



StorHouse/RM Metadata Conversion Manual

StorHouse/RM Release 3.2

Publication Number
900142 Rev. C

March 28, 2002





All rights reserved. No part of this publication may be reproduced, translated, stored in any electronic retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of FileTek, Inc.

This publication Copyright © 1999-2002 by FileTek, Inc., Rockville, MD
Publication Number: 900142 Rev. C

NOTE: U.S. GOVERNMENT USERS

Restricted Rights Legend

Use, duplication or disclosure by the Government is subject to the restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or the Commercial Computer Software - Restricted Rights clause at 48 CFR 52.227-19, as applicable. Unpublished-rights reserved under the copyright laws of the United States. The contractor/manufacturer is:

FileTek, Inc.
9400 Key West Avenue
Rockville, Maryland 20850

Information in this document is subject to change without notice and does not represent a commitment on the part of FileTek, Inc. Further, FileTek, Inc. reserves the right to supplement the document with information not available at the time of creation of the document. FILETEK, INC. PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, AND CANNOT WARRANT THE RESULTS YOU MAY OBTAIN USING THE DOCUMENT. IN NO EVENT SHALL FILETEK, INC. BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OR DATA, INTERRUPTION OF BUSINESS, OR FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND, EVEN IF FILETEK, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES ARISING FROM ANY DEFECT OR ERROR IN THIS PUBLICATION. Some states or jurisdictions do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

FileTek and StorHouse are registered U.S. trademarks of FileTek, Inc. VRAM is a U.S. trademark of FileTek, Inc. All other brand or product names are trademarks or registered trademarks of their respective owners.

Documentation for FileTek's StorHouse product. Protected by the following U.S. Patents: 4,864,572; 5,247,660; 5,727,197; 6,049,804. Other patents pending.

Contents

Welcome	vii
Audience	vii
What's in this guide	vii
For more information	viii
Conventions	viii
 Chapter 1: Conversion overview	 1-1
Why you need to convert metadata	1-1
Features of metadata conversion	1-2
Restartable	1-2
Recoverable	1-2
Restorable	1-2
Auditable	1-2
Protective	1-2
Terminology	1-3
Active load	1-3
Active release	1-3
Base row	1-3
Database container directory	1-3
Database directory	1-3
Environment variable	1-3
Metadata entry	1-4
System tablespace	1-4

What's converted	1-4
Layout of database directories	1-4
Phases of metadata conversion	1-6
Phase 1: Copy the existing metadata	1-6
Phase 2: Create and update the new metadata	1-9
Copyout and copyin commands	1-11
Copyout examples	1-12
Copyin examples	1-13
Sample conversion listings	1-13
Copyout listing	1-14
Copyin listing	1-15
Locking during metadata conversion	1-17
Shared locks during copyout	1-18
Exclusive locks during copyin	1-18
System requirements and considerations	1-18
Disk space	1-18
Conversion scheduling	1-19
Security	1-19
Question and answer	1-20
Which StorHouse/RM software release do I use?	1-20
What's not converted?	1-20
Do I have to convert all databases at the same time?	1-20
Do I have to reboot the domain at any time?	1-20
Does metadata conversion overwrite existing metadata?	1-21
What happens to the source database directory and metadata?	1-21
What happens if I drop a table after conversion and have to revert back?	1-21
What happens if I load data after conversion and have to revert back?	1-21



Chapter 2: Conversion procedures	2-1
Preparing for conversion	2-1
Converting from release 2.3 to 3.2	2-4
Recovering from errors	2-9
Checking copyout or copyin status	2-9
Re-running copyout	2-10
Re-running copyin	2-11
Starting over	2-12
Restoring the previous release	2-15

Index



Contents

Welcome

This *StorHouse/RM Metadata Conversion Manual* explains how to convert metadata from StorHouse/RM release 2.3 to 3.2. This conversion is not required. Perform this metadata conversion only if you want to use the enhanced features in the StorHouse/RM UNIX file system.

Audience

This guide is intended for FileTek Customer Support or for the UNIX® system administrator who is responsible for running StorHouse UNIX utilities and shell scripts in the StorHouse server domain. It assumes you are familiar with StorHouse, the Solaris® operating system, the directory layout of StorHouse/RM software, and StorHouse/RM utilities such as the FileTek FTP Data Loader and the metadata backup utility.

What's in this guide

This document is organized as follows:

Chapter 1, “Conversion overview,” describes general conversion information such as features, terminology, phases, locking, command formats, and system requirements and considerations.

Welcome

For more information

Chapter 2, “Conversion procedures,” contains instructions for preparing for conversion, running conversion utilities, recovering from errors, and restoring the previous software release.

For more information

The *StorHouse Database Administration Guide*, publication number 900108, describes the StorHouse metadata. This guide also contains instructions for running the metadata backup utility, which you should run before and after metadata conversion.

The *FileTek FTP Data Loader Manual*, publication number 900115, explains how to check the status of any active loads, both FTP and MVS. Metadata conversion *cannot run successfully* when loads are incomplete.

Conventions

This guide uses the following conventions:

Convention	Meaning
Courier font	Command formats and examples, directory path names
[]	Optional command parameter
...	Repeatable command parameter
<i>Italics</i>	New terms, emphasized text, and publication titles
▼	Procedures

Conversion overview

This chapter describes general information about metadata conversion, including:

- Why you need to convert metadata
- Features of metadata conversion
- Terminology
- What's converted and not converted
- Layout of database directories
- Phases of metadata conversion
- Locking during metadata conversion
- System requirements and considerations

It also contains information you can reference during conversion, such as:

- Copyout and copyin command formats
- Sample conversion listings

Why you need to convert metadata

Metadata conversion reformats existing metadata so that a new software release can use it. This is necessary when the structure of a system table changes or when new system tables are added for a software release. In other words, metadata conversion rebuilds the existing metadata using the new data structures.

Features of metadata conversion

The *metadata conversion process* is a series of commands and utilities that you run in the StorHouse server domain. The following features help ensure a successful conversion.

Restartable. Metadata conversion consists of phases, so you can restart an interrupted conversion process without having to redo all of the completed work. See “Phases of metadata conversion” on page 1-6 for more information about the conversion process and restart points.

Recoverable. If an error or interruption—such as a power outage—occurs during conversion, metadata is protected against loss. The metadata conversion process does not overwrite existing metadata, but rather it creates a new database hierarchy and rebuilds the metadata in the new hierarchy. The existing metadata is always available for recovery.

Restorable. As long as you have not loaded new data or altered the metadata after converting it, you can restore the old metadata, activate the old software release, and continue operations as if no conversion had been done.

Auditable. The metadata conversion process reports the progress and status of conversion to the standard output device (unless redirected). The listing also includes errors and warnings. See “Sample conversion listings” on page 1-13 for more information about how you can track conversion.

Protective. During conversion, metadata is protected against updates through locking. See “Locking during metadata conversion” on page 1-17 for more information about when locking occurs and types of locks.

Terminology

The following terms are used in this guide.

Active load. A load that is in progress (currently running), successfully completed but not confirmed, or incomplete (failed) with errors. The metadata conversion process cannot run when loads are active.

