IBM FileNet Image Services
4.1

MKF Database Conversion Procedure (Permanent and Transient Databases)

SC19-2955-00

IBM FileNet Image Services
4.1

MKF Database Conversion Procedure (Permanent and Transient Databases)

SC19-2955-00

Note

Before using this information and the product it supports, read the information in "Notices" on page 33.

This edition applies to version 4.1 of IBM FileNet Image Services (product number 5724-R95) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

MKF Database Conversion Procedure 7

MKF Database Conversion Procedure

Introduction

Several years ago, IBM® FileNet® Image Services developed an expanded MKF database format to accommodate the needs of those Image Services systems that are approaching the addressing limit of the original MKF database format. Image Services 4.1 and higher releases support both the original and a new optional MKF database format. This expanded MKF database format is available for those clients who anticipate having an MKF database larger than 16 GB (gigabytes).

Note All fresh installations of Image Services since IS 4.0 use the expanded MKF format.

In the original format, a single MKF database has database pages that are 1 KB with an addressing limit of 16 GB of disk space for the data files. In addition, 16 GB of disk space was required for the recovery logs. In the past, data files in the original format could be no larger than 2 GB – 1 KB. The MKF addressing limit of 16 GB imposed a maximum of eight 2 GB data files.

The individual data and recovery log files are no longer limited to 2 GB each. Individual files can be up to 2 to the 63 bytes, or the maximum size supported by the platform, whichever is smaller.

Database pages in the expanded MKF database format are 4 KB, 8 KB, or 16 KB. The expanded format has an addressing limit of 7E+22 bytes (for a page size of 4 KB, and larger than that for larger page sizes). That limit is in excess of 70,000,000,000,000 GB. Also, each data file can be up to 9E+18 bytes (9,000,000,000 GB) in size, or the maximum file size supported by the operating system, whichever is smaller.

In the expanded format, you can use more than eight data and recovery log files.

Document revision history

IS version	Date	Comment
4.1	Feb. 2010	Removed instructions for updating the MKF Security database, and added references to the separate document, *Updating the MKF Security Database*.
4.1	June 2007	Initial release.

Accessing IBM FileNet documentation

To access documentation for IBM FileNet products:

1. Navigate to the Image Services support overview page (**www.ibm.com/support/entry/portal/Overview/Software/Information_Management/FileNet_Image_Services**).
2. Select **Documentation** from the "Choose your task" pane.
3. Under "Product documentation", select the type of documentation that you want to view.

IBM FileNet education

IBM provides various forms of education. Please visit the IBM Information Management support page at (**www.ibm.com/software/data/support**).

Feedback

We value your opinion, experience, and use of our products. Please help us improve our products by providing feedback or by completing a consumability survey.

Documentation feedback

Send comments on this publication or other IBM FileNet Image Services documentation by e-mail to **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 help topic title, a chapter and section title, a table number, or a page number).

Product consumability feedback

Help us identify product enhancements by taking a Consumability Survey (**http://www-306.ibm.com/software/data/info/consumability-survey/**). The results of this comprehensive survey are used by product development teams when planning future releases. Although we are especially interested in survey responses regarding the most recent product releases, we welcome your feedback on any of our products.

The survey takes approximately 30 minutes to complete and must be completed in a single session; there is no option to save a partially completed response.

Maximum number of documents

Based on an average document size of about 50 bytes, the original MKF database format could store information for a maximum of 300 to 343 million documents in the permanent database. In contrast, based on the same document size, the expanded format can theoretically store 1.4E+21 documents or more. The FileNet system uses 32 bit unsigned integers for document numbers.

Thus, the actual number of documents the system can store in the expanded format is limited by the maximum FileNet document number that can be represented, currently 4 billion.

Note Even though the maximum document number is about 4 billion, the FileNet system cannot store a full 4 billion documents, because not all of the 4 billion integers are available for use as document numbers.

The FileNet system assigns the document and every page of the document one of the 4 billion integers available. If every document were one page, at most 2 billion documents could be stored on one FileNet system.

Storing the maximum of 2 billion documents in the permanent database would require roughly 120 GB of disk for the permanent database data files.

The disk addressing ability for the expanded MKF format is *12 to 13 orders of magnitude* greater than that. So it is, in effect, unlimited, because other limits will come into play well before the new format's addressing limit is reached.

