



## **StorHouse/RFS Administration Guide**

StorHouse/RFS Release 4.0

Publication Number  
900178 Rev. G

May 24, 2006

---

The FileTek logo consists of the word "FileTek" in white, sans-serif font, centered within a teal square. The square is positioned on the right side of a horizontal line that spans the width of the page.



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.

Copyright © 2002-2006 FileTek, Inc. As an Unpublished Licensed Work.  
Publication Number: 900178 Rev. G

#### NOTICE: U.S. GOVERNMENT USERS

This notice applies to all acquisitions of this work by or for the U.S. Government ("Government"), or by any prime contractor or subcontractor (at any tier) under any contract, cooperative agreement or other activity with the Government. By accepting delivery of this work, the Government agrees that this work and the Licensed Program(s) described herein qualify as "commercial" computer software within the meaning of the acquisition regulation(s) applicable to this procurement. The terms of conditions of the license for the Licensed Program(s) shall pertain to the Government's use and disclosure of this work and the Licensed Program(s), and shall supersede any conflicting contractual terms or conditions. If the license for this work and the Licensed Program(s) fails to meet the Government's need or is inconsistent in any respect with Federal law, the Government agrees to return this work and the Licensed Program(s), unused, to FileTek, Inc. The following additional statement applies only to acquisitions governed by DFARS Subpart 227.4 (October 1988) "Restricted Rights - Use, duplication and disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 (OCT. 1988)." Unpublished licensed work property of FileTek, Inc. Unauthorized use, duplication or distribution prohibited. All rights reserved. A copyright notice on this work and/or on the Licensed Program(s) by itself does not constitute publication or public disclosure of this work or the Licensed Program(s). 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.



# Contents

## Welcome

More about StorHouse/RFS .....	xii
More about StorHouse .....	xii
StorHouse/Control Center .....	xiii
Purpose of this guide .....	xiii
Audience.....	xiv
What’s inside .....	xiv
Recommended reading .....	xv
Related documentation.....	xvi
Document conventions.....	xvi
Before you begin.....	xvii



### Chapter 1: Planning

Planning reference .....	1-2
Data components.....	1-2
StorHouse storage resources.....	1-3
StorHouse database resources.....	1-4
Configuration file definitions.....	1-5
Directories .....	1-6
Developing a storage strategy for a collection set.....	1-7
Selecting the medium for a collection set .....	1-8
Using a set of FSETs for faster writing .....	1-9
Developing a storage strategy for database data .....	1-10
Developing a collection strategy.....	1-13
Defining a collector .....	1-13
Setting collection intervals .....	1-14
Setting a wait time .....	1-15
Setting the interval between StorHouse transfers.....	1-15
Setting the maximum size of a collection.....	1-16
Collection interval example.....	1-17
Managing local space .....	1-17
Managing staging space.....	1-18
Managing collection space .....	1-19
Managing cache space.....	1-21
Setting the maximum cache space .....	1-22
Closing StorHouse connections .....	1-22
Naming StorHouse/RFS resources.....	1-23
Getting started.....	1-25

### Chapter 2: Resource setup

What's set up?.....	2-2
Setup examples .....	2-3
Logging in and out of StorHouse/Admin.....	2-5



Creating a group .....	2-6
Creating a VSET .....	2-8
Creating an FSET .....	2-11
Creating a StorHouse account .....	2-14
Creating a StorHouse database .....	2-18
Creating StorHouse tables .....	2-20
Starting and exiting the StorHouse/RFS configuration file utility .....	2-25
Adding a system definition .....	2-25
Adding a storage definition .....	2-32
Adding a collection definition .....	2-38
Adding a collector definition .....	2-42
Adjusting general parameters .....	2-48

## **Chapter 3: Duplexing setup**

Adding ODBC data sources .....	3-2
Configuring the StorHouse/Control Center server .....	3-2
Configuring StorHouse/Admin .....	3-2
Creating duplicate StorHouse resources .....	3-3
Adding system definitions .....	3-4
Specifying system definition names in other definitions .....	3-6

## **Chapter 4: Security**

Assigning permissions to StorHouse/RFS directories .....	4-2
Managing aliases for user and group names .....	4-3
Creating an aliases database .....	4-3
Creating an aliases table .....	4-4



## Contents

Setting aliases parameters .....	4-5
Creating or displaying an alias.....	4-8
Encrypting passwords .....	4-11

## Chapter 5: Storage management

Backing up StorHouse collections.....	5-2
Backing up StorHouse tables .....	5-5
Deleting a collection set .....	5-7
Recovering StorHouse collections .....	5-11
Recovering StorHouse tables.....	5-12

## Chapter 6: Statistics

About statistics.....	6-2
General statistics .....	6-2
Collection definition statistics .....	6-3
System definition statistics .....	6-5
Ways to store statistics .....	6-5
Local statistics file .....	6-6
Statistics database .....	6-9
Statistics duplexing.....	6-9
Statistics interval .....	6-10
Creating a statistics database .....	6-11
Creating statistics tables .....	6-11
Setting statistics parameters.....	6-13
Displaying local statistics from the previous interval .....	6-18
Displaying the current statistics .....	6-20
Querying statistics in the statistics database.....	6-21



## Chapter 7: Operations

Working with log files .....	7-2
Excluding file types .....	7-4
Rereading the StorHouse/RFS configuration file .....	7-7
Restarting the StorHouse/RFS service .....	7-8
Validating the system .....	7-10
Running StorHouse/RFS in simulation mode .....	7-13
Reporting information for a user file .....	7-15

## Appendix A: The StorHouse/RFS configuration file

File maintenance .....	A-2
File format .....	A-2
Section names .....	A-3
Definition names .....	A-4
Parameters .....	A-4
Required and optional parameters .....	A-5
Omitted parameters or values .....	A-5
Ignored parameters .....	A-5
Dynamic and static parameters .....	A-5
Summary of parameters .....	A-6
File example .....	A-9
RFS section .....	A-13
LocalPath .....	A-13
CacheDir .....	A-13
LogFile .....	A-14
MailServer .....	A-14
MaxCacheSpace .....	A-15
FileCleanupTimeout .....	A-15
MaxSMFiles .....	A-16



## Contents

MaxSMWriters .....	A-16
Flush.....	A-17
CleanupTimeout .....	A-18
Simulate.....	A-18
SafetyPath.....	A-19
Version .....	A-19
STATS section.....	A-20
StatsInterval.....	A-20
FileType .....	A-21
Database .....	A-21
TableName.....	A-22
UserId.....	A-22
Password.....	A-23
PasswordType.....	A-23
SystemName.....	A-23
MirrorName .....	A-24
EXCLUSIONS section.....	A-24
ALIASES section .....	A-24
Database .....	A-25
TableName.....	A-25
UserId.....	A-26
Password.....	A-26
PasswordType.....	A-26
SystemName.....	A-27
MirrorName .....	A-27
COLLECTORS section .....	A-27
System definition.....	A-28
DNSName.....	A-29
STHName.....	A-29
DBDriver.....	A-30
DBPort.....	A-30
DBHost .....	A-31
UserId.....	A-31
Password.....	A-32
PasswordType.....	A-32





Group .....	A-32
VSET .....	A-33
FSET .....	A-33
FSETSegments .....	A-34
RetryInterval .....	A-35
TimeoutOverride .....	A-36
VTF .....	A-36
CheckPoint .....	A-37
MailRecipient .....	A-38
Storage definition .....	A-38
SystemName .....	A-39
MirrorName .....	A-39
Database .....	A-39
TableName .....	A-40
UserId .....	A-40
Password .....	A-41
PasswordType .....	A-41
MaxSearchConnections .....	A-41
SearchConnectionTimeout .....	A-42
Collection definition .....	A-42
Storage .....	A-42
CollectionDir .....	A-43
MaxLoadInterval .....	A-43
MaxWriteSize .....	A-44
Retention .....	A-45
Compression .....	A-46
Browse .....	A-46
Collector definition .....	A-47
StagingDir .....	A-47
UserDir .....	A-48
MaxStagingSpace .....	A-49
MaxCollectionSpace .....	A-50
WaitTime .....	A-50
Group .....	A-51
Permissions .....	A-52
KeepSubdirectories .....	A-53