Active release. The StorHouse/RM software release that is current when the StorHouse software is started. Although you can install different software releases in the same domain, only one release can run at a time. The `STH_RELEASE` environment variable in the operator `.cshrc.rm` file defines the active release.

Base row. An entry in the `SYSTABAUTH` system table that the `syscreate` utility inserts when you create a database. The `syscreate` utility grants `SELECT` privilege to `PUBLIC` for each system table except `SYSRANGES`. The metadata conversion process does not convert updates made to base rows.

Database container directory. The directory in the UNIX file system that contains all StorHouse database directories. The `STHDBS` environment variable defines the path to this directory. A *source* database container directory contains existing database directories, and a *target* database container directory contains new database directories.

Database directory. A directory that contains the metadata for a StorHouse database. Each database has its own database directory. A *source* database directory contains existing metadata, and a *target* database directory contains converted metadata.

Environment variable. A parameter that defines an aspect of your working environment, such as your home directory or frequently used directory path. Examples are `STHROOT`, `STHDBS`, and `STH_RELEASE`. You set values for environment variables in the operator `.cshrc` file.

Metadata entry. A row in a system table. For instance, each of the following is a metadata entry: table, index, view, synonym, segment file, column in a table, tablespace, and authorization (GRANT) record.

System tablespace. A synonym for database directory.

What's converted

Two enhancements have been made to the StorHouse/RM UNIX file system that contains the database directories. First, StorHouse/RM release 3.2 supports UNIX large files. This enhancement substantially increases the maximum number of rows that can be stored in system tables. Second, the system tables indexes are now smaller and faster to search.

You can continue to use your existing metadata with StorHouse/RM release 3.2, but the file size is limited to 2 gigabytes (GB). If you want to implement these enhancements for your existing metadata, then convert the metadata with the programs distributed with StorHouse/RM release 3.2. Any new databases that you create with StorHouse release 3.2 use the UNIX large files and enhanced indexes. There are no actual changes in the data stored; all rows are just copied out to temporary files and then copied back into newly created databases.

Note: The metadata conversion process does not convert user tables created in a system tablespace. After conversion, you must copy the user tables (which are UNIX files) to the newly converted database.

Layout of database directories

Each StorHouse database has its own database directory in the UNIX file system under the default path:

```
/filetek/sth/sthdb
```

For example, Figure 1-1 shows the directory layout for a StorHouse/RM release 2.3 system with one database called CALLS.

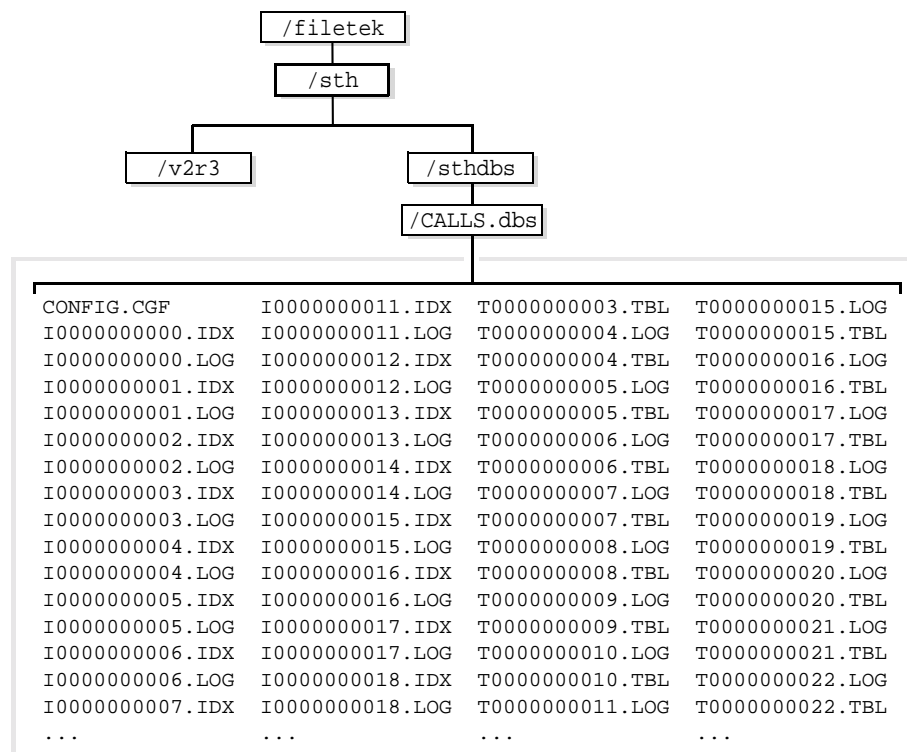


Figure 1-1: Directory layout for StorHouse/RM release 2.3

The directories and files in Figure 1-1 are:

- `/v2r3` contains StorHouse/RM software for release 2.3 (\$STHROOT).
- `/sthdb` is the database container directory for all StorHouse database directories (\$STHDBS).
- `.dbs` extension identifies a database directory. The database name precedes the extension, for example, `CALLS`.

- `CONFIG.CGF` is the system configuration file. Each database has a system configuration file containing a version ID to prevent access to incorrect data. For instance, if a release 3.2 engine attempts to access release 2.2 metadata, the engine terminates immediately with SQL code -50700.
- `Innnnnnnnnnn.IDX` files are system table indexes.
- `Innnnnnnnnnn.LOG` files are system table index logs.
- `Tnnnnnnnnnnn.TBL` files are system tables.
- `Tnnnnnnnnnnn.LOG` files are system table logs.

Phases of metadata conversion

Metadata conversion consists of two phases:

- Phase 1: Copy the existing metadata
- Phase 2: Create and update the new metadata

Phase 1: Copy the existing metadata

After installing the new software and conversion utilities but while the old software is the active release, make a copy of the current metadata by running the copyout utility. You can copy the metadata of one database, multiple databases, or all databases at a time.

The *copyout* utility:

- Reads the metadata in the specified source database directories
- Transforms the metadata entries into binary data

- Creates the following files in a CONVERT subdirectory under each database directory:
 - The STATEMENTS file holds the INSERT statements to be executed during the copyin phase. These statements contain host variable markers in place of actual column values.
 - The system table files contain the entries for each system table to be converted as well as for any new system tables. The file name is the same as the system table name, for instance, SYSCOLUMNS and SYSSTHFILES.
 - The StartTime file contains the time the copyout started.
 - The SaveConfig file contains a copy of the system configuration file.
 - The EndTime file contains the time the copyout completed.
 - The Version file contains the software release you're converting from and to, for instance, from 2.3 to 3.2.

Figure 1-2 shows the contents of the CONVERT subdirectory after running the copyout utility for the CALLS database. The software in the /v2r3 directory is the active release.