The benefits of MKF

Image Services 4.1 does not require that you to convert earlier MKF databases. The MKF subsystem can be configured to use either the original or the expanded formats.

The original MKF database format remains unchanged. However, IS 4.1 and higher uses 63 bit file I/O. That means that individual MKF data files and recovery log files are no longer limited to 2 to the 31 bytes (2 GB) - they can be up to 2 to the 63 bytes (9E+18 bytes), or the maximum size that the operating system supports.

The expanded MKF database format offers the following advantages over the original format:

- allows a larger amount of disk space to be addressed
- allows more concurrency for database updates and fewer deadlocks

The following is a list of the tasks you must accomplish to convert from the original format to the expanded format (or vice versa).

- Run MKF_export
- Run fn_edit, fn_build –a, and the appropriate platform dependent tool to update the logical volumes
- Run MKF_ddl –initialize
- Run MKF_import.

Important Make backups of the database before and after the conversion.

Selecting the MKF database format

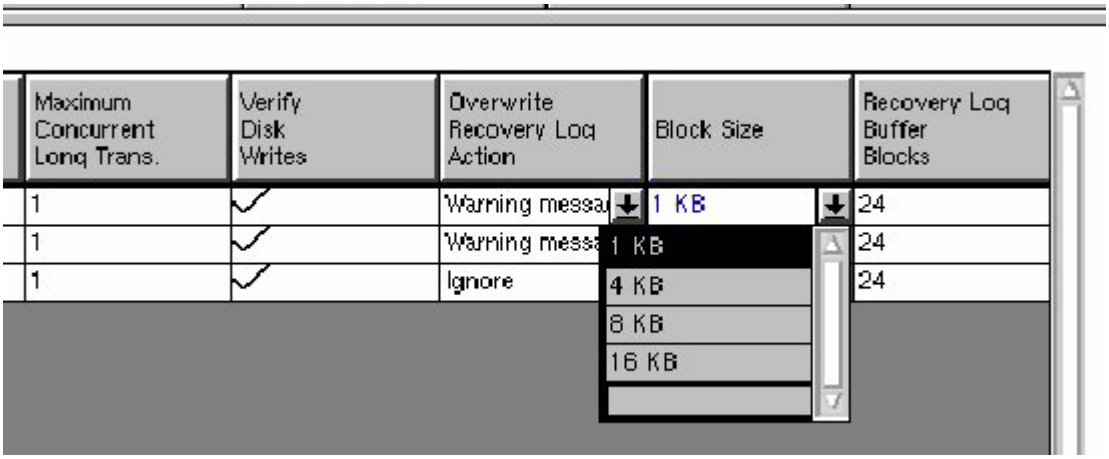
Use the FileNet System Configuration Editor, **fn_edit**, to create a new MKF database. For more information on using fn_edit, see the ***Image Services System Configuration Overview***.

To create a database in the original format

1. Start **fn_edit**.
2. Select the MKF databases tab. In the Block Size column, select a page size of 1 KB.

To create a database in the expanded format

1. Start **fn_edit**.
2. Select the MKF Databases tab. In the Block Size column, select a page size of 4 KB, 8 KB, or 16 KB.



Converting the MKF database

You might decide to convert your MKF database for the following reasons:

- You want to benefit from increased concurrency of database updates
- You expect the database size will eventually approach or exceed 16 GB

Deciding which MKF databases to convert

Use the procedures in this document to convert the following databases:

- permanent database
- transient database

Important To convert the MKF security database, see *Updating the MKF Security Database*.

Do not convert the Network Clearinghouse (NCH) database. The NCH database must be up and running for remote proce-

dure calls to work, and since it must be up and running even to make backups with EBR, converting the NCH database is a special case and is not covered by this document.

Planning the conversion

The primary reason for converting an MKF database to the new format is that more than 16 GB of disk space is needed. That is a large amount of data to export, import, and back up twice. You need to estimate how much time it will take. If you backup to tape, and export and import using tape, it will take about 36 hours to convert a 16 GB database. If you backup to disk and export and import using disk files, you might be able to convert the data in a single eight-hour shift.

Output file sizes

Both EBR backup files and MKF_export output files are compressed, and contain error correction data to increase robustness and performance.

