation that is involved. This feature is currently not used as denoted by the number 65535.
cluster_index	Cluster column number.
fam_id	Optical disk family number for this document. 0=clustering
wfl_queue_name	WorkFlo queue name for documents entering the system.
wfl_sys_name	WorkFlo system name for documents entering the system.
retent_disp	How to handle document once retention offset is reached. 0 = delete 1 = archive
retent_base	When to start the retention offset "clock" running. 0 = date of closing 1 = date of entry
retent_offset	Number of months until the document will be deleted or archived.
act_pages	Total number of pages in the batch.
act_docs	Total number of documents in the batch.
next_doc_id	Next doc id to assign (0 relative).
target_ims	Indicates the target FileNet Image Services system in a cross committal system. NULL = local
csum_state	Indicates the state of checksumming that is being used on this batch. 0 = unknown 1 = undetermined 2 = no checksumming 3 = checksumming 4 = resetting checksum
delete	0 = not marked for deletion 1 = delete fast batch 2 = delete normal batch

The following example shows a typical record from the **batch\_stat\_hdr** table:

```
batch_id.....8   queue.....0
qtime.....757619523
open_user_id.....11   open_flag.....0   create_user_
id.....21
tab_out_flag.....0   num_indices.....1   num_reqd_
indcs.....1
num_menus.....0   cluster_space.....0   cluster_
index.....0
fam_id.....2   retent_disp.....49   retent_
base.....49
retent_offset.....12   act_pages.....11   act_
docs.....9
next_doc_id.....13   csum_state.....2
open_time.....757786838 => 11/1/5 08:20:38
create_time.....757619523 => 11/1/3 09:52:03
doc_class_name....."test"
indexing_form....."F_IXFORM1"
access_restrct[0]....0x000000010000000100000001
```

**batch\_data**

The **batch\_data** table is the Batch services batch object data table. There is one entry in this table per object data.

Field Name	Field Description
data_index	Unique search key composed of batch_id, object_type, object_id, and sequence_id keys - (group).
batch_id	Batch identifier.

Field Name	Field Description, Continued
object_type	Object type.
object_id	Object identifier.
sequence_id	ID of this data row in a sequence.
data_length	Number of bytes in the buffer.
data_value	Value of data.

## batch\_folder

The **batch\_folder** table is the Batch services folder node table. There is one entry in this table for each folder node.

Field Name	Field Description
node_id	Folder node id.
relation	The parent and base node. Composed of parent_id and base_node_name - (group).
parent_id	Parent Identifier.
base_node_name	Base node name.
batch_id	Batch Identifier. 0= non-batch node. Batch id does not need to be unique.
last_update	Time stamp of creation or update of the node.
info_len	Length associated with the information.
info_data	Actual information.

## Cache Tables and Contents

Each table (and its contents) in the Transient database used by Cache services is discussed in this section; some of the tables contain examples of the contents. You can use the MKF\_tool to view these tables and contents directly.

### **csm\_caches**

The **csm\_caches** table contains a list of all of the logical cache id numbers. It is referenced before any interaction with a cache is performed.

Field Name	Field Description
cache_cache_id	System-assigned identification number for a specific type of cache.
cache_name_len	Number of characters in the cache's name.
cache_name	Name of the cache.
inuse_objects	Total number of objects in use.
inuse_sectors	Total number of sectors in use.
locked_objects	Total number of locked objects currently in use.
locked_sectors	Total number of locked sectors currently in use.

The following example shows a typical record from the **csn\_caches** table:

cache_cache_id	cache_name_len	cache_name
-----	-----	-----
5	11	"page_cache1"
6	16	"sys_print_cache1"
7	13	"fillin_cache1"
8	10	"bes_cache1"
9	16	"app_print_cache1"

## csn\_free\_space

The **csn\_free\_space** table keeps track of all of the empty sectors in the caches. Whenever the contents of the csn\_used\_space table are modified, this table is updated.