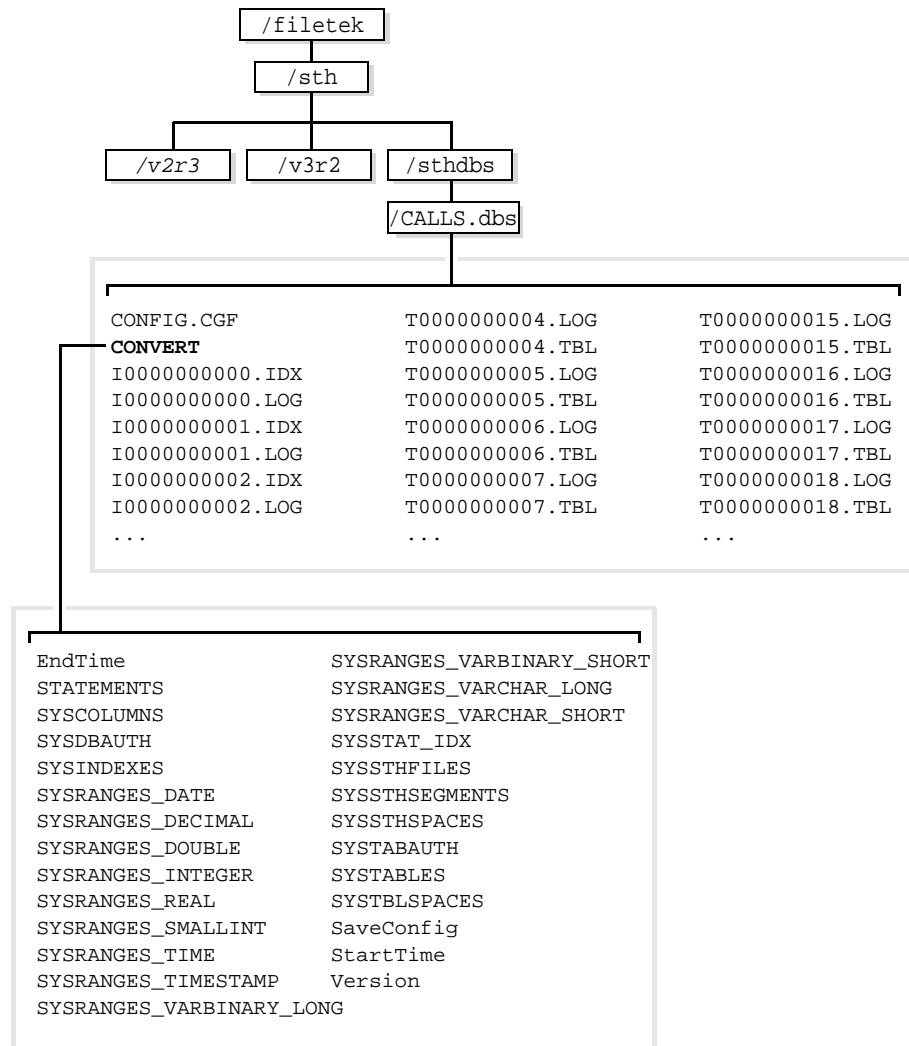


Figure 1-2: CONVERT subdirectory after running the copyout utility

Phase 2: Create and update the new metadata

After you have copied all of the metadata for all databases, create the new metadata with the `syscreate` utility and update that metadata with the `copyin` utility. You can do this for one database, multiple databases, or all databases at a time. Before you can start phase 2, however, you must:

- Shut down the StorHouse software
- Rename the source database container directory
- Create a target database container directory
- Activate the new software release
- Change (`cd`) to the source database container directory

The *copyin* utility.

- Verifies the current working directory and prompts you (if the output has not been redirected) to verify that it is the source database container directory
- Checks the `StartTime` file in the `CONVERT` subdirectory to verify that no metadata updates have occurred since copyout began
- Checks the `SaveConfig` file in the `CONVERT` subdirectory to determine the next table ID and index ID in the new database
- Checks the `EndTime` file in the `CONVERT` subdirectory to verify that copyout completed successfully
- Checks the `Version` file in the `CONVERT` subdirectory to make sure that the correct conversion is being performed (for instance, from 2.3 to 3.2)
- Executes the SQL statements in the `STATEMENTS` file in the `CONVERT` subdirectory, getting the data from the system table files
- Stores the converted metadata entries in the new system tables

The source database directory and the files in the CONVERT subdirectory are kept in case you need to restore the previous software and metadata state. Figure 1-3 shows sample database directories after running copyin. The source database container directory is renamed `/sthdb.v2r3` and the target database container directory is named `/sthdb`. The software in the `/v3r2` directory is the active release.

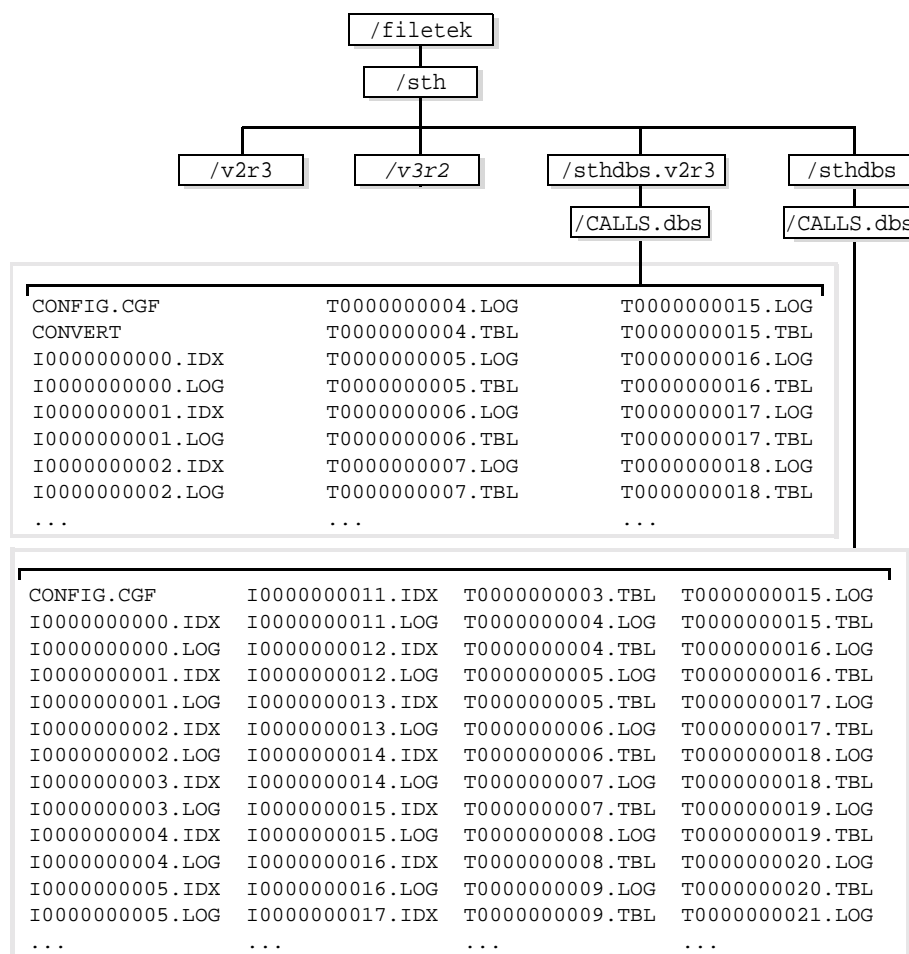


Figure 1-3: Database directories after running the copyin utility

Copyout and copyin commands

The copyout and copyin utilities reside in the `$STHROOT/bin` directory. The command formats are:

```
copyout database_name [database_name ...][-p][-n][-h]
```

```
$STHROOT/bin/copyin database_name [database_name ...]  
[-p][-n][-h][-f]
```