The EBR backup file for an MKF database is typically about one half the size of the portion of the database in use; that is, the amount of data up to the high water mark.

Tip You can determine the high water mark of an MKF database by running **Xapex** > System Monitor > Storage > Databases, or by running the command-line utility **MKF_stats**.

The MKF_export output file is typically about one third the size of the portion of the MKF database in use.

If you are converting because you need to make the database larger, decide how much larger you are going to make the database. Make sure you have enough disk for the database and, if you are backing up to disk and exporting to disk, for two backups plus the export data file.

If you are backing up to tape or exporting to tape, compute the number of tape cartridges needed. EBR backup tape cartridges must be prelabeled, but tape cartridges holding MKF export data are not prelabeled.

Magnetic disk considerations

To avoid a significant degradation in performance when configuring MKF and relational databases, the data files should be on one set of disk arms, and the recovery logs should be on another set of disk arms. Otherwise, it may be impossible, after a disk crash, to restore the database and then roll the database forward using the disk-resident recovery logs.

Note Some customer databases are configured incorrectly, in that the recovery log and the data files are on the same disk arm or set of disk arms. Check for this mistake in your current configuration. If it exists, a database conversion using two sets of disk arms can correct this problem.

When using RAID, it is best to have the database data files on one RAID subsystem, and the recovery logs on another.

Magnetic tape considerations

Backing up and exporting to magnetic tape is more than an order of magnitude slower than backing up or exporting to magnetic disk files. Allot additional time for the conversion if you plan to use magnetic tape.

You can reduce the backup time if you have multiple tape drives. Use multiple EBR threads to back up your tape drives concurrently.

The MKF_export and MKF_import utilities use a single tape drive, and cannot benefit from the use of multiple tape drives. However, in IS 4.1, MKF_export and MKF_import can handle tape overflow reels.

For most tape drives, EBR, MKF_export, and MKF_import should all drive the tape at the drive's maximum throughput. Using the tape drive's hardware compression will not increase the throughput or capacity of the tape drive because the data sent to the tape drive by EBR, MKF_export, and MKF_import is already compressed. Tape cartridge capacity and throughput are quoted both with and without compression turned on. Typically, the figures quoted for compression- turned- on are twice as large as the no-compression figures for cartridge capacity and tape drive throughput. When estimating your conversion times, use the figures for no-hardware-compression.

Conversion example

The following example is based on a database conversion which was run in the software lab. We converted an MKF permanent database containing 300,000,000 rows in the docs table and used an HP-UX system with four CPUs that were dedicated solely to the conversion effort.

The data files of the permanent database and the recovery log were on separate sets of disk arms which resulted in maximum performance for normal database operation as well as for the conversion.

The database was backed up to a disk file. For maximum backup performance, the data files of the permanent database and the EBR backup file were on separate sets of disk arms.

The disk arms for the EBR backup file and the export data file were the same set of disk arms used for the recovery log. This did not affect the performance of the backup, the export, or the import with the "init" option, because the recovery log was not used when these utilities were running. If the "add" option had been used for the import, then it would have been important for performance to have the export data file on a separate set of disk arms from both the database files and the recovery log files.

This is a best-case performance conversion example for a system containing 300,000,000 documents in the permanent database. The major steps of the conversion, and the time for each step, were:

1. EBR backup: 1,600 seconds (26:40)
2. MKF_export: 2,487 seconds (46:27)
3. MKF_ddl -initialize: 2,255 seconds (37:30)
4. MKF_import init (import part only): 2,487 seconds (46:27)
5. MKF_fixup (fixup part of MKF_import init): 11,500 seconds (3:11:40)
6. EBR backup: 1,600 seconds (26:40)

The total was 23,739 seconds, or 6:05:29.

These times are somewhat better than what can be expected for a customer system, because the allocated size of the new database for this example was exactly the same as the allocated size of the original database. In an actual conversion, the database is probably being converted because it is too small and the size of the new database will be significantly larger than the size of the original database. This will result in larger times for steps 3, 5, and 6.

This example used the "init" option for the MKF_import run. If you use the "add" option to MKF_import, normal MKF inserts will be done by MKF_import, eliminating the MKF_fixup step. That would have increased the total time for steps 3 and 4 above to 2.3 to 3.5 twentyfour-hour days.