Field Name	Field Description
free_block	Unique search key composed of the size and group_start keys - (group).
size	The number of available contiguous sectors from a given start sector.
group_start	Same as "start".
start	The last three bytes indicate the starting sector number (relative to the partition), of available free space.

The following example shows a typical record from the **csn\_free\_space** table:

size	group_start	start
----	-----	-----
1	0x010001cf	0x010001cf
1	0x0100052f	0x0100052f
9	0x0100033c	0x0100033c
37	0x01000277	0x01000277
45	0x01000711	0x01000711
38626	0x0100091e	0x0100091e

**csn\_temp\_id**

The **csn\_temp\_id** table contains a list of temporary id numbers. A temporary id number is assigned to any object, except an image, that is being transferred into the cache.

Field Name	Field Description
temp_id	Unique search key. Temporary system assigned identification number for each object or page being transferred into the cache. Once the transfer is complete, the temporary id number is cleared.

The following example shows a typical record from the **csn\_temp\_id** table:

```
temp_id
-----
4160000005
4160000006
4160000007
4160000008
4160000009
4160000010
```

## csn\_used\_space

The **csn\_used\_space** table keeps track of every image, document, or object in the cache partitions. The contents of this table are referenced or updated during scanning, committal, retrieval, and printing.

Field Name	Field Description
object	Unique search key composed of the cache_id, ssn, obj_id, and page keys - (group).
cache_id	Number indicating the type of cache. The number can be cross referenced in the table "csn_caches".
ssn	System serial number where document (or image) was created.
obj_id	Image or document id number.
page	Page number of this object. Uncommitted images have page number 65335. Documents that are committed but not yet migrated to optical disk start with page number 0. Migrated documents start with page 1.
duration	Defined by the time in the 'read' field below added to the number specified in fn_edit, under Server Application services, Cache Duration, Refresh Cache Duration.



Field Name	Field Description, Continued
start	Location of an object in cache or first chunk. High byte is equal to the partition numbers and low 3 bytes is equal to the start sector number.
maxlen	Amount of space reserved for this image or page in the cache.
curlen	Current length. The size of this particular page (in bytes).
created	Date and time when the image (page), was scanned.
read	Date and time when page was last read. Only applies to cache_id 1 (page-cache). All other caches display a constant date of 1969/12/31.
temp_flag	Not used. 0 = not temporary 1 = temporary
sec_len	Length (in bytes) of the "sec_info"
sec_info	There are three security field for csm objects: read permission, write permission, and append/execute permission. The values in the sec_info identify the permission levels of each of the user groups. If there is no security for the object, these fields are set to 'ANYONE'.
refcnt	Reference count. The number of times this page has been accessed while in cache.
client_attr	Type of document, Image Form, Mixed etc.
seqnum	The seqnum field keeps track of how often an object has been opened. It is for internal IS use only.
numchunk	Number of contiguous chunks used to hold the object.
startary	The last three bytes of information indicate the starting sector number of the page relative to the partition.
lenary	Number of sectors used to hold the object.
csum	Check on the image. Value is no if there is no checksumming.

Field Name	Field Description, Continued
lastupdate	Date and time that the object was last updated.
incomplete	Replaces temp_flag.  NULL or 0 = completed object non-NULL = incomplete object

The following example shows a typical record from the **csn\_used\_space** table:

cache_id.....5	ssn.....100000	obj_id.....100090
page.....0	maxlen.....128	curlen.....128
temp_flag.....0	sec_len.....0	refcnt.....1
seqnum.....1	numchunk.....1	startary....0x01000043
lenary.....1		
created.....756518115	=> 93/12/21 15:55:15	
read.....0	=> 69/12/31 16:00:00	
sec_info	[0].....	
client_attr	[0]....0,2,0,0,0,0,0,0	

## Print tables and contents

### print\_docs

When multiple documents are to be printed from a single print request, a record is created in the **print\_docs** table. One record exists for each document to be printed. Data I/O with this table is primarily accomplished through the PRLI shared library.