Parameters for copyout and copyin commands

Parameter	Description
database_name	<p>(required) Provides the names of the databases to be converted. You must specify at least one database. A database name is case sensitive and may include or consist of the wildcard character (*). You must use quotes (single or double) or the escape character (\) with the wildcard to prevent the UNIX shell from expanding the * to a list of matching files. Valid wildcards:</p> <ul style="list-style-type: none"> ■ “*” ■ ‘*’ ■ * <p>Examples:</p> <ul style="list-style-type: none"> ■ <code>copyout CALLS</code> – copies the metadata of a database called CALLS ■ <code>copyout CALLS SERVICE</code> – copies the metadata of two databases named CALLS and SERVICE ■ <code>copyout db“*”</code> – copies the metadata of all databases with names that start with db, such as db1, db2, db3, and so on, in the current working directory ■ <code>copyout *</code> – copies the metadata of all databases in the \$STHDBS directory

Parameters for copyout and copyin commands (continued)

Parameter	Description
-p	(optional) Password option. Supplies the password for the StorHouse SYSADM account. If you omit the -p parameter on copyout or copyin, the utility prompts you to supply the password. Note, however, that if you redirect the output, you must use this option to supply the password on the copyout and copyin commands.
-n	(optional) No execution option. Checks and reports the copyout or copyin status of one or more databases. You can use the -n option while the StorHouse software is up or down. Note that before running copyin with the -n parameter, you must first rename the \$STHDBS directory, create a target database container directory, and create the new database directory.
-h	(optional) Help option. Displays information about copyout and copyin command syntax and available parameters.
-f	(optional) Force option. Indicates that the specified databases should be copied in, even though metadata updates may have occurred after copyout began. Use this option only when you're sure that the metadata has not changed since copyout began, for example, if the change is from a UNIX touch command.

Copyout examples

- To check the copyout status of all databases:

```
copyout \* -n
```

- To copy out the metadata of a database named CALLS:

```
copyout CALLS
```

- To copy out the metadata of all databases:

```
copyout "*"
```

Copyin examples

- To check the copyin status of the CALLS database:

```
$STHROOT/bin/copyin CALLS -n
```

- To copy in new metadata for a database named CALLS:

```
$STHROOT/bin/copyin CALLS
```

- To copy in new metadata for all databases:

```
$STHROOT/bin/copyin '*'
```

Sample conversion listings

The copyout and copyin utilities log status, warning, and error messages to the standard output device. The listings provide an audit trail for conversion, so you may want to redirect the output to a file in the `/filetek/operator` directory. You should not store output in the `$STHROOT/bin` directory.

Copyout listing

Excerpts from a copyout listing follow. In this example, the CALLS database is being converted.

```
sthdev2.32> $STHROOT/bin/copyout CALLS -p sysadm

27Mar2002-14:35:40: The following 1 databases will be converted:

27Mar2002-14:35:40:      CALLS

CALLS 27Mar2002-14:35:40: Starting copyout
CALLS 27Mar2002-14:35:45: Checking loads
CALLS 27Mar2002-14:35:45:      ... OK
CALLS 27Mar2002-14:35:45: Next id:

CALLS 27Mar2002-14:35:45: User table / index:           131           136
CALLS 27Mar2002-14:35:45: System table / index:         40           41

CALLS 27Mar2002-14:35:45: select id, tsname, "STORAGE_MANAGER", num_subspaces from
SYSTBLSPACES where id > 1

CALLS 27Mar2002-14:35:45: Rows fetched = 1

CALLS 27Mar2002-14:35:45: select * from SYSDBAUTH where grantee != 'SYSADM'

CALLS 27Mar2002-14:35:46: Rows fetched = 6

CALLS 27Mar2002-14:35:46: select * from SYSTABAUTH where tbl in (select tbl from
sysadm.systables where id >= 100) or ( grantee != 'SYSADM' and (grantee !=
'PUBLIC' or (ins = 'y' or del = 'y' or upd = 'y' or ndx = 'y' or 'alt' = 'y' or ref
= 'y' )) )

CALLS 27Mar2002-14:35:48: Rows fetched = 23

CALLS 27Mar2002-14:35:48: select * from SYSCOLAUTH

CALLS 27Mar2002-14:35:48: select * from SYSVIEWS

CALLS 27Mar2002-14:35:48: Rows fetched = 3

CALLS 27Mar2002-14:35:48: select * from SYSSYNONYMS

CALLS 27Mar2002-14:35:48: Rows fetched = 1

...

CALLS 27Mar2002-14:36:01: select * from SYSINDEXES where id >= 100

CALLS 27Mar2002-14:36:01: Rows fetched = 10

CALLS 27Mar2002-14:36:01: copyout completed successfully
```

Copyout checks for active loads and logs an error if any are detected.

These are the next table and index IDs to be assigned in the converted database.

Copyout reads the system tables to determine which tables need to be converted and the number of rows to be converted.

Copyin listing

Excerpts from a copyin listing follow. Note that the copyin utility reports v3r0 as the target release. The conversion works correctly because there are no metadata differences (that matter to conversion) between 2.3, 3.0, 3.1, and 3.2.

You must first change to the source database container directory.

```
sthdev2.57> cd /filetek/sth/sthdb.v3r2
sthdev2.58> $STHROOT/bin/copyin CALLS -p sysadm
```

Copyin displays the current working directory and requests confirmation.

```
copyin from /filetek/sth/sthdb.v3r2 to release v3r0.

Is this correct? [Enter y to confirm]: y

27Mar2002-14:40:42: The following 1 databases will be converted:

27Mar2002-14:40:42:     CALLS

27Mar2002-14:40:42: Starting copyin

CALLS 27Mar2002-14:40:42: Checking new database directory ...

CALLS 27Mar2002-14:40:42: Database directory /filetek/sth/sthdb/CALLS.dbs
```

Copyin checks CONFIG.CGF in the new database directory to be sure no new user tables and indexes have been created since copyin began.

```
CALLS 27Mar2002-14:40:42: Next id:

CALLS 27Mar2002-14:40:42: User table / index:           100           100
CALLS 27Mar2002-14:40:42: System table / index:         40           41

CALLS 27Mar2002-14:40:42: Checking copyout status of /filetek/sth/sthdb.v3r2/
CALLS.dbs

CALLS 27Mar2002-14:40:42: Next id:

CALLS 27Mar2002-14:40:42: User table / index:           131           136
CALLS 27Mar2002-14:40:42: System table / index:         40           41

CALLS 27Mar2002-14:40:42: Checking T0000000000.TBL

CALLS 27Mar2002-14:40:42: T0000000000.TBL is up to date
```

1**Conversion overview**

Sample conversion listings

```

CALLS 27Mar2002-14:40:42: Checking T0000000001.TBL

CALLS 27Mar2002-14:40:42: T0000000001.TBL is up to date

CALLS 27Mar2002-14:40:42: Checking T0000000002.TBL

CALLS 27Mar2002-14:40:42: T0000000002.TBL is up to date

CALLS 27Mar2002-14:40:42: Checking T0000000003.TBL

CALLS 27Mar2002-14:40:42: T0000000003.TBL is up to date

CALLS 27Mar2002-14:40:42: Checking T0000000004.TBL

CALLS 27Mar2002-14:40:42: T0000000004.TBL is up to date

CALLS 27Mar2002-14:40:42: Checking T0000000005.TBL

...

CALLS 27Mar2002-14:40:42: copyout is up to date