Why, then, would you want to consider using the "add" option? The resulting performance of the final database would tend to be better for the "add" option, because normal inserts tend to group the B-tree pages and the corresponding data pages

together. The "init" options tend to maximally separate the B-tree pages from the data pages that they reference.

No deletions were performed on the docs table in the above example, and no optical disks were imported. If you need to do either of these things to your permanent database, step 5 will take longer.

If anything else is running during the conversion, the time will be increased.

The throughput of the conversion example was limited primarily by the bandwith of magnetic disk I/O. For this reason, even if the system was a single-processor system instead of a multi-processor system, the conversion time would not be much greater.

Converting MKF databases on multiple servers

1 Stop all the servers in the proper order.

2 Convert the MKF databases on the root server.

3 Start the FileNet software on the root server.

Leave the FileNet software running on the root server while you convert the MKF databases on the Storage Library and Application servers.

Conversion procedure

Logging on to the system

WIN

On Windows servers, log on to Windows as the FileNet software user, such as **fnsw**, and open a command prompt window. To do this, click Start > Run, and enter **cmd** in the dialog box.

UNIX

On UNIX® servers, log on as a user with **root** privileges in a shell window. You must be **root** user to run EBR and the platform dependent tools that manage logical volumes. For most of the other steps, however, being **fnsw** user is adequate. If you run as **root** user, and root is a member of the fn_admin group, you should have no problems. If **root** is not a member of the fn_admin group, some steps will have to be done as **root**, and some as **fnsw**.

Backing up the MKF database

1. In a shell window, put the FileNet software in backup mode by entering:

   initfnsw backup

2. Make a full offline backup of the MKF database to be converted. If anything goes wrong during the upgrade, or if you run out of time, restoring this backup will enable you to get back to where you were.

   You can run the FileNet Enterprise Backup Restore program, EBR, in a shell window to perform the backup. You can back up to either tape or disk.

Note Backing up the database to disk is an order of magnitude faster than backing up to tape.

Stopping the FileNet software

1. If Xtaskman is running, stop it.

2. In a shell window, enter the following command. (This will not kill Xtaskman, so make sure Xtaskman is not running.):

UNIX

```
initfnsw -y stop
killfnsw -DAy
```

WIN

```
initfnsw -y stop
killfnsw -Dy
```

Note If you have trouble running **killfnsw**, and a warning message indicates that the utility cannot be run because there is a guard file, run **killfnsw -r** to remove the guard file, and start again. This warning only happens when **initfnsw** fails to complete properly. The guard file ensures that only one copy of **initfnsw** can run at a time.

3. Ensure that no FileNet processes are running by entering **ps -ef | grep fnsw** on UNIX, or by running the Windows task manager on Windows and looking at all the processes.

   It is important that all FileNet processes are terminated, including the TM_daemon process and Xtaskman.

Starting the MKF database

MKF_export requires the database to be in normal mode when it starts. To accomplish this, run MKF_startup to start up the database.

Note If you forget to start up the database before running MKF_export, you will notice that MKF_export will not progress, displaying instead, that zero rows have been exported. In this

case, open another shell window and run MKF_startup. MKF_export will then start exporting data.

Start the MKF database by running the MKF_startup command:

UNIX

UNIX example:

MKF_startup /fnsw/dev/1/permanent_db0

WIN

Windows example: (Replace **d** with the correct drive letter on your system.)

MKF_startup d:\fnsw\dev\1\permanent_db0

Exporting the data

When MKF_export runs, it periodically displays:

- the number of seconds that have elapsed since it started
- an estimate of how many seconds remain to completion
- how many database rows have been exported
- the percentage of the database disk it has read so far.

MKF_export requires the database to be in normal mode when it starts. When MKF_export runs, it shuts down the database. Thus, the FileNet software should not be running when MKF_export is running. After MKF_export completes, it starts up the database.

Note If MKF_export is killed or terminates abnormally, it might be necessary to run MKF_startup to start the database.

MKF_export to disk typically exports over 100,000 rows a second when exporting to a disk file. (This assumes that the database data files and the exported data file are on separate

disk arms.) When exporting to tape, MKF_export is limited by tape speed. The size of the exported data is typically about one third the size of the portion of the database in use (that is, the data up to the database high water mark).