## **Contents**



## List of figures

Figure 1-1: Collection set on multiple volumes in a VSET.....	1-8
Figure 1-2: Configuration with one StorHouse database .....	1-11
Figure 1-3: Configuration with multiple StorHouse databases .....	1-12
Figure 1-4: Assigning definitions.....	1-14
Figure 1-5: Collection space sizing example .....	1-21
Figure 2-1: Group for the e-mail collection set.....	2-7
Figure 2-2: Group for the compliance collection set.....	2-7
Figure 2-3: VSET for the e-mail collection set .....	2-9
Figure 2-4: VSET for the compliance collection set .....	2-10
Figure 2-5: FSET for the e-mail collection set.....	2-12
Figure 2-6: Set of FSETs for the compliance collection set.....	2-13
Figure 2-7: Account for the e-mail collection set .....	2-15
Figure 2-8: Account for the compliance collection set .....	2-16
Figure 2-9: Database for the e-mail collection set .....	2-19
Figure 2-10: Database for the compliance collection set .....	2-19
Figure 2-11: System definition for the e-mail collection set.....	2-27
Figure 2-12: Storage definition for the e-mail collection set .....	2-34
Figure 2-13: Collection definition for the e-mail collection set.....	2-39



## List of figures

Figure 2-14: Collector definition for the e-mail collection set .....	2-44
Figure 2-15: General parameters for a Windows installation .....	2-49
Figure 3-1: Examples of system definitions for duplexing.....	3-5
Figure 3-2: Example of a storage definition for duplexing .....	3-7
Figure 3-3: Example of a statistics definition for duplexing .....	3-8
Figure 3-4: Example of an aliases definition for duplexing .....	3-9
Figure 4-1: Example of an aliases definition .....	4-6
Figure 4-2: Example of the aliases HTML page.....	4-9
Figure 6-1: Example of a local statistics file in text format .....	6-7
Figure 6-2: Example of a local statistics file in XML format .....	6-8
Figure 6-3: Example of a statistics definition.....	6-15
Figure 6-4: Displaying previous statistics in Windows.....	6-18
Figure 6-5: Displaying current statistics in a Windows environment.....	6-20
Figure 6-6: Column names in the statistics tables.....	6-22
Figure 6-7: Example of a result set containing statistics .....	6-24
Figure 7-1: Example of log files in a log file directory.....	7-3
Figure 7-2: Example of file masks in the exclusions list .....	7-5
Figure 7-3: Sample of a Validation and Health Report.....	7-12
Figure 7-4: Sample user file report in XML format .....	7-18



# Welcome

*StorHouse/Relational File System (RFS)* is the file system interface that enables your applications to store a virtually unlimited number of files on *StorHouse*®—FileTek’s data storage and management system for capturing, storing, moving, and accessing gigabytes (GB) to petabytes of relational and non-relational enterprise data. With StorHouse/RFS, your enterprise can:

- Store any type of *user file*—document, spreadsheet, presentation, e-mail, voice mail, video clip, photo, and so on—on StorHouse
- Retrieve a file and open it with the native application, for instance, open an e-mail message with Microsoft® Outlook or a presentation with PowerPoint®



## **More about StorHouse/RFS**

StorHouse/RFS presents itself to a host as a local or remote volume, or drive. Applications use this *virtual file system* as the storage location for archived items. Files are archived to a folder, or directory, on the virtual file system. StorHouse/RFS collects the files and writes them to StorHouse at user-defined intervals.

When an application or user requests an archived item, and that item is on the virtual file system, StorHouse/RFS locates the file and returns it through the virtual file system. The actual storage location—local, network, or StorHouse—is transparent to the user.

## **More about StorHouse**

StorHouse is the infrastructure behind StorHouse/RFS. StorHouse technology combines industry-leading, scalable storage devices and Open System processors with specialized storage management and relational database management system (RDBMS) software components.

With StorHouse/RFS, applications view StorHouse simply as another file system. However, StorHouse is much more than that. It is a comprehensive system that manages a complete storage hierarchy of the following:

- Redundant array of independent disk (RAID)
- Serial ATA (SATA) RAID
- Massive arrays of idle disks (MAID)
- Write-once-read-many (WORM) and erasable optical disk jukeboxes



- High-performance and high-capacity WORM and erasable tape in automated libraries
- Shelf storage

This highly scalable storage architecture offers a virtually limitless capacity to cost-effectively store data online on different media types, depending on user-dictated access and performance requirements. For example, you can store massive amounts of files that are infrequently accessed on tape media and archive compliance-driven data on WORM tape to support your enterprise's regulatory retention requirements.

## **StorHouse/Control Center**

*StorHouse/Control Center* is FileTek's Windows®-based application for StorHouse system and database administration. It provides an easy-to-use graphical user interface that simplifies StorHouse storage and database management tasks. You use StorHouse/Control Center, specifically the *StorHouse/Admin module*, to set up the resources necessary to store application data on StorHouse.

## **Purpose of this guide**

This book explains how to set up the resources to store application data on StorHouse through StorHouse/RFS. This guide also contains planning information, and it describes storage management and other administrative tasks you may perform while StorHouse/RFS is operational.



## Welcome

This manual includes procedures for using the StorHouse/RFS configuration file utility (in a Windows environment) and a text editor in a UNIX environment.

## Audience

This guide has been written for the person responsible for managing StorHouse/RFS or for defining how application data is stored on StorHouse. This guide assumes you have read the publications listed in the “Recommended reading” section on page xv.

## What’s inside

This manual consists of the following chapters:

- Chapter 1, “Planning,” contains information to help you define a data collection and storage strategy.
- Chapter 2, “Resource setup,” explains how to implement the storage of application data on StorHouse. This chapter contains examples for an e-mail archiving application and a compliance document archive.
- Chapter 3, “Duplexing setup,” summarizes the setup procedures required to implement StorHouse/RFS duplexing to store data on two StorHouse systems.
- Chapter 4, “Security,” identifies the directories that require permissions, explains how to encrypt passwords in a UNIX environment, and explains how to manage user and group name changes.





- Chapter 5, “Storage management,” describes how to back up, recover, and remove data on StorHouse.
- Chapter 6, “Statistics,” describes the statistics that StorHouse/RFS can generate and explains your options for storing and displaying statistics.
- Chapter 7, “Operations,” describes some of the daily or periodic tasks you may perform while StorHouse/RFS is running.
- Appendix A, “The StorHouse/RFS configuration file,” describes the operating parameters that you configure for StorHouse/RFS.

## Recommended reading

The following publications contain information that can help you with StorHouse/RFS administration. This guide assumes you have read these publications.

- The *StorHouse/RFS Concepts*, publication number 900159, defines the StorHouse/RFS hardware and software components, storage and retrieval processes, security features, and StorHouse/RFS terminology, such as collection, virtual file system, and staging area.
- The *StorHouse Concepts and Facilities Manual*, publication number 900026, presents an overview of StorHouse and describes how data is allocated, stored, and managed.
- The *Getting Started with StorHouse/Admin*, publication number 900135, explains how to perform StorHouse administration tasks with StorHouse/Admin. The document contains an introduction to the software features, windows, menus, tool bars, and selected system and database management tasks.



## Related documentation

The following StorHouse/Control Center publications describe how to use StorHouse/Admin to complete StorHouse system and database administration tasks.

- The *StorHouse/Admin System Administrator's Quick Reference*, publication number 900147, contains procedures for performing system administration tasks with StorHouse/Admin. Some of the tasks include setting system parameters, backing up data, and creating groups, accounts, volume sets, and file sets.
- The *StorHouse/Admin Database Administrator's Quick Reference*, publication number 900148, contains procedures for performing database administration tasks with StorHouse/Admin. Some of the tasks include creating databases, using Interactive SQL (ISQL), and backing up metadata.

## Document conventions

The following conventions are used in this book.

**Table i: Document conventions**

Convention	Meaning
<i>Italics</i>	Emphasized text, new terms, and document titles
<b>Bold</b>	Control names for boxes, check boxes, lists, and options
This font	Data that you enter
This font	StorHouse/RFS configuration file formats and values, UNIX commands, report examples
▼	Procedures



## Before you begin

This guide assumes your StorHouse/RFS system is installed. Refer to the applicable StorHouse/RFS installation manual for instructions on installing StorHouse/RFS.

- *StorHouse/RFS Installation Manual for Windows*, publication number 900174
- *StorHouse/RFS Installation Manual for Solaris*, publication number 900175
- *StorHouse/RFS Installation Manual for AIX*, publication number 900183

This guide also assumes your StorHouse/Control Center system is installed and running. Refer to *Getting Started with StorHouse/Control Center*, publication number 900138, for complete installation information.



**Welcome**

# C H A P T E R

# 1



## **Planning**

This chapter contains information to help you plan a strategy for collecting data and storing it on StorHouse. This chapter also describes space management considerations, and it defines the concepts that are helpful to understand before setting up StorHouse/RFS resources.