```

Copyin locks the system tables, checks for active loads, and verifies that the number of entries expected matches the number of actual entries.

```

CALLS 27Mar2002-14:40:43: Locking system tables

CALLS 27Mar2002-14:40:46: Checking loads

CALLS 27Mar2002-14:40:46:    ... OK

CALLS 27Mar2002-14:40:46: Checking number of entries in SYSTBLSPACES

CALLS 27Mar2002-14:40:47: Checking number of entries in SYSDBAUTH

CALLS 27Mar2002-14:40:47: Checking number of entries in SYSTABAUTH

CALLS 27Mar2002-14:40:47: Checking number of entries in SYSCOLAUTH

CALLS 27Mar2002-14:40:47: Checking number of entries in SYSVIEWS

CALLS 27Mar2002-14:40:47: Checking number of entries in SYSSYNONYMS

CALLS 27Mar2002-14:40:47: Checking number of entries in SYSSTHFILES

...

```

Copyin inserts the entries into the system tables.

```

CALLS 27Mar2002-14:40:48: insert into SYSTBLSPACES values ( :id, :tsname,
:storage_mgr, :num_subspaces )

...

CALLS 27Mar2002-14:40:48: 1 rows inserted

```



```
CALLS 27Mar2002-14:40:48: insert into SYSDBAUTH values ( :grantee, :accdba,
:accres, :accscn )
```

```
CALLS 27Mar2002-14:40:48: 6 rows inserted
```

```
CALLS 27Mar2002-14:40:48: insert into SYSTABAUTH values ( :grantor, :grantee,
:tblowner, :tbl, :ins, :del, :upd, :sel, :ndx, :alt, :ref )
```

```
CALLS 27Mar2002-14:40:49: 23 rows inserted
```

```
...
```

Copyin checks
CONFIG.CGF in the
new database
directory to be sure no
new user tables or
indexes were created
since copyin began.
Then it checks the old
CONFIG.CGF to
obtain the next table
and index IDs.

```
CALLS 27Mar2002-14:40:50: 14 tables converted
```

```
CALLS 27Mar2002-14:40:51: Rechecking CONFIG file
```

```
CALLS 27Mar2002-14:40:51: Next id:
```

```
CALLS 27Mar2002-14:40:51: User table / index:           100           100
```

```
CALLS 27Mar2002-14:40:51: System table / index:         40           41
```

```
CALLS 27Mar2002-14:40:51: Updating CONFIG file
```

```
CALLS 27Mar2002-14:40:51: Next id:
```

```
CALLS 27Mar2002-14:40:51: User table / index:           131           136
```

```
CALLS 27Mar2002-14:40:51: System table / index:         40           41
```

```
CALLS 27Mar2002-14:40:51: Copyin completed successfully
```

```
27Mar2002-14:40:51: Copyin complete
```

Locking during metadata conversion

Metadata conversion attempts to minimize system down time and at the same time prevent metadata changes during the process. To do this, metadata conversion places both shared (or read) and exclusive (or write) locks on system tables. Locking occurs during copyout and copyin phases.

Shared locks during copyout

The copyout utility places *shared locks* on the system tables. This means that query activity can continue during this phase of metadata conversion. To capture a consistent and known metadata state, however, activities that update metadata—such as active loads or Data Definition Language (DDL) statements like GRANT, REVOKE, CREATE, and DROP—cannot be running. If they are running, the metadata conversion process queues the active loads and DDL requests when you start the copyout utility and re-starts them when copyout completes. This invalidates the copyout, that is, the copyin utility will fail and you must re-run copyout.

Exclusive locks during copyin

The copyin utility places *exclusive locks* on the metadata. This prevents queries, loads, and DDL statements from executing while copyin is running. The copyin utility checks to be sure that the metadata has not been altered between conversion phases. If the metadata has been updated, you must re-run copyout before you can run copyin. After copyin completes, any queued activities will proceed in the order in which they were queued.

System requirements and considerations

To help you plan for conversion, this section describes space requirements, conversion scheduling, and security requirements.

Disk space

Sufficient disk space must be available to hold two copies of metadata (the old metadata and the new metadata) and copyout output (files in the CONVERT subdirectory) for each database. This space is in the same partition as the

\$STHDBS directory. The conversion utilities do not check disk space before running; however, they terminate when space is unavailable. If necessary, you can create symbolic links for the CONVERT subdirectories in the primary file partition to the actual directories located in a different partition. See page 2-3 for instructions on creating symbolic links.

Conversion scheduling

A database is unavailable from the time StorHouse software is shut down after copyout until the database has been completely copied in. You should plan for the appropriate amount of down time, allowing additional time for other conversion tasks, such as verifying the conversion.

Note: The conversion utilities (both copyout and copyin) can process approximately 500,000 metadata entries per hour (on a 250MHz machine).

If you need to convert multiple databases, you can schedule the copyout phase for different databases at different times. This enables you to copy out one database while loads are running in other databases. You can also schedule the copyin phase for different databases at different times; but once the new software release is active, a database is unavailable until copyin has completed.

Security

To perform metadata conversion, you must log into the StorHouse server with the operator account and password. This ensures that the conversion utilities have the required access to UNIX files and directories. Additionally, when running copyout and copyin, you must provide the account password for the StorHouse SYSADM account. This ensures that the conversion utilities can read and write to the system tables.

Question and answer

This section summarizes important information about metadata conversion.

Which StorHouse/RM software release do I use?

You use the old software release during copyout and the new software release during copyin. Both software releases can reside in the same domain but cannot run at the same time. Only one StorHouse/RM release can be active at a time.

What's not converted?

Metadata conversion does not reference, change, or move user data (segments). If any user tables are stored in the system tablespace, the metadata conversion process logs a warning but does not convert this data. If this happens, contact FileTek for further assistance. Additionally, metadata conversion does not convert base entries in system tables. If you changed any base entries in the old database, you must make those changes in the new database.

Do I have to convert all databases at the same time?

No, you do not have to convert all databases at one time. You can copy out the metadata of one database, multiple databases, or all databases at the same time or at different times. This enables you to continue running loads in some databases while converting others. You should, however, copy out all databases before activating the new release and running copy in. You can then copy in the new metadata of one or more databases at the same time or at different times.

Do I have to reboot the domain at any time?

No, you do not reboot the domain. After running copyout, you must stop the StorHouse software and activate the new StorHouse/RM release, but a domain reboot is not necessary. See page 2-6 for more information about stopping the StorHouse software and activating the new StorHouse/RM release.

Does metadata conversion overwrite existing metadata?

No, metadata conversion does not overwrite existing metadata, it makes a copy of the existing metadata. After metadata conversion completes, there are two copies of metadata: the old metadata in the source database directory and the new metadata in the target database directory.

What happens to the source database directory and metadata?

Nothing happens to the source database directory and metadata. They are retained in case you need to restore the old software release and metadata.

What happens if I drop a table after conversion and have to revert back?