Export to multiple cartridges

If the data is exported to tape and the tape cartridge fills up, MKF_ export closes and rewinds the tape. The utility then asks the operator to mount another tape. The utility continues to request as many overflow cartridges as needed to hold all the data. If multiple tape cartridges are needed, it is important to write the number of the cartridge (such as 1, 2, and so on) on the cartridge by hand (using a pen or marker, for example). Later, when you run MKF_import, the cartridges must be read in the order they were written.

Note If exporting to tape, be careful to mount only tape cartridges that contain no useful data. MKF_export writes to the tape without checking the existing content, destroying any data that was already on the tape.

Tape drive incompatibility

The supported platforms for this conversion are AIX, HP-UX, Solaris, and Windows Server.

You can export to disk or tape from a host running any supported platform, and import the data to the same or any other host running any of the supported platforms. However, FileNet can not guarantee that a tape written by one operating system (OS) on one vendor's hardware will be readable by another OS using the same type of tape drive. Sometimes tapes written by one OS cannot be read by another OS. Sometimes tape drive calibration is off a little, and a tape written by one drive cannot be read by another drive of the same type, even of the same system. Because of this potential tape drive incompatibility, if

you plan to export on one system and import on a different system, we recommended that you write a small file to tape and see if the other system can read the tape. If the drives on the two systems are not compatible, consider exporting to disk and using **ftp** or some other utility to export the data.

To begin exporting data, in a shell window, run MKF_export on the target database. For example:

UNIX

```
MKF_export  /fnsw/dev/1/permanent_db0  out=/tmp/permanent.dat

MKF_export  /fnsw/dev/1/permanent_db0  out=tape  type=8mm \
    device=/dev/rmt0
```

WIN

```
MKF_export  d:\fnsw\dev\1\permanent_db0  out=e:\tmp\perm.dat

MKF_export  d:\fnsw\dev\1\permanent_db0  out=tape  type=8mm
    device=TAPE0
```

Defining the new MKF database

1. In a shell window, start **fn_edit** to make the configuration changes to the MKF database. Enter the following command:

   fn_edit &

2. On the MKF databases tab, change the block size of the database being updated to 4 KB, 8 KB, or 16 KB. For large databases, 16 KB will minimize the height of the B-tree and prevent degraded performance. Performance gains may also be mitigated because:

   - 16 times as much data will be copied in memory and written to the recovery log.
   - recovery log will tend to require almost 16 times as much disk as the original format.

3. In the datasets tab, specify the size of the database and the recovery log.

For MKF databases in the expanded format, after rounding up to the next multiple of 4, the base data file must have at least 12 MB, 28 MB, and 48 MB for 4 KB, 8 KB, and 16 KB pages, respectively, where 1 MB is 1,024 times 1,024 bytes. The expanded format preallocates substantially more space than the original format.

The recovery log must have at least 64 MB for the new format. The first recovery log file should have at least 8 MB. The recommended minimum backup frequency is a full backup at least weekly, and an interval backup at least once a day. If rollforward is necessary after restoring a backup, then the recovery log should be big enough to hold two days worth of recovery log data, as that allows for missing one daily backup.

4 To change the buffer pool size for the database and the maximum number of concurrent transactions, start **fn_edit**. Find buffer pool size (in KB) on the Performance Tuning tab under Server memory. Note that the number of buffers is the buffer pool size in KB divided by the page size in KB. Find the maximum number of concurrent transactions under the MKF Databases tab.

You need enough buffers so that each concurrent transaction has a reasonable number of buffers to work with. For example, if you had 150 buffers for an MKF database in the original format, the buffer pool size was 150 KB. If you had 3 concurrent transactions, each concurrent transaction had 50 buffers as its share. If you change the page size to 16 KB, you would only have 9 buffers (150 KB divided by 16 KB per buffer), which is not enough for even one transaction, let alone three concurrent transactions. In this case, increase the buffer pool size substantially, perhaps even by a factor equal to the number of KB per page (16 for this example).

In general, for page sizes larger than 1 KB, you should increase the buffer pool size. The transient database, however, might be an exception. The buffer pool size for the transient

database can be very large, and might already be at the maximum size you are willing to allocate. In this case, leave the buffer pool size of the transient database unchanged. Since the buffer pool was already very large, each concurrent transaction should still have a lot of buffers for its share, even after dividing the buffer pool size in KB by the new page size in KB.