# Planning reference

This section defines the StorHouse/RFS resources and terms used in this guide.

## Data components

The types of data that StorHouse/RFS manages or generates are as follows.

- A *user file*, or object, or record, is the data that StorHouse/RFS collects and manages through the virtual file system.
- A *load file*, or *.ldr file*, contains file locator data for each user file. *File locator data* is information about each user file. StorHouse/RFS generates file locator data when a file is renamed during the collection process and uses it to locate and retrieve user files in StorHouse collections.
- A *local collection* is a group of user files and corresponding file locator data that have not yet been written to StorHouse.
- A *StorHouse collection* is a group of user files and corresponding file locator data that have been written to StorHouse.
- A *collection set* is a group of StorHouse collections. For example, if an e-mail collection is written to StorHouse once a day, then after a month, the e-mail collection set would consist of 30 or 31 StorHouse collections, each consisting of data and associated file locator data.
- *Collection metadata* identifies the StorHouse files that have been written to StorHouse. StorHouse/RFS generates collection metadata after a StorHouse collection is successfully written.



- *Security entries* are operating system security descriptors for Windows environments. StorHouse/RFS captures and immediately stores security information when a directory's security is created or updated. These security entries are not used in UNIX installations.
- *Aliases* are correlations between old and new user and group names. Their purpose is to enable access to user files using either the old or new name.
- *Statistics* and *logs* are information for monitoring StorHouse/RFS activity and usage.

## StorHouse storage resources

StorHouse/RFS uses the following resources to write StorHouse collections. You use StorHouse/Admin to create and maintain these storage resources.

- A *group*, also called *file access group*, is a set of files stored together for access control. All StorHouse collections belong to a group.
- A *volume set (VSET)* is one or more physical *volumes* (for example, tape cartridges) that are treated as a logical unit of storage. VSETs enable you to control the physical grouping of StorHouse collections.
- A *file set (FSET)* is an area of storage within a VSET. Files are stored in FSETs. A StorHouse collection is a file.
- A *StorHouse account* is a set of privileges and attributes used for access control. This account must have the required privileges to write StorHouse collections and to own, load, and query StorHouse tables.



### StorHouse database resources

StorHouse/RFS uses the following resources to store file locator data, collection metadata, security entries, aliases, and statistics. You use StorHouse/Admin and the StorHouse/RFS tblgen utility to create and maintain these database resources.

- A *StorHouse database* contains StorHouse tables used by StorHouse/RFS. Each database has a *system tablespace* created automatically when you create a database. The tables used by StorHouse/RFS are stored in the system tablespace.
- A *StorHouse table* contains data used by StorHouse/RFS. When you run the tblgen utility to create StorHouse tables, you supply the owner and base table name, and StorHouse/RFS adds a suffix to the name as described below.
  - Collection metadata is stored in a *collections table*. This table name ends in *\_C*.
  - File locator data is stored in a set of StorHouse tables called a *table array*. After you create a collections table, StorHouse/RFS automatically creates an initial set of 32 *file locator tables* and creates additional sets of tables when needed. These table names end in *\_0*, *\_1*, *\_2*, and so on.
  - Security entries are stored in a *security table*. This table name ends in *\_S*.
  - Aliases are stored in an *aliases table*. This table name ends in *\_ALIASES*.
  - Statistics are stored in a set of three *statistics tables*, with names ending in *\_RFS*, *\_TABLES*, and *SYSTEMS*.





## Configuration file definitions

The StorHouse/RFS configuration file contains definitions that you use to implement a collection and storage strategy. In a Windows environment, you use the StorHouse/RFS configuration file utility to create and maintain definitions. In a UNIX environment, you use a text editor to work with definitions.

- A *system definition* identifies a StorHouse system for storing StorHouse collections, file locator data and collection metadata, statistics, and aliases. For StorHouse collections, a system definition specifies the logical organization (group, VSET, and FSET) of data on StorHouse. You create one system definition for each separate group/VSET/FSET to contain StorHouse collections.
- A *storage definition* specifies parameters for managing file locator data and collection metadata. You associate a storage definition with a collection definition. Multiple collection definitions can use the same storage definition.
- A *collection definition* defines collection set options, such as retention requirements, whether users will be able to browse a collection set, and the maximum load interval and maximum write size. You assign a collection definition to a collector definition. Multiple collector definitions can use the same collection definition.
- A *collector definition* identifies a collector's staging area (staging directory and user directory), specifies security permissions, and contains parameters for managing staging and collection space.
- An *aliases definition* contains all of the parameters required to use aliases.
- A *statistics definition* contains all of the parameters required to capture and store statistics.



### Directories

StorHouse/RFS uses the following directories, all located on or accessible to the StorHouse/RFS server platform.

- A *staging directory* is where StorHouse/RFS writes physical files on behalf of applications. You identify a staging directory and set the maximum amount of staging space in a collector definition.
- A *user directory* is a subdirectory under a staging directory. You identify a user directory in a collector definition. Together, the staging directory and the user directory comprise a *staging area*.
- A *rename directory* is where StorHouse/RFS renames user files found in staging area during the collection process. StorHouse/RFS automatically creates rename directories.
- A *collection directory* is where StorHouse/RFS creates load files containing file locator data. You identify the collection directory and set the maximum collection space in a collection definition.
- A *cache directory* is where StorHouse/RFS places data that it reads from StorHouse. You identify the cache directory and set the maximum cache space in the RFS, or General, section of the StorHouse/RFS configuration file.
- A *safety directory* is where StorHouse/RFS writes secondary copies of user file updates (for instance, file deletes, file renames, and security changes) that are not associated with the current local collection. StorHouse/RFS uses these safety copies in the event a load file becomes corrupt. You identify the safety directory in the RFS, or General, section of the StorHouse/RFS configuration file.
- A *local directory* contains a copy of statistics generated during a previous statistics interval and other temporary files used by



StorHouse/RFS. You identify the local directory in the RFS, or General, section of the StorHouse/RFS configuration file.

- A *log directory* contains all of the log files generated by StorHouse/RFS. You identify the log directory in the RFS, or General, section of the StorHouse/RFS configuration file.

## Developing a storage strategy for a collection set

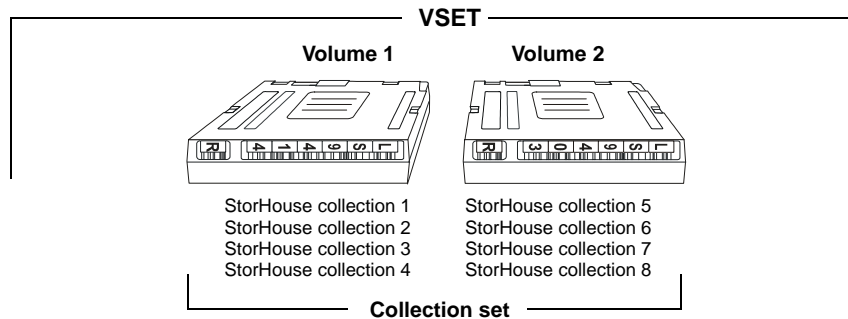
One of your first planning tasks is to determine how to organize your user data on StorHouse. You define the following storage resources for each collection set. For StorHouse/RFS duplexing, you create identical resources on both StorHouse systems.

- Group
- VSET
- FSET
- System definition

On StorHouse, each collection set is stored on multiple *volumes* (or physical media, for instance, tape cartridges) that are treated as a logical unit.



Figure 1-1 illustrates a collection set on multiple volumes in a VSET.



**Figure 1-1: Collection set on multiple volumes in a VSET**

Factors that affect a storage strategy include:

- The storage medium to use for a collection set
- The number of collections written to StorHouse at a time

### Selecting the medium for a collection set

When you create a VSET, you select the library device and storage medium. StorHouse supports the following storage media:

- Redundant array of independent disk (RAID)
- Serial ATA (SATA) RAID
- Massive Arrays of Idle Disks (MAID)
- Write-once-read-many (WORM) and erasable optical disks in jukeboxes
- High-performance and high-capacity (WORM and erasable) tape in automated libraries



StorHouse also supports shelf storage. The cost and performance of each medium and your site's data retrieval requirements are the primary considerations for selecting the appropriate mix of media for your StorHouse system and the medium for a particular collection set. For instance, magnetic disk supports multiple concurrent reads and writes to the same volume and does not present access conflicts. This medium may be a good choice for a collection set with frequently accessed data. On the other hand, high volume data that's less frequently accessed may be best stored on tape or if it's compliance-driven, WORM tape. Your FileTek customer support representative can help you make storage media decisions.