After the copyin is completed, if you drop a user table (or an index) and then revert to the previous release, the data and index are gone. The two sets of metadata (old and converted) refer to exactly the same data. Be careful about dropping user data after conversion.

What happens if I load data after conversion and have to revert back?

After the copyin is completed, if you load data into the converted database or add new user tables and indexes (or submit any DDL) and then revert to the previous release, your user data and metadata will be out of sync. In other words, the old metadata will not contain new entries for the loaded data or changed entries for the DDL updates. Be certain that you are ready to use the converted database before loading data or submitting DDL statements.

1

Conversion overview

Question and answer

Conversion procedures

This chapter explains how to:

- Prepare for conversion
- Convert metadata from release 2.3 to 3.2
- Recover from errors
- Restore the previous release

Preparing for conversion

Before converting from release 2.3 to 3.2, use this procedure to prepare for conversion.

▼ To prepare for metadata conversion

1. For each database to be converted, submit queries for testing purposes.

After conversion, you can run these same queries to check for matching results.

2. For each database to be converted, determine whether any updates were made to base rows in the SYSTABAUTH system table by submitting the following query:

```
SELECT * FROM SYSADM.SYSTABUTH
WHERE tbl LIKE 'SYS%' AND
grantee = 'PUBLIC' AND
```

```
grantor = 'SYSADM' AND  
tblowner = 'SYSADM' AND  
( ins <> " OR del <> " OR upd <> " OR  
sel <> 'y' OR ndx <> " OR alt <> " OR ref <> " ) ;
```

The copyout utility ignores any changes made to base rows. You must re-do any updates after running the copyin utility.

3. For each database to be converted, check the status of loads and complete any active loads.

The copyout and copyin utilities check for active loads and log an error if any exist. To check the status of active loads in a database, use File Transfer Protocol (FTP) to submit a `status` keyword on the FTP `put` command with the FileTek FTP Data Loader. You can do this for both MVS and FTP loads. Refer to the *FileTek FTP Data Loader Manual* for more information about using FTP to check the status of loads and to confirm and abort loads.

The following example checks the status of all loads for the `CALLS` database.

```
ftp> put - status,dbn=CALLS
```

If no loads are listed, then all loads are confirmed and you can continue with metadata conversion. If loads are listed, you must confirm them, abort them, or wait until they finish running and then confirm them before continuing with conversion.

4. For each database to be converted, run the metadata backup utility.

Refer to the *StorHouse Database Administration Guide* for instructions on backing up metadata. The following example is the StorHouse command to back up metadata (in this example, for the `CALLS` database). Remember to enclose lowercase database names in double quotes.

```
? run meta_bkup "CALLS"
```


5. Check the disk space in the `$STHDBS` directory.

```
df $STHDBS
```

All of the database directories are stored under the `$STHDBS` directory. There should be enough space to hold two copies of metadata for each database.

- If there is enough space, go to step 7 on page 2-4.
 - If there isn't enough space, create symbolic links for the `CONVERT` subdirectories (go to step 6).
6. Create any symbolic links for the `CONVERT` subdirectories in the primary file partition to a directory in a different partition.
 - a. First, create the directory to hold the conversion output files.

The following example uses the `/filetek2` partition to hold the conversion files for a database named `CALLS`. This example assumes that the `/filetek` partition holds the database directories.

```
mkdir -p /filetek2/sth/sthdbms/CALLS.dbs/CONVERT
```

- b. Then create the symbolic link to the directory.

For example:

```
ln -s /filetek2/sth/sthdbms/CALLS.dbs/CONVERT  
$STHDBS/CALLS.dbs/CONVERT
```

7. In the `filetek/operator` directory, check the `cschrc.rm` link to determine the current software release number (in case you need to reactivate the previous release).

```
ls -l .cschrc.rm
```

In the following example, 2.3 is the current software release number:

```
.cschrc.rm ->filetek/sth/v2r3/.cschrc.rm
```

8. Install the StorHouse/RM release 3.2 software and accompanying conversion programs.

Follow the installation instructions provided with the software. Do *not* activate the new software release yet. StorHouse/RM release 2.3 must be the current release for the first phase of conversion.

Converting from release 2.3 to 3.2

After you have prepared for conversion, use this procedure to convert the metadata from release 2.3 to 3.2.

▼ To convert metadata from release 2.3 to 3.2

1. Log in to the StorHouse server with the operator account and password.
2. Change to the `$STHROOT/bin` directory.

```
cd $STHROOT/bin
```

3. Run the copyout utility for each database to be converted.

See page 1-11 for copyout command format and parameters. When working in `$STHROOT/bin`, include `./` before the command. If you redirect the

output, provide the `-p` parameter and store the output in a different directory. Some copyout examples follow.

- To copy out the `CALLS` database without redirecting the output:

```
./copyout CALLS
```

When you omit the `-p` parameter, the copyout utility prompts you to supply the password for the `SYSADM` account.

```
Enter password for sysadm account:
```

- To copy out the `CALLS` database without redirecting the output and working in a different directory:

```
$STHROOT/bin/copyout CALLS
```

- To copy out the `CALLS` database, provide the `-p` parameter (where password is the password), redirect the output to a file called `copyout.out` in the `/filetek/operator` directory, and run the command in the background (`&`):

```
./copyout CALLS -p password >& \  
/filetek/operator/copyout.out &
```

4. Check the output.

For each database converted, look for this message:

```
copyout completed successfully
```

If the copyout completed successfully, continue with the next step. If the output contains asterisks (*), which flag errors, see page 2-9, “Recovering from errors.”

5. Stop the StorHouse software.

```
stopsm
```

6. Clear any resource locks (in case the structure of the system resource locks changed between software releases).

```
sxm cleanipcs
```

7. Rename the source database container directory as follows:

```
mv $STHDBS $STHDBS.v2r3
```

The database directories under this container directory remain intact.

8. Create a target database container directory as follows:

```
mkdir $STHDBS
```

The converted metadata will be stored under this container directory.

9. Activate the new StorHouse/RM software release.

- a. In the `/filetek/operator` directory, rename the `.cshrc.rm` link.

For example:

```
mv .cshrc.rm .cshrc.rm.old
```

- b. Add the following symbolic link:

```
ln -s /filetek/sth/v3r2/.cshrc.rm .cshrc.rm
```

- c. Log off, and then log in with the operator account and password.

10. Start the StorHouse software.

```
startsm
```

11. Run the syscreate utility for each database to be converted.

For example, to create the `CALLS` database:

```
syscreate CALLS
```

12. Change to the source database directory container.

For example, to change to the directory containing the 2.3 databases:

```
cd /filetek/sth/sthdb.v2r3
```

13. Run the copyin utility for each database to be converted.

See page 1-11 for copyin command format and parameters. Some copyin examples follow.

– To copy in the `CALLS` database without redirecting the output:

```
$STHROOT/bin/copyin CALLS
```

When you omit the `-p` parameter, the copyin utility prompts you to supply the password for the `SYSADM` account.

```
Enter password for sysadm account:
```

The copyin utility displays the working directory and conversion release and requests verification, for example:

```
copyin from /filetek/sth/sthdb.v2r3 to release v3r0.  
Is this correct? [Enter y to confirm]:
```

At the prompt `Is this correct? [Enter y to confirm]`, type `y` if you are in the correct working directory that contains the old databases. The target release (`v3r0`) is okay. If you type `n`, the `copyin` utility terminates. Change to the source database directory container, and run the `copyin` utility again.