Note The CDB files for the configuration database are ASCII text files. If you view a CDB file, it will still say number_of_buffer = N for the size of the buffer pool for each MKF database. However, the number, N, is the number of KB in the buffer pool, not the number of buffers, despite what the field name says (unless, of course, the page size is 1 KB).

Stopping the FileNet software

Stop the FileNet software and clear the shared memory by entering the following commands. :

UNIX

```
initfnsw -y stop
killfnsw -DAy
```

WIN

```
initfnsw -y stop
killfnsw -Dy
```

Increasing the file size of the databases

The minimum number of pages in the base data file required by the new format is 3,739, 3,239, and 2,989 for 4 KB, 8 KB, and 16 KB pages, respectively. (The new format preallocates 2,739 pages, which is a good deal more than the original format preallocates.)

Note The expanded disc format preallocates more space than the original disk format. You might get an error unless the new database datafiles contain at least the following amount of

additional space:

4 KB pages — 15 MB
8 KB pages — 13 MB
16 KB pages — 12 MB

To calculate the number of megabytes required in the new format, multiply the number of pages by the new page size in KB (4, 8, or 16, for example), then divide by 1,024, taking care to round up. This results in a minimum base data file size of 12 MB, 28 MB, and 48 MB for 4KB, 8KB, and 16KB pages, respectively. If you are converting a database because you have reached the 16 GB addressing limit of the original format, you do not have to worry much about the minimum database size: The size of the new database must be larger than the size of the original database, which was already much larger than the minimum.

When converting to the new format, you probably want to reduce the number of database files and recovery log files. In earlier releases, MKF database data and recovery log files had to be less than 2 gigabytes (GB), because 2 GB is the maximum signed 32 bit integer. On IS 3.6.0 and later, MKF uses 64 bit integers for file I/O. Therefore, MKF data and recovery log files can be as large as the operating system supports.

Prior to IS 3.6.0, in order to reach the 16 GB limit of the original format, 8 files of 2 GB each were required. When converting such a database to the new format, you probably want to have only one data file and one recovery log file.

To delete data files or recovery log files

In **fn_edit**, select the Procedures tab > Delete Additional Datasets.

To increase or decrease the size of a data file or recovery log file

In **fn_edit** select the Datasets tab. Type in the new size in megabytes (where fn_edit defines 1 MB to be 1,024 times 1,024 bytes, or K times K bytes).

Note FileNet software prefers to allocate disk space in multiples of 4 MB, so round up the size in megabytes to the next greater multiple of 4.

Save your changes and exit **fn_edit**.

Run **fn_build -a** to generate the DDL text files from the updated CDB database information.

Platform dependent steps

Now you can increase the size of some logical volumes, deleted some logical volumes, create some logical volumes, etc., in accordance with the changes made by fn_edit. (fn_edit only changes the information in the FileNet CDB database, and fn_build -a only creates the MKF DDL text files from the information in the CDB database).

Creating, updating, and deleting logical volumes is a platform dependent procedure. To increase the database file size, follow the appropriate set of commands:

AIX To increase the database file size on AIX systems:

1 Log on as a user with **root** privileges.

Note To verify that the quantum is 4 MB, open another window and run

lsvg fnvg

Select Show characteristics of a logical volume to show the original size as well as the PP Size and check that the value of PP Size is 4.

2 Start up the system manager.

smit &

3 Select System Storage Management > Logical Volume Manager > Logical Volumes.

You can delete the extra logical volumes, if any. If you do, you must also delete the links.

Note You may want to delete extraneous logical volumes. To delete a logical volume, use the Remove a Logical Volume option. If logical volumes are deleted, their corresponding links must be deleted as well.

For example, if you want to increase the size of volume db_0 and you already have volumes db_1 or above, you must first delete db_1, db_2 and above before you can increase the size of db_0.

4 To increase the size of an existing logical volume, select Set Characteristics of a Logical Volume > Increase the Size of a Logical Volume.

5 The Increase the Size of a Logical Volume dialog box appears. To select the name of the logical volume, click List and select a logical volume. Click **OK**. Type the number of additional logical partitions you want in 4 MB increments and click **OK**.