Field Name	Field Description
print_doc	Unique search key composed of the request_id and doc_order keys - (group).
request_id	System-assigned request identification number.
doc_order	The order in which the selected documents will be printed (starting from 0 for the first document).
doc_id	The document id number of the selected document.
first_page	Beginning page number of a document.
last_page	Ending page number of a document.
ssn	System serial number if the column "svc_type" has a value of 2.
svc_type	The service that requested the print job. 1 = Document services 2 = Cache services
delete_after	This value is set by the client software. 0 = do not delete from cache 1 = delete after printing
svc_id	Service id of service which has data

## print\_options

The **print\_options** table keeps a record of the options chosen for each entry in the print\_requests table. Options include paper size, printer, header pages, etc. Data I/O with this table is primarily accomplished through the PRIL shared library.

Field Name	Field Description
request_id	Unique search key - system-assigned request identification number.
print_time	Non-unique search key - time at which page fetching will begin.
priority	Priority level of the print job. If no value is present, the default priority level of 4 is being used. The valid range is 0 through 9, with 9 having the highest priority.
paper_size	Paper size to be used. NULL = default (8.5 x 11) 0 = unknown                      1 = letter 2 = legal                          3 = B-size 4 = C-size                        5 = D-size 6 = E-size                        7 = A0 8 = A1                            9 = A2 10 = A3                          11 = A4 12 = A5                          13 = B4 14 = B5                          15 = 18x24 16 = top tray                    17 = bottom tray 18 = third tray                  19 = default 20 = half letter                21 = best available 22 = 10x14                      24 = executive (For FAX servers the value is always 11.)
collate	A value will only exist in this column if the printer has been configured to collate documents (through the client software).
staple	Not used.
two_sided	If no value is present, the print job is single-sided.

Field Name	Field Description, Continued
scaling	0 = normal (exact vert, clip horiz) 1 = clip both (Approx size, clipping) 2 = exact (exact fit) 3 = scale approx (Approx size, no clip) 4 = original size 5 = center (Not specified by user, automatic for screen prints) 6 = enhanced exact (not currently used)
orientation	NULL = default (does not appear) 1 = landscape 2 = portrait 3 = no rotation
annotations	0 = no annotations 1 = print annotations
request_header	0 = no header page 1 = print header page
doc_headers	If no value is present, document header pages will not be printed.  If a value of 1 exists, a header page containing the document id number will be printed.
form_name_len	The number of bytes in the form_name column.
form_name	Name of the form to be printed.
note_len	The number of bytes in the note column (max 42).
note	Message up to 42 characters long that will appear on the header page of the print job.
overlay	Not used.
phone_num_len	The number of bytes in the phone_num column.
phone_num	Fax machine phone number.
headline_len	The number of bytes in the headline column.
headline	Headline message to be sent to the fax as part of the print job. The message can be up to 98 characters long.

Field Name	Field Description, Continued
fax_mode	NULL = coarse 1 = fine 2 = 300 dpi. 3 = 400 dpi. 4 = filenet 5 = other
page_footnote	1 = page footnotes used
time_footnote	1 = time footnotes used
eject_tray	Paper output tray number
phone_ext_len	Length of the phone_ext field.
phone_ext	TBD
mem_len	TBD
memo	Contains a memo entered through the client software.
cover_doc	TBD
cover_ssn	TBD

## print\_svcs

The **print\_svcs** table is used to map Document and Cache services names to a 2 byte id. Print services uses IDs instead of names in the print\_requests and print\_docs tables, and the print\_svcs table is used to save the id to mapping. There is one record in the print\_svcs table for each document or cache service that Print services has communicated with.

Field Name	Field Description
svc_id	Unique search key - TBD
svc_type	1 = Document services 2 = Cache services

Field Name	Field Description, Continued
object_name	Object name of the service.
domain_name	Domain name of the service.
org_name	Organization name of the service.