### Using a set of FSETs for faster writing

When StorHouse/RFS writes a collection to StorHouse, it typically writes the data to the StorHouse *performance buffer* first and then to the primary, or resident, FSET. The *VTF* parameter in a system definition determines whether StorHouse/RFS:

- Bypasses the performance buffer and writes directly to the FSET
- Copies the data to the FSET immediately after writing to the performance buffer
- Copies the data to the FSET later during the next StorHouse *write-back operation* (StorHouse BACKUP command)

If you expect StorHouse/RFS to write multiple StorHouse collections at a time to the same VSET, then you can improve write performance by creating a *set of FSETs*. Using a set of FSETs improves the write speed from the performance buffer to tape or directly to the primary FSET. A set can contain any number of FSETs. StorHouse/RFS writes one StorHouse collection to the first FSET, the second StorHouse collection to the second FSET, and so on.



## Chapter 1 - Planning

To implement a set of FSETs, you must:

- Create the FSETs on StorHouse. The FSET names must end in a number, for example, COMPLY0, COMPLY1, and COMPLY2. The first FSET number in a set must be 0.
- Specify an FSET name for the FSET parameter in the system definition. You just specify the base FSET name (not the number). For example, if the FSET names are COMPLY0, COMPLY1, and COMPLY2, then you would specify COMPLY at the FSET parameter in the system definition.
- Specify a value for the FSETSegments parameter in the system definition. For example, if there are three FSETs in your set, you would specify 3 for the FSETSegments parameter.

Refer to the *StorHouse System Administrator's Guide* or the *StorHouse Concepts and Facilities Manual* for more information about the StorHouse performance buffer, write-back operation, and VTF parameter. See page 2-13 for an example set of FSETs and page 2-27 for an example system definition that implements the set of FSETs.

## Developing a storage strategy for database data

Another planning task is to determine a strategy for storing the following StorHouse/RFS data on StorHouse.

- File locator data in a table array
- Collection metadata in a collections table
- Security entries (if applicable) in a security table

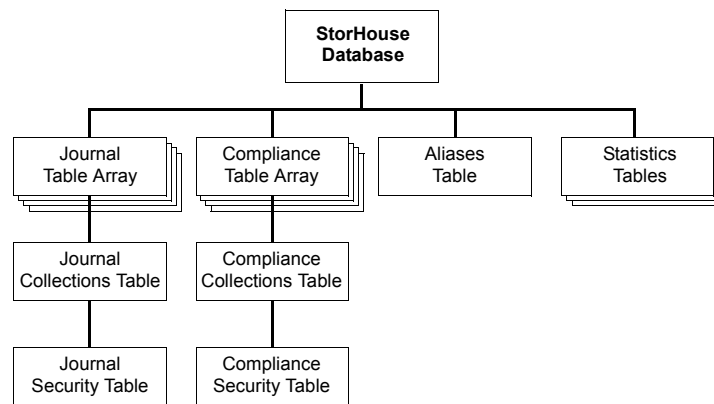


- Aliases (if required) in an aliases table
- Statistics (if desired) in the statistics tables

You define the following resources to store this data:

- StorHouse database
- StorHouse tables
- Storage definition
- Aliases definition (if required)
- Statistics definition (if desired)

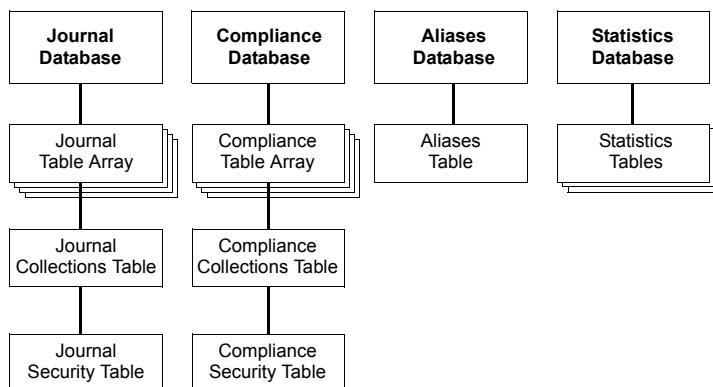
You can create a single StorHouse database and then create all tables for all collection sets in that database. Or you can create a separate StorHouse database and set of tables for each collection set. For example, Figure 1-2 illustrates a single StorHouse database containing StorHouse tables for two collection sets (journal and compliance), aliases, and statistics.



**Figure 1-2: Configuration with one StorHouse database**



Figure 1-3 illustrates a configuration with a separate StorHouse database for each collection set and for aliases and statistics.



**Figure 1-3: Configuration with multiple StorHouse databases**

Although you can create an unlimited number of StorHouse databases, only 100 databases can be accessed concurrently. If you plan to store hundreds of collection sets on StorHouse and expect them to be open concurrently for reading and writing, consider creating a fewer number of databases.

A storage definition in the StorHouse/RFS configuration file identifies the StorHouse database and the base table name of the table array and collections table. The storage definition also identifies the system definition used to store corresponding StorHouse collections. Likewise, the aliases definition and statistics definition identify the StorHouse database and the base table name of the aliases table and the statistics tables.





## Developing a collection strategy

After you have defined a strategy for storing user data and supporting database data on StorHouse, the next task is to develop a strategy for collecting the data. You define a collection strategy for each type of application data you want to store on StorHouse. For example, if you plan to store e-mail messages and compliance data on StorHouse, you can define one collection strategy for e-mail and another strategy for compliance data.

A collection strategy includes:

- Defining a collector
- Setting collection intervals

### Defining a collector

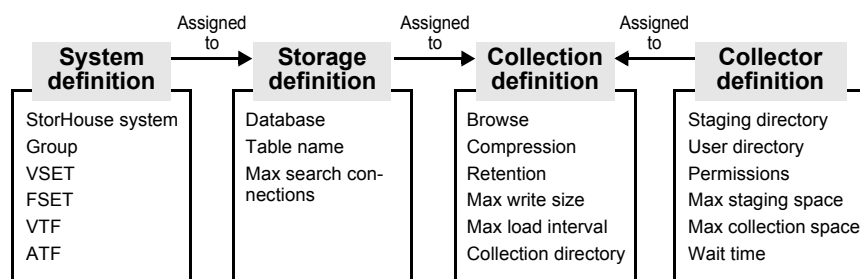
You define a collector in a StorHouse/RFS configuration file by:

- Adding a collector definition in the StorHouse/RFS configuration file
- Assigning a collector to a collection definition

In a collector definition, you specify where the collector looks for files to collect (which staging directory and user directory). Users and applications write user files to this location. The corresponding collection definition specifies collection set options such as retention requirements, whether the StorHouse collection will be compressed, and whether users may browse collections. The collection definition also identifies the storage definition which specifies the StorHouse database used to store the corresponding file locator data and collection metadata and the system definition used to store StorHouse collections.



In summary, a system definition is assigned to a storage definition, a storage definition is assigned to collection definition, and a collector definition is assigned to a collection definition. Figure 1-4 illustrates definition assignments and the key information in each definition.



**Figure 1-4: Assigning definitions**

You can define one collector for each application, one collector for multiple applications, or multiple collectors for the same application data. For example, you can define two collectors to collect videos from two different locations and store all videos in the same collection. In this case, you would create two collector definitions and assign them to the same collection definition. See “Adding a collector definition” on page 2-42 for more examples.

## Setting collection intervals

A *collection interval* defines the movement of data from a staging area to a rename directory and finally to StorHouse. The purpose of configuring intervals is to transfer data from local storage to StorHouse in a timely



manner. This ensures there is always enough local disk space to continue collecting more user files.

Some factors to consider when determining collection intervals are:

- The length of time a file should wait in a staging area to become eligible for collection (*WaitTime* parameter in a collector definition)
- The maximum time between transfers and writes to StorHouse (*MaxLoadInterval* parameter in a collection definition)
- The maximum size you want a collection to grow (*MaxWriteSize* parameter in a collection definition)

### Setting a wait time

The *WaitTime* parameter in a collector definition specifies the number of minutes a file must be idle in a staging area before it is eligible for collection. This parameter prevents files that are in the process of being written to the virtual file system from being prematurely collected. You set a *WaitTime* value for each collector. You need a reasonable amount of time to ensure that a file is completely written. For pure archiving (when users aren't opening or changing files after moving or copying them to the virtual file system), consider a one- or two-minute wait time.