- To copy in the `CALLS` database, provide the `-p` parameter (where password is the password), redirect the output to a file called `copyin.out` in the `/filetek/operator` directory:

```
$STHROOT/bin/copyin CALLS -p password >& \  
/filetek/operator/copyin.out
```

14. Check the output.

For each database converted, look for this message:

```
Copyin completed successfully
```

If the `copyin` completed successfully, continue with the next step. If the output contains asterisks (*), which flag errors, see page 2-9, “Recovering from errors.”

15. Verify the converted databases.

- Submit the test queries and compare the results.

The results should be the same as the queries submitted on the 2.3 database. If the results don't match, contact FileTek. Be prepared to back out to the previous release (see “Restoring the previous release” on page 2-15).

- Load data into the new database.



Warning: Once you have loaded data or updated the converted metadata, you will lose those updates and the data will be inaccessible if you restore the previous release.

16. Run a metadata backup for each successfully converted database.

Recovering from errors

This section explains how to handle errors or irregularities that occur during metadata conversion.



Warning: Once a database has been copied in, do not use DROP TABLE and DROP INDEX statements to recover from problems. These statements delete data and indexes. If you use DROP statements, the only way to recover is to re-create the table and indexes and then reload the data.

Checking copyout or copyin status

You can check the conversion status of one or more databases. This is helpful if you're not sure whether copyout or copyin completed successfully. The copyout or copyin utility reports the conversion status on the listing.

▼ To check the conversion status

Submit the copyout or copyin command with the `-n` parameter. Provide a database name to check the status of a specific database or use the wildcard (`*`) to check the status of all databases.

For example, to check the copyout status of the CALLS database (working in the `$STHROOT/bin` directory):

```
./copyout CALLS -n
```

The utility checks the metadata and logs the status on the listing. For instance, the CALLS database has been copied out.

```
CALLS 27Mar2002-13:18:43: Starting copyout  
CALLS 27Mar2002-13:18:43: Checking T0000000000.TBL
```

```

CALLS 27Mar2002-13:18:43: T0000000000.TBL is up to date
CALLS 27Mar2002-13:18:43: Checking T0000000001.TBL
CALLS 27Mar2002-13:18:43: T0000000001.TBL is up to date
...
CALLS 27Mar2002-13:18:43: Checking T0000000025.TBL
CALLS 27Mar2002-13:18:43: T0000000025.TBL is up to date
CALLS 27Mar2002-13:18:43: copyout is up to date
CALLS 27Mar2002-13:18:43: Copyout complete

```

The SERVICE database has not been copied out.

```

SERVICE 27Mar2002-11:10:48: Starting copyout
SERVICE 27Mar2002-11:10:48: /filetek/sth/sthdb/STHDBS/SERVICE.dbs/CONVERT/
SERVICE 27Mar2002-11:10:48: StartTime does not exist
SERVICE 27Mar2002-11:10:48: ** copyout is not up to date
SERVICE 27Mar2002-11:10:48: Copyout complete

```

Re-running copyout

If a copyout terminated with an error, or if the metadata was updated after the copyout utility completed but *before* the copyin utility started, use this procedure to re-run the copyout.

Note the following:

- StorHouse/RM release 2.3 must be the current release.
- The copyout utility checks for completed databases and does not re-copy them.

▼ To re-run copyout

1. Check the copyout listing and correct any errors.

The listing should contain enough information to help you correct the problem.

2. Run the copyout utility for each database to be converted.

See page 1-11 for copyout command format and parameters.

3. When copyout completes successfully, continue with the conversion procedure starting at step 5 (Stop the StorHouse software) on page 2-6.

Re-running copyin

If the copyin utility reported an error or was interrupted, you can re-run it. If you attempt to copy in a database that has already been copied in, the copyin utility terminates with an error.

Note: If copyin failed because of an invalid copyout, that is, the metadata was updated after copyout started, then see “Starting over” on page 2-12.

▼ To re-run copyin

1. Verify that you are in the correct database container directory:

```
ls -d $STHDBS*
```

You should see a directory named `/filetek/sth/sthdb`s and a directory named `/filetek/sth/sthdb`s.v2r3.



Warning: If you do not see both directories, do *not* delete anything. Contact FileTek if you do not see these directories.

2. Remove the target database directory with the following command, replacing `database_name` with the name of the database:

```
rm -r $STHDBS/database_name.dbs
```

For example, to remove the `CALLS` database directory:

```
rm -r $STHDBS/CALLS.dbs
```



Warning: Be sure you are deleting the newly created metadata, not the old metadata. You will lose your metadata if you delete the wrong database directory. Contact FileTek if you aren't sure whether you're deleting the correct database directory.

3. Run the `syscreate` utility to create the database.

For example, to create the `CALLS` database:

```
syscreate CALLS
```

4. Run the `copyin` utility for the database.

See page 1-11 for `copyin` command format and parameters.

5. When `copyin` completes successfully, continue with the conversion procedure starting at step 15 (Verify the converted databases) on page 2-8.

Starting over

The `copyin` utility fails immediately if any system tables were modified since `copyout` started. You can recover from this two ways:

- Reactivate the previous release and re-run `copyout`.

- Force the copyin utility to continue by using the -f parameter on the copyin command. Use this method only if you are certain that the metadata has not actually changed since copyout, for example, if a UNIX touch command made a change.



Warning: If the metadata has actually been changed (by any DDL statement) and you do not re-run copyout, you will lose those changes.

▼ **To reactivate the previous software release and re-run copyout**

1. Stop the StorHouse software.

```
stopsm
```

2. Clear any remaining resource locks.

```
sxm cleanips
```

3. Activate the previous software release.

- a. In the `/filetek/operator` directory, rename the `.cshrc.rm` link.

```
mv .cshrc.rm .cshrc.rm.v3r2
```

- b. Rename the old link back to `.cshrc.rm`.

For example:

```
mv .cshrc.rm.old. cshrc.rm
```

- c. Add a symbolic link for the appropriate software release.

For example, to activate release 2.3:

```
ln -s /filetek/sth/v2r3/.cshrc.rm .cshrc.rm
```

- d. Log off, and then log in with the operator account and password.

4. Start the StorHouse software.

```
startsm
```

5. Run the copyout utility for the applicable databases.

6. Stop the StorHouse software.

```
stopsm
```

7. Clear any remaining resource locks.

```
sxm cleanipcs
```

8. Activate the new StorHouse/RM software release.

- a. In the `/filetek/operator` directory, rename the `.cshrc.rm` link.

For example:

```
mv .cshrc.rm .cshrc.rm.old
```

- b. Add the following symbolic link:

```
ln -s /filetek/sth/v3r2/.cshrc.rm .cshrc.rm
```

- c. Log off, and then log in with the operator account and password.

9. Start the StorHouse software:

```
startsm
```

10. Run the copyin utility for each database to be converted.

11. When copyin completes successfully, continue with the conversion procedure starting at step 15 (Verify the converted databases) on page 2-8.