The content of a typical **csm temp id** table record is shown below

```

svc_id.....0    svc_type.....1
object_name..."DocServer"
domain_name....."corona"
org_name....."FileNet"
-----
---
svc_id.....1    svc_type.....2
object_name....."sys_print_cache1"[for printing documents]
domain_name....."corona"
org_name....."FileNet"

```

## print\_reqopt

The **print\_reqopt** table is not actually a database table. It is an alias (view) used by MKF\_tool. For example, the following command retrieves all of the information from both the **print\_requests** and **print\_options** tables concerning this document.

```
select print regopt * where doc id = 100127
```

## print\_requests

The **print\_requests** table keeps a record of each outstanding print request. Data I/O with this table is primarily accomplished through the use of the PRLI shared library.

Upon the completion of a print job, the print request record is cleared within one minute from this table by the program PRI\_daemon. If a print request is cancelled by a user, PRI\_daemon waits ten minutes before actually clearing the record from this table.

Field Name	Field Description
request_id	Unique search key - system-assigned request identification number.
done_time	Time at which the request was completed, cancelled, or failed.
request_time	When the print request was placed into the queue.
request_length	If printing Text=size of print job (in bytes); Image=# of pages.
print_err	Reason for print job failure. fn_msg command decodes hex number.
pages_printed	Number of pages successfully printed.
notify_addr	Address of server to notify when print job completes. 0 = do not notify
fax_request	NULL = print request 1 = fax request
security	Document or text file security information.
doc_id	Document id number of document being printed. Number 4160000001, and up indicates text, not a document, is being printed.
first_page	Number of the first page that will be printed.
last_page	Number of the last page that will be printed.
ssn	System serial number if the svc_type column has a value of 2.
svc_type	The service that requested the print job. 1 = Document services 2 = Cache services
delete_after	0 = do not delete from cache 1 = delete after printing
svc_id	ID of service holding data to print
user_id	ID of user issuing print request



Field Name	Field Description, Continued
terminal_id	ID of terminal print request issued from
print_order	Unique search key composed of the printer_num, inverse_prio, and sequence_num keys - (group).
printer_num	Printer number assigned this print request
inverse_prio	Inverse priority (10-priority #) if job ready to print. Else, 10 if print job is suspended or is scheduled for printing at later time.
sequence_num	Sequence number of print request. ex. 5010
request_stat	0= unknown      3 = completed      6 = waiting 1 = queued      4 = failed      7 = fetching 2 = printing      5 = cancelled      8 = suspended
copies	Number of copies to print. NULL for default (1 copy)
notify_addr_len	Number of bytes in the notify_addr_str column.
notify_addr_str	Notification address for IPv6. Address of server to notify when print job completes.

# Requests tables and contents

## bkg\_request

The **bkg\_request** table keeps track of all background job requests. Optical disk to disk copies, optical disk imports, and optical disk annotation copies are considered background jobs.

Field Name	Field Description
job_number	Job number.
job_type	1 = Copy Documents      4 = Create Archive      7 = Migrate Documents 2 = Import Documents    5 = Find Open Docs     8 = Scan Surfaces 3 = Copy Annotations    6 = Import Archive     9 = Update Surfaces
prevjob	Previous job in queue.
jobqueueenum	Job number to use for messages.
surfid	ID of the current optical disk surface. 0 = multiple surfaces are required.
suspended	0 = running 1 = suspended 2 = queued
num_long_wds	Number of words in the array.
long_wd_ary	Data relative to task type.
pid	Process ID of background job.
start_time	The time that the background job was started.

## write\_request

The **write\_request** table keeps track of each document in the page cache that needs to be written to optical disk. This table is updated

during committal and then cleared after the successful transfer of a document to optical disk.

Field Name	Field Description
doc_id	Unique search key - document ID.
cluster_space	Which clustering application.
cluster_id	Cluster ID.
family_id	Primary family to write.
new_cluster	Whether this is the first document in cluster. 00 = not a new cluster 01 = new cluster
num_pages	Number of pages in document.
req_type	0 or NULL = document 1 = batch
commit_error	Error during committal. 0 = none
batch_name	Batch name. Prefix of "F_" indicates that the system software created the batch.