### Setting the interval between StorHouse transfers

The *MaxLoadInterval* parameter in a collection definition specifies the number of minutes between transfers to StorHouse for each local collection. You define this interval in each collection definition. The minimum value is 30 minutes. The interval starts when a collector collects, or renames, the first file in the current collection. Therefore, this interval is the maximum amount of time the first file will be in a local collection before StorHouse/RFS writes the collection to StorHouse. For instance, if the maximum load interval is 60 minutes and StorHouse/RFS



collects the first file 20 minutes after the last collection closed, then StorHouse/RFS closes the current collection and transfers data from the rename and collection directories to StorHouse 80 minutes after the last collection closed.

### Setting the maximum size of a collection

The *MaxWriteSize* parameter in a collection definition specifies the maximum number of megabytes (MB) a local collection can contain before a collector closes it, that is, no longer renames files to the local collection. The minimum value is 100 MB and the maximum value is 2 GB.

You define the *MaxWriteSize* in each collection definition. Larger collections are more efficient but require more local disk space. If you want to (or need to) limit the amount of local disk on the StorHouse/RFS server platform, then writing smaller collections more frequently will make the most amount of storage available locally.

For performance, the maximum number of files in a collection is 200,000. If a collection does not exceed the *MaxWriteSize* but contains more than 200,000 files, StorHouse/RFS closes the current collection and starts a new one.

Although the maximum collection size is 2 GB, StorHouse/RFS can collect a single user file up to 10 terabytes (TB) in size. StorHouse/RFS places a file that is larger than the maximum write size into its own collection. For example, StorHouse/RFS renames a 100-GB file as a single local collection and then a single collection to StorHouse. The collection contains one file (and its associated file locator data). Files that are smaller than the maximum write size are collected normally.



### Collection interval example

Assume the following settings:

- The WaitTime in a collector definition is 1 minute
- The MaxLoadInterval in a collection definition is 720 minutes (twice a day)
- The MaxWriteSize in a collection definition is 1 GB

While StorHouse/RFS checks for completed local collections every 60 seconds, the collector:

- Checks for files in the staging area (staging directory and user directory)
- Accumulates eligible files (those that have not been modified for 60 seconds) to the local collection for 720 minutes or until the local collection reaches 1 GB

If the collector collects 4 GB of data in a day, then StorHouse/RFS creates 4 collections that day (because the MaxWriteSize is 1 GB). If the collector collects less than 1 GB a day, then StorHouse/RFS may create 2 collections that day (because the MaxLoadInterval is twice a day), depending on when StorHouse/RFS collects the first file in a collection.

## Managing local space

Space is required for storing data in staging areas, in rename and collection directories, and in the cache directory. You identify the staging areas, collection directories, and cache directory in the StorHouse/RFS configuration file. StorHouse/RFS automatically creates and manages the rename directories. These directories can be located on the



StorHouse/RFS server or on any machine accessible to it. You can manage local space to accommodate collection activity and to enhance retrieval performance. In addition, you can monitor local space usage by checking statistics, as described in Chapter 6, “Statistics.”

### Managing staging space

After StorHouse/RFS writes a file to a staging area, the file waits in the staging area until after the `WaitTime` (specified in a collector definition) elapses. Each collector runs for two minutes before stopping to flush the local collection file buffers and to purge files just collected from a staging area. Files are not removed from the staging area until after they are safely flushed with the collection. Therefore, the longest time a file may actually wait in a staging area is the `WaitTime` plus 2 minutes for the collector cycle.

You can set the maximum amount of storage (in MB) for each staging area. This parameter, in a collector definition, is called *MaxStagingSpace*. When the maximum amount is reached, StorHouse/RFS returns an out of space condition appropriate for the operating system. Users and applications may write files to the virtual file system when space permits.

Factors that affect how you set the `MaxStagingSpace` parameter include:

- The size and number of files written to the virtual file system
- The length of time files wait (`WaitTime`) in the staging area before they are eligible for collection
- The space available on the computer(s) with the designated staging areas
- The speed of the disk with the designated staging areas (for instance, if the `WaitTime` is 1 minute (60 seconds) and the disk write speed is



20 MB per second, then 1200 MB plus 2 minutes more worth of disk space is required when the system is running at peak)

For instance, assume the following:

- Two applications write data to the virtual file system.
- You've defined two collectors to check for files in two staging areas that are located on the same machine.
- The WaitTime for both collectors is 1 minute.
- In a 60-second period, one application typically writes 2 MB of data to the virtual file system and the second application writes 10 MB of data to the virtual file system.

In this example, in a 180-second period (WaitTime of 1 minute plus a collector cycle time of 2 minutes), one staging area may hold 6 MB of data and the other staging area may hold 30 MB of data. You could set the MaxStagingSpace parameter to 6 MB and 30 MB in the respective collector definitions. If you continually run out of staging space, you could increase the MaxStagingSpace parameter, decrease the WaitTime parameter in the collector definition(s) for one or both collectors, or add staging disk space. If you add staging disk space, then you should also increase the MaxStagingSpace value in the collector definitions.

## **Managing collection space**

A collector accumulates user files and file locator data to a local collection in a rename directory and collection directory until one of the following occurs:

- The MaxLoadInterval specified in a collection definition has elapsed
- The MaxWriteSize specified in a collection definition is exceeded
- The .ldr file contains more than 200,000 entries



## Chapter 1 - Planning

StorHouse/RFS can keep local collections after they have been written to StorHouse. The advantage to retaining local collections is that they are readily available during a local search. StorHouse/RFS performs a StorHouse search only when collections are not available locally.

StorHouse/RFS automatically creates rename directories on the same disk and directory level as the staging directory. You specify the collection directory location with the `CollectionDir` parameter in each collection definition. You can set the *MaxCollectionSpace* parameter in a collector definition to determine the maximum amount of space (in MB) for local collections, that is, for the renamed files in the rename directory and the file locator data in the collection directory.

Collection space includes the current data being collected as well as old local collections (space permitting). When the maximum amount is reached, StorHouse/RFS deletes the oldest local collection to make room for new ones. Because you set a `MaxCollectionSpace` for each collector definition, you can reserve a larger amount of space for local collection data that requires fast retrieval and a smaller amount of space for collection data that doesn't require fast retrieval.

Factors that affect how you set the `MaxCollectionSpace` parameter in a collector definition include:

- The number of local collections created and retained and the size of those local collections
- The space available on the machine(s) with the designated collection directories

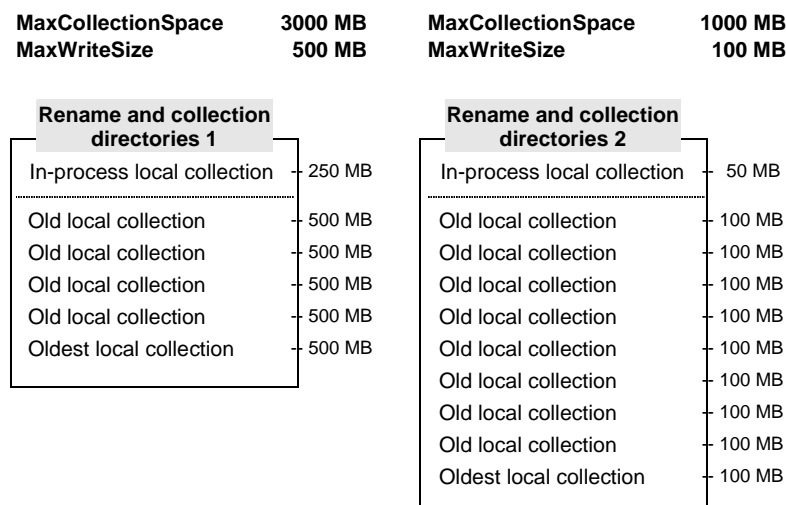
Here's an example. Assume you are using StorHouse/RFS for two different collection sets. You've designated two staging areas and collection directories on the StorHouse/RFS server and have 4 GB of space you can use on the server.





- The first collection set has a MaxWriteSize of 500 MB and a MaxCollectionSpace of 3000 MB.
- The second collection set has a MaxWriteSize of 100 MB and a MaxCollectionSpace of 1000 MB.

Assuming the MaxWriteSize always occurs before the MaxLoadInterval expires, the rename and collection directories for the first collection set can hold up to 6 local collections and the rename and collection directories for the second collection set can hold up to 10 local collections at a time.



**Figure 1-5: Collection space sizing example**

## Managing cache space

StorHouse/RFS stores data retrieved from StorHouse in the cache directory. You specify the location of the cache directory with the CacheDir parameter in the General, or RFS, section of the StorHouse/RFS configuration file. When an application requests to read a



file that is in a StorHouse collection and not on local storage, StorHouse/RFS retrieves the data from StorHouse and places it in a cache file in the cache directory. StorHouse/RFS reads data from the cache directory and delivers it to the virtual file system for the application. You set two parameters in the StorHouse/RFS configuration file that control the maximum cache space and when to close StorHouse connections.