Note You can achieve better performance by interleaving logical volumes across multiple physical volumes.

6 Skip to the section, **“All Platforms – Initialize database” on page 29**.

HPUX

To increase the database file size on HP-UX systems:

1 Log on as a user with **root** privileges.

2 Start the system manager.

```
sam &
```

You must be running on the console, not remotely.

3 Select Disks and File Systems, then Logical Volumes.

4 Skip to the section, **“All Platforms – Initialize database” on page 29**.

SOL

To increase the database file size on Solaris systems:

1 The Veritas volume manager is used to manage logical volumes. If you are not already running as a user with **root** privileges, set the current user to **root**. Start the Veritas Volume Manager by entering the following:

```
cd /opt/VRTSvmsa/bin
./vmsa &
```

2 To add a volume group, right-click on fndg in the Veritas volume manager GUI. Select the new volume option.

For example, to add a second recovery log logical volume to the permanent database, The volume would be fn_perm_rl1. Make a link by runningthe **ln** command in the shell window:

```
ln -s /dev/vx/rdsk/fndg/fn_perm_rl1 /fnsw/dev/1/
permanent_rl1
```

3 To increase a volume group, right-click on fndg in the Veritas volume manager GUI. Select the increase volume option and enter new values. Note that you must get the user, group, and permissions right on both the logical volume and on the link. In the case of the logical volume, you must get these right before

exiting the Veritas volume manager. Using **chmod** and **chown** afterwards is not sufficient. The permissions can become incorrect after a system reboot unless they were set properly within the Veritas volume manager.

4 Skip to the section, **"All Platforms – Initialize database" on page 29**.

WIN

To increase the database file size on Windows Server systems, since ordinary files are used, not logical volumes, you simply delete the extra data and recovery log files. Also, if any existing file is too large, delete it. If an existing file is too small, you do not have to do anything – MKF_ddl increases the size of the file.

If you make a mistake and delete all the data and recovery log files for the database being converted, MKF_ddl automatically recreates them.

All Platforms – Initialize database

1 Although root permissions were required to update the logical volumes, the root user may or may not be a member of the fn_admin group. If it is not, then set the current user to fnsw. If it is, continue.

2 Stop the FileNet software and clear the shared memory by entering the following command:

UNIX

```
initfnsw -y stop
killfnsw -DAy
```

WIN

```
initfnsw -y stop
killfnsw -Dy
```

3 Initialize the database by running the MKF_ddl tool with the -initialize option:

UNIX

UNIX example:

MKF_ddl /fnsw/local/sd/1/permanent.ddl -initialize

WIN

Windows example:

MKF_ddl d:\fnsw_loc\sd\1\permanent.ddl -initialize

Importing the data

To begin importing the data, run the MKF_import command.

- Use the **init** option if you want to maximize the speed of the import.
- Use the **add** option if you want to maximize the performance of the resulting database. The **add** option is much slower than the **init** option.

UNIX

UNIX examples:

MKF_import /fnsw/dev/1/permanent_db0 init in=/tmp/perm.exp

MKF_import /fnsw/dev/1/permanent_db0 init in=tape type=8mm device=/dev/rmt0

WIN

Windows examples:

MKF_import c:\fnsw\dev\1\permanent_db0 init in=e:\tmp\perm.exp

MKF_import d:\fnsw\dev\1\permanent_db0 init in=tape type=8mm device=TAPE0

Using the init option

If you use the init option, MKF_import imports the rows very quickly without generating the B-trees. Then MKF_import runs MKF_fixup automatically. MKF_fixup generates all the B-trees.

Using the add option

If you use the add option, MKF_import performs normal MKF database inserts, so MKF_fixup is not run. However, the add option is quite a bit slower than the init option.

Backing up the database

Make a full offline backup of the new database by using EBR. See:

- *Image Services Enterprise Backup and Restore User's Guide*

To download this document from the IBM support page, see **"Accessing IBM FileNet documentation" on page 8**.

Important Making a backup after this update to the MKF databases is especially important. Earlier backups cannot be restored after the database has been modified.

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 Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at **www.ibm.com/legal/copytrade.shtml**.

Microsoft and Windows 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 may be trademarks or service marks of others.

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.

Index

Program Number: 5724-R95

Printed in USA

SC19-2955-00