## Security database tables and contents

The Security database contains information about all users, groups, and devices on the system. The tables are used exclusively by the Security services.

This chapter discusses the following Security database tables:

- sec\_object
- sec\_system
- sec\_deleted
- sec\_groups
- sec\_functions
- sec\_funcmbr
- sec\_namemap
- sec\_dbinfo
- sec\_rm\_config
- sec\_map\_prin\_to\_dn
- sec\_ce\_dom\_to\_id
- sec\_map\_cn\_to\_prin

# Tables and contents

Each table (and its contents) in the Transient database used by the Security service is discussed in this section; some of the tables contain examples of the contents. Some of the information in these tables is encrypted. You can use MKF\_tool to view these tables and contents directly.

## sec\_object

This table contains one record for each object on the system (i.e., user, group, and device).

Field Name	Description
obj_class	The type of object: 1 = system      3 = user 2 = group      4 = device
obj_name	ASCII string containing object name
obj_id	Unique object ID
admin_class	A bitfield representation of the administrative class for the given object. Identifies the object as belonging to one or more of the following: supervisor, principal, account, group, audit, or "not administrator." If someone is "not administrator" then he or she is not any of the other administrator types. This field is not generally human-readable due to the way it is stored.
dev_class	The type of device: 1= terminal 2 = printer 5= fax NULL=not a device
epasswd	Encrypted password
prim_group	Primary group ID
admin_group	Administrative group ID
language	Native user language

Field Name	Description, Continued								
dev_security	<p>If the object is a user, this indicates if the user is subject to device ("terminal") security.</p> <p>If this is set, then a user may be restricted from accessing certain documents depending on what terminal he/she is using to do so. If the object is a device, this indicates whether the device is subject to device security, which means that it has different security than other terminals from which users can access documents.</p>								
creation time	Database creation date								
from_min	Login restriction time - start time minutes (0-59)								
to_min	Login restriction time - end time minutes (0-59)								
from_hour	Login restriction time - start time hours (0-23 military)								
to_hour	Login restriction time - end time hours (0-23 military)								
from_dweek	Day of week (0-6: 0 = Sunday)								
to_dweek	Day of week (0-6: 0 = Sunday)								
exp_date	UNIX time for account termination								
succ_log_time	Last successful login time								
success_where	Last successful login location								
fail_log_time	Last failed login time								
failed_where	Last failed login location								
failed_err	Last failed login error cause								
nbr_logons	Total number of logons								
log_bits	<p>Bits indicating logging options:</p> <table> <tr> <td>log successful logons</td> <td>1 (bit value 1)</td> </tr> <tr> <td>log failed logons</td> <td>01 (bit value 2)</td> </tr> <tr> <td>log security changes</td> <td>001 (bit value 3)</td> </tr> <tr> <td>log all</td> <td>111</td> </tr> </table>	log successful logons	1 (bit value 1)	log failed logons	01 (bit value 2)	log security changes	001 (bit value 3)	log all	111
log successful logons	1 (bit value 1)								
log failed logons	01 (bit value 2)								
log security changes	001 (bit value 3)								
log all	111								
pwd_exp_time	Calculated UNIX expiration time								
pwd_fail_time	Time of password failure								

Field Name	Description, Continued
pwd_fail_atts	Number of failed password attempts
sys_override	0 = use system, 1 = use users
max_session	Number of allowed nbr sessions
comments	General comments
sess_group	Session group
dn_len	The number of bytes in the dn column.
dn	Distinguished name
dn_hash	Hash key of the distinguished name.
concurrent_hwm	High-water mark for concurrent logons
pwd_never_expire	Password never expire flag 0 = off 1 = on
hash_type	0 = NULL column 1 = SEC_HASH_TYPE_PED_SHA1 2 = SEC_HASH_TYPE_FIPS_SHA1
hashpasswd_len	The number of bytes in the hashpasswd column.
hashpasswd	Hash key of the password.

## sec\_system

The **sec\_system** table contains the system record for the Security service.