### Setting the maximum cache space

You can set the maximum amount of space (in MB) to be used for caching data retrieved from StorHouse. This parameter, in the General, or RFS, section is called *MaxCacheSpace*. The minimum value is 100 MB and the maximum value is 1000 MB. Cache uses a least recently used/most recently used scheme to manage data. When space is needed to cache a record read from StorHouse, StorHouse/RFS replaces the least recently used with the new one.

Factors that affect how you set the MaxCacheSpace parameter include:

- The average user file size
- The number of files typically retrieved from StorHouse and open at a time
- The amount of space available on the machine with the designated cache directory

For example, assume that the average user file size is 5 MB and you estimate that 50 files are typically retrieved from StorHouse at a time. A MaxCacheSpace of at least 250 MB could accommodate this activity.

### Closing StorHouse connections

You can set the interval (in minutes) to be used for closing StorHouse connections. This parameter, in the General, or RFS, section is called



*CleanupTimeout.* After the interval expires, StorHouse/RFS checks for any cache files that have not been used since the last cleanup interval and closes the StorHouse connections for those files.

Note that *MaxSMFiles*, another parameter in the General, or RFS, section, controls the maximum number of StorHouse connections that can be open for reading. One StorHouse connection is used for reading any number of user files in a StorHouse collection, that is, one connection per StorHouse collection. If the number is reached and a StorHouse connection is needed, then StorHouse/RFS force closes the least recently used connection even if the *CleanupTimeout* has not expired.

# Naming StorHouse/RFS resources

Another planning task is to develop a scheme for naming StorHouse/RFS resources. The following table summarizes the naming conventions.

Table 1-1: Resource naming conventions

Structure	Format
Group	1 to 8 characters including A-Z (uppercase), 0-9, _ (underscore), and dollar sign). Examples: COMPLY and JOURNAL
VSET	1 to 8 characters including A-Z (uppercase), 0-9, _ (underscore), and \$ (dollar sign). Examples: COMPLY and JJAN02
FSET	1 to 8 characters including A-Z (uppercase), 0-9, _ (underscore), and \$ (dollar sign). The FSET name must be unique within a VSET but it can be the same name as the VSET. Examples: COMPLY and JJAN02



**Table 1-1: Resource naming conventions (continued)**

Structure	Format
Set of FSETs	<p>Same conventions as FSET except the name must end in a number starting with 0. The set name must therefore have seven or fewer characters.</p> <p>Examples: COMPLY0 and COMPLY1</p>
StorHouse account	<p>1 to 12 characters including A-Z (uppercase), 0-9, \$ (dollar sign), and _ (underscore).</p> <p>Examples: CACCOUNT and JACCOUNT</p>
Account password	<p>1 to 32 characters including A-Z (uppercase), 0-9, \$ (dollar sign), and _ (underscore).</p> <p>Examples: CPASSWORD and JPASSWORD</p>
StorHouse database	<p>1 to 32 contiguous characters (no blanks) including a-z (lowercase), A-Z (uppercase), 0-9, and _ (underscore). The database name must start with a letter, is case sensitive, and must be unique in a StorHouse system. Use descriptive names that include the business entity or geographical location.</p> <p>Examples: COMPLYDATABASE and JOURNALDATABASE</p>
StorHouse table	<p>Two-part name in the format:</p> <p>owner.base_table_name</p> <p>The combination of owner and base_table_name must be unique in a StorHouse database. The owner is a StorHouse account name and the base_table_name must start with a letter, cannot be a StorHouse SQL reserved word, and can consist of 1 to 32 contiguous characters (no blanks) including a-z (lowercase), A-Z (uppercase), 0-9, and _ (underscore). StorHouse/RFS adds a suffix to the base table name as described on page 1-4.</p> <p>Examples: CACCOUNT.COMPLYTABLE and JACCOUNT.JOURNALTABLE</p>
System definition	<p>1 to 20 characters (no maximum length when updating the StorHouse/RFS configuration file with a text editor) including a-z (lowercase), A-Z (uppercase), 0-9, and _ (underscore). This name is arbitrary but must be unique in a StorHouse/RFS configuration file.</p> <p>Examples: ComplySet and JournalSet</p>

**Table 1-1: Resource naming conventions (continued)**

Structure	Format
Storage definition	<p>1 to 20 characters (no maximum length when updating the StorHouse/RFS configuration file with a text editor) including a-z (lowercase), A-Z (uppercase), 0-9, and _ (underscore). This name is arbitrary but must be unique in a StorHouse/RFS configuration file.</p> <p>Examples: ComplyTable and JournalTable</p>
Collection definition	<p>1 to 20 characters (no maximum length when updating the StorHouse/RFS configuration file with a text editor) including a-z (lowercase), A-Z (uppercase), 0-9, and _ (underscore). This name is arbitrary but must be unique in a StorHouse/RFS configuration file. The collection definition name is the prefix in the StorHouse file name for each StorHouse collection and should be unique for each StorHouse/RFS server. The collection definition name is also the prefix for renamed files in rename directories.</p> <p>Examples: ComplyCollection and JournalCollection</p>
Collector definition	<p>1 to 20 characters (no maximum length when updating the StorHouse/RFS configuration file with a text editor) including a-z (lowercase), A-Z (uppercase), 0-9, and _ (underscore). This name is arbitrary but must be unique in a StorHouse/RFS configuration file.</p> <p>Examples: ComplyCollector and JournalCollector</p>

## Getting started

If you are just getting started with StorHouse/RFS, complete the following initial administration tasks.

**Table 1-2: Resource setup tasks**

Task	See
Create StorHouse storage resources	Chapter 2
Create StorHouse database resources	Chapter 2



## Chapter 1 - Planning

**Table 1-2: Resource setup tasks (continued)**

Task	See
Add definitions to the StorHouse/RFS configuration file	Chapter 2
Set up a StorHouse/RFS duplexing environment (if applicable)	Chapter 3
Assign permissions to StorHouse/RFS directories	Chapter 4
Prepare for user or group name changes (aliases)	Chapter 4
Develop a backup plan for StorHouse collections	Chapter 5
Develop a backup plan for StorHouse databases	Chapter 5
Define requirements and resources for capturing statistics	Chapter 6
Identify file types to be excluded from searches and collections	Chapter 7

## C H A P T E R   **2**



# **Resource setup**

This chapter describes the procedures required to set up the storage and database resources for application data on StorHouse. Perform these procedures, in the listed sequence, each time you want to store a new type of application data, or collection set, on StorHouse.



# What's set up?

To store application data on StorHouse, you need to:

- Create the StorHouse storage resources used to store StorHouse collections
- Create the StorHouse database resources used to store database data
- Define the storage and collection strategy in StorHouse/RFS configuration file definitions

Table 2-1 summarizes the setup procedures and lists the interfaces you use to perform them.

**Table 2-1: Resource setup procedures and interfaces**

Procedure	Interface
Create a group	StorHouse/Admin
Create a VSET	StorHouse/Admin
Create an FSET	StorHouse/Admin
Create a StorHouse account	StorHouse/Admin
Create a StorHouse database	StorHouse/Admin
Create StorHouse tables	StorHouse/RFS tblgen utility
Add a system definition	StorHouse/RFS configuration file utility or text editor
Add a storage definition	StorHouse/RFS configuration file utility or text editor
Add a collection definition	StorHouse/RFS configuration file utility or text editor



**Table 2-1: Resource setup procedures and interfaces (cont.)**

Procedure	Interface
Add a collector definition	StorHouse/RFS configuration file utility or text editor
Adjust general StorHouse/RFS parameters	StorHouse/RFS configuration file utility or text editor

For StorHouse/RFS duplexing, you must create the same group, VSET, FSET, account, database, and tables on the secondary StorHouse system. All resource names must be the same. See Chapter 3, “Duplexing setup,” for more information about implementing StorHouse/RFS duplexing.

## Setup examples

This chapter contains examples for a Windows-based e-mail archive and a UNIX-based compliance archive. Table 2-2 summarizes the characteristics of the e-mail archive.