▼ **To force the copyin utility to continue even if copyout may not be up to date**

1. Re-run the copyin utility, specifying the `-f` parameter on the copyin command.

For example, to force the copyin utility to convert the `CALLS` database:

```
$STHROOT/bin/copyin CALLS -f
```

2. When copyin completes, continue with the conversion procedure starting at step 15 (Verify the converted databases) on page 2-8.

Restoring the previous release

If it is necessary to return to the previous software release, use this procedure to abandon the conversion.



Warning: If you loaded data into the converted database or updated the metadata, the data and metadata changes will be lost when you restore the previous release.

▼ **To restore the previous release**

1. Stop the StorHouse software.

```
stopsm
```

2. Clear any remaining resource locks.

```
sxm cleanips
```

3. Rename the target database container directory as follows:

```
mv $STHDBS $STHDBS.v3r2
```

4. Rename the source database container directory as follows:

```
mv $STHDBS.v2r3 $STHDBS
```

5. Activate the previous software release.

- a. In the `/filetek/operator` directory, rename the `.cshrc.rm` link.

```
mv .cshrc.rm .cshrc.rm.v3r2
```

- b. Rename the old link back to `.cshrc.rm`.

For example:

```
mv .cshrc.rm.old .cshrc.rm
```

- c. Add a symbolic link for the appropriate software release.

For example, to activate release 2.3:

```
ln -s /filetek/sth/v2r3/.cshrc.rm .cshrc.rm
```

- d. Log off, and then log in with the operator account and password.

6. Start the StorHouse software.

```
startsm
```

Index

Symbols

\$STHDBS environment variable 1-5
\$STHROOT environment variable 1-3
* wildcard character 1-11
.cshrc file 1-3
.dbs extension 1-5

A

abandoning conversion 2-15
aborting loads 2-2
account, operator 1-19
activating new software 2-6
active load 1-3
active release 1-3

B

backing up metadata 2-2

C

case sensitive 1-11
checking
 disk space 2-3
 status of loads 2-2
cleanipcs command 2-6
clearing resource locks 2-6
command options
 -f (force) 1-12
 -h (help) 1-12
 -n (no execution) 1-12
 -p (password) 1-12
commands
 cleanipcs 2-6
 copyin 1-11
 copyout 1-11
 df 2-3
 ln 2-3
 ls 2-11
 mkdir 2-6
 mv 2-6
 put 2-2
 rm 2-12
 run meta_bkup 2-2
 startsm 2-7
 stopsm 2-6
 syscreate 2-7
 touch 2-13
completing active loads 2-2
confirming loads 2-2
CONVERT subdirectory 1-8

- converting from 2.3 to 3.2
 - activating new software 2-6
 - clearing resource locks 2-6
 - creating target database container 2-6
 - introduction 2-4
 - renaming source database container 2-6
 - running a metadata backup 2-9
 - running copyin utility 2-7
 - running copyout utility 2-4
 - starting StorHouse software 2-7
 - stopping StorHouse software 2-6
 - verifying conversion 2-8

- copyin utility
 - commands and parameters 1-11
 - examples 1-13
 - locking 1-18
 - re-running 2-11
 - running 2-7
 - sample listing 1-15
 - what it does 1-9

- copying existing metadata 1-6

- copyout utility
 - commands and parameters 1-11
 - examples 1-12
 - locking 1-18
 - re-running 2-10
 - running 2-4
 - sample listing 1-14
 - what it does 1-6

- creating
 - new metadata 1-9
 - symbolic links 2-3
 - target database container 2-6

D

- database container
 - creating 2-6
 - definition 1-3
 - renaming 2-6
 - source 1-3
 - target 1-3

- database directory
 - after copyin 1-10
 - after copyout 1-8
 - definition 1-3
 - disposition of 1-21
 - layout 1-4
 - removing 2-12

- database name
 - lowercase 2-2
 - on copyout and copyin 1-11

- DDL statements 1-18

- definitions
 - active load 1-3
 - active release 1-3
 - database container 1-3
 - database directory 1-3
 - environment variable 1-3
 - exclusive locks 1-18
 - metadata conversion 1-1
 - metadata entry 1-4
 - shared lock 1-18
 - source database container 1-3
 - source database directory 1-3
 - system tablespace 1-4
 - target database container 1-3
 - target database directory 1-3

- df command 2-3

- disk space requirements 1-18

disposition of source database directory 1-21

E

EndTime file 1-9

environment variables

 \$STHDBS 1-5

 \$STHROOT 1-5

 definition 1-3

exclusive locks during copyin 1-18

F

-f parameter 1-12, 2-13

features of metadata conversion 1-2

file names, metadata 1-6

force option (-f) 1-12, 2-15

FTP put command 2-2

H

-h parameter 1-12

help option (-h) 1-12

I

installing software 2-4

L

layout of database directories 1-4

listings

 copyin sample 1-15

 copyout sample 1-14

 to stdout 1-13

ln command 2-3

loads

 aborting 2-2

 active 1-3

 checking the status of 2-2

 confirming 2-2

locking 1-17

locks

 exclusive 1-18

 resource 2-6

 shared 1-18

lowercase database names 2-2

ls command 2-11

M

metadata backup utility 2-2

metadata changes 1-4

metadata conversion

 abandoning 2-15

 definition 1-1

 features 1-2

 locking 1-17

 scheduling 1-19

metadata entry, definition 1-4

metadata file names 1-6

mkdir command 2-6

mv command 2-6

N

-n parameter 1-12

no execution option (-n) 1-12

O

operator account and password 1-19

P

-p parameter 1-12

password option (-p) 1-12

passwords

operator 1-19, 2-4

SYSADM 1-12, 1-19

phases of metadata conversion 1-6

preparing for conversion

backing up metadata 2-2

checking disk space 2-3

checking the status of loads 2-2

completing any active loads 2-2

creating symbolic links 2-3

installing new software 2-4

introduction 2-1

submitting test queries 2-1

procedures

converting from 2.3 to 3.2 2-4

preparing for conversion 2-1

recovering from problems 2-9

restoring previous release 2-15

put command 2-2

R

reboot 1-20

recovering from problems

introduction 2-9

re-running copyin 2-11

re-running copyout 2-9

starting over 2-12

removing target database directory 2-12

renaming source database container 2-6

re-running

copyin 2-11

copyout 2-10

restoring previous release 2-15

rm command 2-12

run meta_bkup command 2-2

running

copyin utility 2-7

copyout utility 2-4

S

SaveConfig file 1-9

scheduling conversions 1-19

security requirements 1-19
shared locks during copyout 1-18
source database container 1-3
source database directory 1-3
standard output device 1-2
starting conversion over 2-12
starting StorHouse software 2-7
startsm command 2-7
status, checking load 2-2
stopping StorHouse software 2-6
stopsm command 2-6
StorHouse Database Administration Guide viii
StorHouse software
 activating 2-6
 starting 2-7
 stopping 2-6
submitting queries 2-1
symbolic links 1-19, 2-3
SYSADM password 1-19
syscreate utility 1-9, 2-7
system requirements
 disk space 1-18
 security 1-19
system tablespace
 definition 1-4

T

target database container 1-3

target database directory 1-3
terminology 1-3
touch command 2-13

V

verifying converted databases 2-8
Version file 1-7, 1-9

W

wildcard character 1-11