Field Name	Description
system_id	Record which equals unique key
ceiling_id	Number to track assigned user ids
ceiling_snbr	Value of last session number assigned

Field Name	Description, Continued
ceiling_fid	Value of next function number
ceiling_lan	Value of current language ID
dev_security	0 = off, 1 = on
no_func_def_ok	0 = not ok, 1 = ok
update_time	Stamp of time db last updated
from_min	Start time in minutes (0-59)
to_min	End time in minutes (0-59)
from_hour	Start time in hours (0-23: military)
to_hour	End time in hours (0-23: military)
from_dweek	Day of week (0-6: 0 = Sunday)
to_dweek	Day of week (0-6: 0 = Sunday)
max_sessions	The maximum number of current logons per user. This ensures that one user cannot use up all the SLUs for a given server.
highwater	Highest number ever logged on. This field is currently not used.
pwd_spec_char	Special character required (0 = no, 1 = yes)
pwd_min_len	Maximum password length (0 = minimum, 8 = maximum)
pwd_rnwl_days	Password renewal from last change (0 = not set, 1 = minimum, 365 = maximum)
pwd_grace	Time (in days) until lack of password change causes account suspension (0 = not set, 1 = minimum, 90 = maximum)
pwd_attempts	Used to identify nbr of password attempts until account is disabled (0 = not set, 1 = minimum, 100 = maximum)
pwd_fail_mins	0 = not set, 1 = minimum, 100 = maximum
log_bits	Bits indicating logging options: log successful logons      1 (bit value 1) log failed logons          01 (bit value 2) log security changes       001 (bit value 3) log all                      111



Field Name	Description, Continued
language	ASCII value indicating language
db_level	DB level of Security database
allow_over	Override system definitions (0 = no, 1 = yes)
ceiling_db_id	Pseudo_ID for DB logon
annot_security	Annotation security (0 = no, 1 = yes)
pwd_chg_upon_reset	Password change upon reset (0 = no, 1 = yes)
ext_pwd_validation	External vendor-acquired password validation (0 = no, 1 = yes)
extensible_auth	Extensible authentication (0 = no, 1 = yes)
fallback_auth	Fallback authentication (0 = no, 1 = yes)

## sec\_deleted

The **sec\_deleted** table is not currently used. It was designed to contain one record for each deleted object on the system.

Field Name	Description
obj_id	Deleted object ID
obj_name	ASCII string containing object, domain, and organization
obj_class	Object class
dev_class	Deleted device class
del_time	Deletion time

## sec\_groups

The **sec\_groups** table contains one record for each direct membership occurrence.

Field Name	Description
grp_group	The identifier for this group, containing group_id and member_id.
group_id	The numeric identifier (Object ID) for the group.
member_id	The numeric identifier (Object ID) for the member of the group.
grp_member	A member of the group specified in grp_group, containing group_id2. Does not display.
member_id2	Same as member_id (MKF requirement for multiple key setup).
group_id2	Same as group_id (MKF requirement for multiple key setup). Does not display.
member_class	Displays what kind of member it is: 1=admin 2=the group entry (specifying the same group) 3=user of group

## sec\_functions

The **sec\_functions** table contains the name and ID of each function that is known to Security services. Certain functions, such as “read from COLD,” “write to COLD,” and some user-defined functions can have function security that restricts access to these functions.

Field Name	Description
fun_name	ASCII string containing object, domain, and organization.
fun_id	A unique numeric ID of the function name specified in the fun_name field.

## sec\_funcmbr

The **sec\_funcmbr** table is very similar to the **sec\_groups** table. It contains one record for each “function\_member.” Each function in the table can have members, just like a group.