**Table 2-2: E-mail archive characteristics**

Characteristic	
Approximate number of e-mail files a day	100,000
Approximate size of each e-mail file	100 KB
Number of collections created each day	6
Virtual file system location	V:\Journal
Media selection	WORM tape



## Chapter 2 - Resource setup

Table 2-3 lists the names assigned to the resources for the e-mail archive

**Table 2-3: E-mail archive resource names**

Resource	Name
Group	JOURNAL
VSET	JJAN02
FSET	JJAN02
StorHouse account	JACCOUNT
Account password	JPASSWORD
StorHouse database	JOURNALDATABASE
StorHouse table	JACCOUNT.JOURNALTABLE
System definition	JournalSet
Storage definition	JournalTable
Collection definition	JournalCollection
Collector definition	JournalCollector
Directory name on virtual file system	Journal

Table 2-4 summarizes the characteristics of the compliance archive.

**Table 2-4: Compliance archive characteristics**

Characteristic	
Approximate number of files a day	50,000
Approximate size of each file	1 MB
Number of collections created each day	125
Virtual file system location	All staging directories on the mount point serv01:/rfs
Media selection	WORM tape



Table 2-5 lists the names assigned to the resources for the compliance archive.

**Table 2-5: Compliance archive resource names**

Resource	Name
Group	COMPLY
VSET	COMPLY
FSETs	COMPLY0 and COMPLY1
StorHouse account	CACCOUNT
Account password	CPASSWORD
StorHouse database	COMPLYDATABASE
StorHouse tables	CACCOUNT.COMPLYTABLE
System definition	ComplianceSet
Storage definition	ComplianceTable
Collection definition	ComplianceCollection
Collector definition	ComplianceCollector

## Logging in and out of StorHouse/Admin

You use StorHouse/Admin to perform StorHouse administration tasks, such as creating groups, VSETs, FSETs, StorHouse accounts, and StorHouse databases. At the computer with the installed StorHouse/Admin software, access StorHouse/Admin with the SYSADM account and password. The SYSADM account has the required privileges to perform the setup procedures.



## Chapter 2 - Resource setup

### ▼ To log in to StorHouse/Admin

1. On the desktop, double-click the **StorHouse/Control Center** shortcut.
2. In the selection bar, click the **StorHouse/Admin** shortcut.

If your site has multiple StorHouse systems and/or StorHouse/Control Center servers, then the selection bar may contain multiple shortcuts. Click the appropriate shortcut for the StorHouse system you want to access.

3. In the **Login** dialog box, type SYSADM in the **Account ID** box and the associated password in the **Password** box.
4. Click **Login**.

The StorHouse Resources window is displayed.

### ▼ To log out of StorHouse/Admin

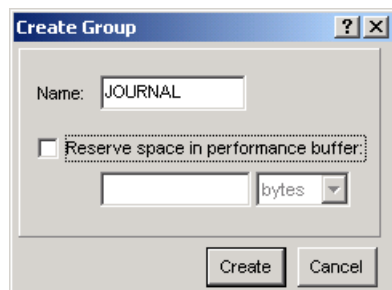
On the **File** menu, click **Exit**.

## Creating a group

A *group*, or *file access group* is a set of named StorHouse files. Each collection set belongs to a group. To create a group, you simply assign a group name and optionally specify the initial number of bytes of performance buffer space to be reserved for the group. The group name is part of the StorHouse file name. Your FileTek customer support representative can help you determine whether to reserve space in the performance buffer for the group.

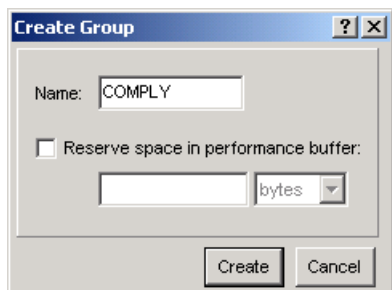


Figure 2-1 illustrates the JOURNAL group for the e-mail collection set.



**Figure 2-1: Group for the e-mail collection set**

Figure 2-2 illustrates the COMPLY group for the compliance collection set.



**Figure 2-2: Group for the compliance collection set**

▼ **To create a group with StorHouse/Admin**

1. If necessary, log in to StorHouse/Admin.
2. In the folder list, click the **Groups** folder.
3. On the **Edit** menu, click **Add/Create**.



4. In the **Create Group** dialog box, type the group name in the **Name** box.
5. Click **Create**.

## Creating a VSET

A *VSET* is one or more physical volumes that are treated as a logical unit of storage. You must create at least one VSET for each collection set. This VSET is a *primary VSET* because it contains primary file copies that are used for normal access. Information about primary files is stored in the StorHouse primary directory.

If you plan to back up data on StorHouse, you must also create a *backup VSET* or an *archive VSET*. See “Backing up StorHouse collections” on page 5-2 for more information about creating a backup VSET. The procedure in this section assumes you will create the backup or archive VSETs later.

At a minimum, you must specify the Directory (primary), the Library, and the Media. Your FileTek customer support representative can help you determine how to set any other VSET attributes.



Figure 2-3 illustrates an example VSET for the e-mail collection set. The VSET will use non-erasable tape media (TGD) and reside in library device L01.

**Create VSET**

VSET name:

---

Create the new VSET with these properties:

Cycle: <input type="text" value="0"/> days	Size: <input type="text" value="0"/> Bytes	Archive VSET: <input type="text"/>
Deactivate: <input type="text" value="0"/> days	Limit: <input type="text" value="0"/> Bytes	Archive FSET: <input type="text"/>
Expire: <input type="text" value="0"/> days	Directory: <input type="text" value="Primary"/>	Backup VSET: <input type="text"/>
Library: <input type="text" value="L01"/>	Hold: <input type="text"/>	Backup FSET: <input type="text"/>
Media: <input type="text" value="TGD"/>		

**Figure 2-3: VSET for the e-mail collection set**



## Chapter 2 - Resource setup

Figure 2-4 illustrates an example VSET for the compliance collection set. The VSET will use non-erasable tape media (TGD) and reside in library device L01.

**Create VSET**

VSET name:

---

Create the new VSET with these properties:

Cycle:	<input type="text" value="0"/> days	Size:	<input type="text" value="0"/> Bytes	Archive VSET:	<input type="text"/>
Deactivate:	<input type="text" value="0"/> days	Limit:	<input type="text" value="0"/> Bytes	Archive FSET:	<input type="text"/>
Expire:	<input type="text" value="0"/> days	Directory:	<input type="text" value="Primary"/>	Backup VSET:	<input type="text"/>
Library:	<input type="text" value="L01"/>	Hold:	<input type="text"/>	Backup FSET:	<input type="text"/>
Media:	<input type="text" value="TGD"/>				

**Figure 2-4: VSET for the compliance collection set**

### ▼ To create a VSET with StorHouse/Admin

1. If necessary, log in to StorHouse/Admin.
2. In the folder list, click the VSETs folder.
3. On the **Edit** menu, click **Add/Create**.
4. In the **Create VSET** dialog box, type a VSET name in the **VSET name** box.
5. In the **Library** list, click the library you want to use. And if the library has multiple media types, click the media you want to use in the **Media** list.





6. In the **Directory** list, accept the default Primary.
7. Enter any other attributes as required.
8. Click **Create** and then click **OK**.
9. Create another VSET or click **Done** if you are finished.

## Creating an FSET

An *FSET* is an area of storage within a VSET. You can create one or more FSETs for a VSET. If you create a set of FSETs, each FSET name must end in a number starting with 0, for example, COMPLY0.

This procedure explains how to create an FSET for a VSET that will reside in a library device. Refer to the *StorHouse/Admin System Administrator's Quick Reference* for the procedure to create an FSET for magnetic disk. At a minimum, you must assign an FSET name. Your FileTek customer support representative can help you determine how to set any other FSET attributes, such as retention.



## Chapter 2 - Resource setup

Figure 2-5 illustrates an example FSET for the e-mail collection set.

**Create FSET** ? X

FSET name: JJAN02

In VSET: JJAN02

Size: 0 Bytes

Limit: 0 Bytes  
0 = No limit

Update: %

Allocation: ☒ Make contiguous

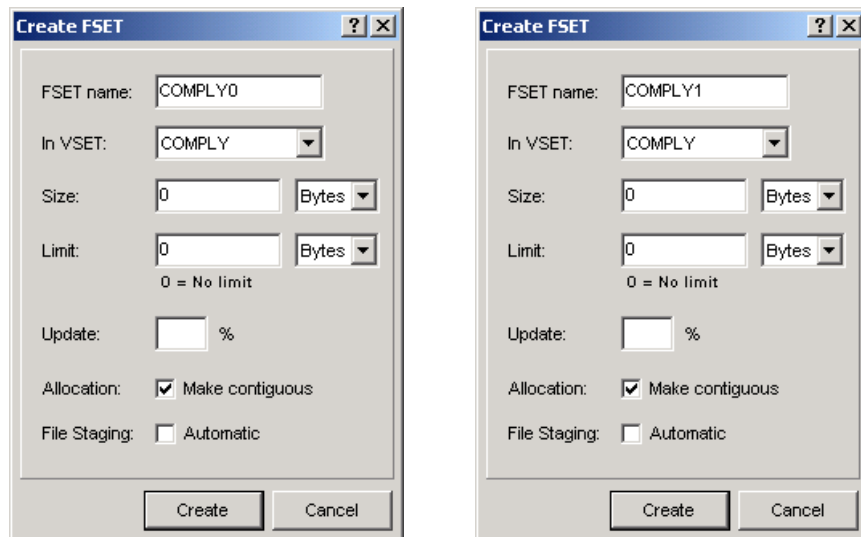
File Staging: ☐ Automatic

Create Cancel

**Figure 2-5: FSET for the e-mail collection set**



Figure 2-6 illustrates an example set of FSETs for the compliance collection set.