Field Name	Description
fun_owner	This is a field grouping in MKF that indicates a membership. Contains the same value as the fun_member field. Does not display.
fun_id	Indicates the MKF ID of the function from the functions table. Contains the same value as the fun_id2 field.
member_id	Indicates the MKF object ID of the member of the function. Contains the same value as the member_id2 field.
fun_member	This is a field grouping in MKF that indicates a membership. Contains the same value as the fun_owner field. Does not display.
member_id2	Indicates the MKF object ID of the member of the function. Contains the same value as the member_id field.
fun_id2	Indicates the MKF ID of the function from the functions table. Contains the same value as the fun_id field.
member_class	Indicates the object class of the member: 0 = anyone 1 = admin 2 = group 3 = user

## sec\_namemap

The **sec\_namemap** table and the **sec\_dbinfo** table are used to map FileNet users onto native database logons (for Oracle, DB2, and SQL Server). One of these is created for each mapping made to the DB

logon. For example, FileNet user *<name1>FN* is the same as Oracle user *<name1>ORA*.

Field Name	Description
filenet_map	Field grouping in MKF that indicates a mapping. Contains the same value as the db_map field. Does not display.
obj_id	Indicates the object id of the FileNet user that is to be mapped to a database user. Contains the same value as the obj_id2 field.
db_pseudo_id	An index of a row in the sec_dbinfo table indicating the database user that this FileNet user is mapped to.
db_map	Field grouping in MKF that indicates a mapping. Contains the same value as the filenet_map field. Does not display.
db_pseudo_id2	An index of a row in the sec_dbinfo table indicating the database user that this FileNet Image Services user is mapped to.
obj_id2	Indicates the object id of the FileNet Image Services user that is to be mapped to a database user. Contains the same value as the obj_id field.

## sec\_dbinfo

The **sec\_dbinfo** table contains a record for each db logon added by the Administrator.

Field Name	Description
db_name	Native DB user name
db_pseudo_id	Native DB pseudo ID
db_pwd	Native DB encrypted password
db_creator_id	ID of creator

## sec\_rm\_config

This table contains a single record that defines the system-level Record Management configuration.

Field Name	Description
key_id	A numeric identifier that defines the security level. The current support is for system-level security only.
declare_level	A numeric identifier that sets the declaration level (i.e., lockdown mode). Level 1 = READ-ONLY (default) Level 2 = No Write (only new annotations can be created) Level 3 = No Change to document security (No op)
rm_group_id	The object ID of the configured “Record Management” group
purge_level	Not used.
scrub_level	Not used.
log_level	A numeric identifier that defines the “log level” mode. Level 0 = Minimal (errors only) Level 1 = Verbose
lock_sysadming	Not used.

## sec\_map\_prin\_to\_dn

The **sec\_map\_prin\_to\_dn** table holds mappings of FileNet Image Services objects to their LDAP distinguished names (dn) for exporting FileNet Image Services annotations to Content Engine in a CFS-IS environment.

Field Name	Description
id_to_dn	Unique search key composed of obj_ud and ce_dom_id keys - (group).
obj_id	Object id from the sec_object table
ce_dom_id	Links to the ce_dom_to_id table

Field Name	Description, Continued
dn_to_id	Unique search key composed of dn_hash and ce_dom_id2 keys - (group).
dn_hash	Distinguished name hash key
ce_dom_id2	Links to ce_dom_to_id table
dn_len	Distinguished name string length
dn	Actual distinguished name

### sec\_ce\_dom\_to\_id

The **sec\_ce\_dom\_to\_id** internal table contains all unique Content Engine GUIDs.

Field Name	Description
ce_dom_GUID	Unique GUID key for a Content Engine domain
ce_dom_id	Independent internal key number

### sec\_map\_cn\_to\_prin

The **sec\_map\_cn\_to\_prin** table maps LDAP common names to FileNet Image Services Principals (Object Names) for Extensible User Authentication.

Field Name	Description
obj_id	Object id from the sec_object table
ldap_cn	LDAP common name
prin_to_cn	Unique search key composed of obj_name and ldap_cn2 keys - (group).
obj_name	Object name from sec_object table
ldap_cn2	Same as ldap_cn (MKF requirement for multiple key setup). Does not display.

# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Corporation  
J74/G4  
555 Bailey Avenue  
San Jose, CA 95141  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
1623-14, Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.



Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
J46A/G4  
555 Bailey Avenue  
San Jose, CA 95141-1003  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims

related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

## Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names might be trademarks of IBM or other companies.

## U.S. Patents Disclosure

This product incorporates technology covered by one or more of the following patents: U.S. Patent Numbers: 6,094,505; 5,768,416; 5,625,465; 5,369,508; 5,258,855.

## A

- application program
  - request handler 73
  - stub 73
- application programs 73
- Application server 23

## C

- Combined server 21
- configuration database
  - databases 43
- configuration files
  - generated files 50
  - supplied 46

## D

- database tables
  - annot\_log 103
  - annotations 104
  - batch\_ctl 126
  - batch\_data 139
  - batch\_doc 127
  - batch\_dyn\_hdr 128
  - batch\_folder 140
  - batch\_hdr 132
  - batch\_image 132
  - batch\_ixdir 134
  - batch\_ixval 135
  - batch\_stat\_hdr 137
  - bkg\_request 154
  - cluster\_map 105

- csm\_caches 141
- csm\_free\_space 142
- csm\_temp\_id 143
- csm\_used\_space 144
- dbinfo 164, 165, 166
- docs 106
- family\_disk 110
- family\_locator 112
- lib\_surfaces 113
- od\_stats 114
- print\_docs 147
- print\_options 148
- print\_reqopt 151
- print\_requests 151
- print\_svcs 150
- remote\_family 115
- scalar\_numbers 116
- sec\_deleted 161
- sec\_funcmbr 163
- sec\_functions 162
- sec\_groups 162
- sec\_namemap 163
- sec\_object 157
- sec\_system 159
- surf\_dyn\_info 116
- surf\_info 119
- surf\_locator 119
- surf\_stat\_info 121, 123
- write\_request 154
- databases

- Permanent database tables 102
- Security database tables 156
- Transient database tables 125
- descriptions
  - application programs 73
  - Permanent database tables 102
  - Security database tables 156
  - shared libraries 88
  - subsystems 35
  - Transient database tables 125
- directory contents 28
- directory structure 27
- Dual server configuration 22

**I**

- IS directories
  - contents 28
  - structure 27

**P**

- Permanent database tables
  - annot\_log 103
  - annotations 104
  - cluster\_map 105
  - docs 106
  - family\_disk 110
  - family\_locator 112
  - lib\_surfaces 113
  - od\_stats 114
  - remote\_family 115
  - scalar\_numbers 116
  - surf\_dyn\_info 116
  - surf\_info 119
  - surf\_locator 119
  - surf\_stat\_info 121, 123

**R**

- Remote Entry server 25
- request handler 73
- request handlers 61

**S**

- Security database tables
  - sec\_ce\_dom\_to\_id 166
  - sec\_dbinfo 164
  - sec\_deleted 161
  - sec\_funcmbr 163
  - sec\_functions 162
  - sec\_groups 162
  - sec\_map\_cn\_to\_prin 166
  - sec\_map\_prin\_to\_dn 165
  - sec\_namemap 163
  - sec\_object 157
  - sec\_rm\_config 165
  - sec\_system 159
- server configurations 21
  - Application server 23
  - Combined server 21
  - Dual server 22
  - Entry server 25
- shared libraries
  - descriptions 88
  - locations 87
- stub 73
- subsystem descriptions 35

**T**

- Transient database tables
  - batch\_ctl 126
  - batch\_data 139
  - batch\_doc 127

batch\_dyn\_hdr 128  
batch\_folder 140  
batch\_hdr 132  
batch\_image 132  
batch\_ixdir 134  
batch\_ixval 135  
batch\_stat\_hdr 137  
bkg\_request 154  
csm\_caches 141  
csm\_free\_space 142  
csm\_temp\_id 143  
csm\_used\_space 144  
print\_docs 147  
print\_options 148  
print\_reqopt 151  
print\_requests 151  
print\_svcs 150  
write\_request 154

---





Product Number: 5724-R95

Printed in USA

SC19-3325-00