**Figure 2-6: Set of FSETs for the compliance collection set**

▼ **To create an FSET with StorHouse/Admin**

1. If necessary, log in to StorHouse/Admin.
2. In the folder list, open the folder for the primary VSET in the VSETs list.
3. Click the FSETs folder.
4. On the **Edit** menu, click **Add/Create**.
5. In the **Create FSET** dialog box, type the name of the primary FSET in the **FSET name** box.



6. In the **In VSET** list, verify the name of the primary VSET that will contain the new FSET.
7. Enter any other attributes as required.
8. Click **Create**, click **OK**, and then click **Done**.

# Creating a StorHouse account

StorHouse/RFS uses a *StorHouse account* to write StorHouse collections and to own, load, and query StorHouse tables. For a collection set, you can create one account to perform both functions, or you can create two accounts to perform the separate functions. Later, you specify the account (UserId parameter) to write StorHouse collections in a system definition and the account to load and query StorHouse tables in a storage definition. For simplicity, consider creating one account for both functions.

An account has a set of StorHouse privileges and attributes. Table 2-6 summarizes the privileges and attributes required to write StorHouse collections and to own, load, and query StorHouse tables.

**Table 2-6: StorHouse account privileges and attributes**

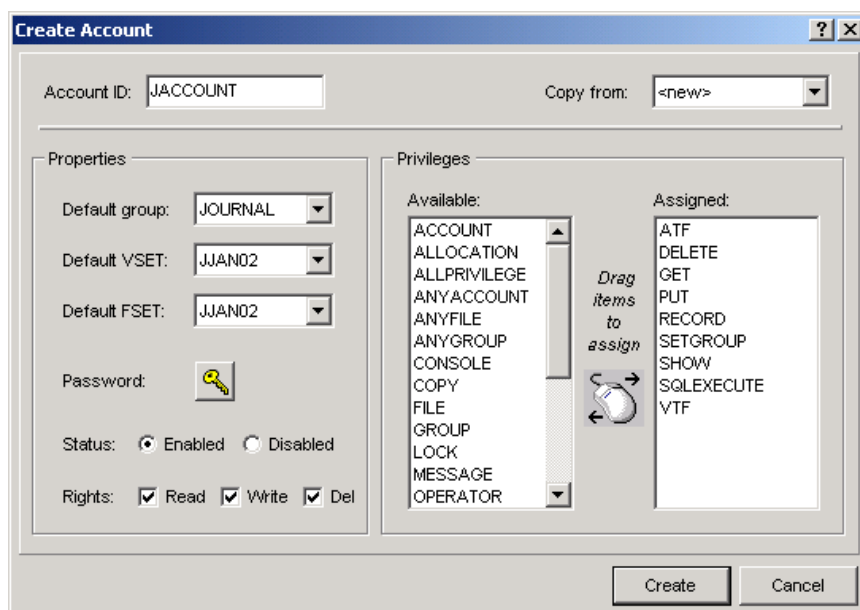
Default group:	Name of the group created for the collection set
Default VSET:	Name of the primary VSET created for the collection set
Default FSET:	Name of the primary FSET created for the collection set
Rights:	Read, Write, and Delete
Privileges:	ATF, DELETE, GET, PUT, RECORD, RESOURCE, SETGROUP, SHOW, SQLEXECUTE, and VTF



**Note:** The SQLEXECUTE privilege is required to load and query StorHouse tables. The RESOURCE privilege is required for StorHouse/RFS to create table arrays. The remaining privileges listed in Table 2-6 are required to write StorHouse collections.

The account must have a password, which you assign when you create the account. You also specify this password later in the system definition and the collection definition.

Figure 2-7 illustrates an example account for the e-mail collection set. This account has all of the attributes required to write StorHouse collections to the current default group (JOURNAL), VSET (JJAN02), and FSET (JJAN02) and to own, load, and query the StorHouse tables that you will create later.

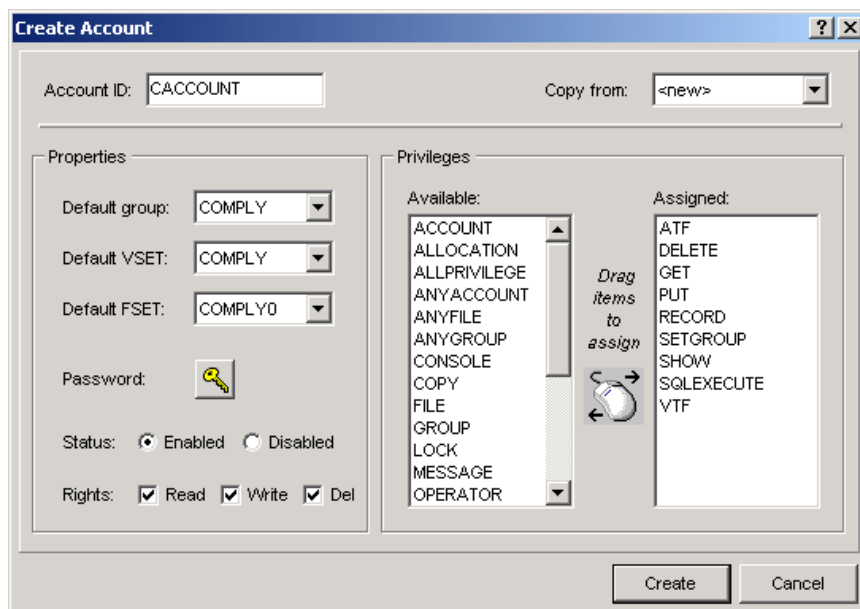


**Figure 2-7: Account for the e-mail collection set**



## Chapter 2 - Resource setup

Figure 2-8 illustrates an example account for the compliance collection set. This account has all of the attributes required to write StorHouse collections to the current default group (COMPLY), VSET (COMPLY), and FSET (COMPLY0) and to own, load, and query the StorHouse tables that you will create later.




**Figure 2-8: Account for the compliance collection set**

### ▼ To create a StorHouse account with StorHouse/Admin

1. If necessary, log in to StorHouse/Admin.
2. In the folder list, click the **Accounts** folder.
3. On the **Edit** menu, click **Add/Create**.
4. In the **Create Account** dialog box, type the account name in the **Account ID** box.



5. In the **Default group** box, click the group to use as the account's default group.
6. In the **Default VSET** box, click the VSET to use as the account's default VSET.
7. In the **Default FSET** box, click the FSET to use as the account's default FSET.
8. For the **Password** check box, click the  button.
  - a. In the **Enter Passwords** dialog box, type the password in the **New** box. The password does not display when you type it. You will provide this password later when adding a system definition and a collection definition.
  - b. Retype the password in the **Retype** box and press the Enter key.
9. In the Status area, be sure **Enabled** is selected.
10. In the Rights area, select the **Read**, **Write**, and **Del** checkboxes.
11. In the Privileges area, assign the appropriate privileges (drag them from the **Available** list to the **Assigned** list).

These privileges are required to write StorHouse collections: ATF, DELETE, GET, PUT, RECORD, SETGROUP, SHOW, and VTF.

The SQLEXECUTE is required to load and query StorHouse tables. The RESOURCE privilege is required to extend the table array. See "To assign the RESOURCE privilege to an account" on page 2-18 for instructions on assigning the RESOURCE privilege.

12. Click **Create** and then click **OK**.



13. Create another account or click **Done** when you are finished.

▼ **To assign the **RESOURCE** privilege to an account**

1. In the folder list, expand the **Databases** folder.
2. Expand the database.
3. Click the **DB Accounts** folder.
4. In the **Database Accounts** list, right-click the account name.
5. Click **Resource**. (A check mark indicates the privilege is granted.)

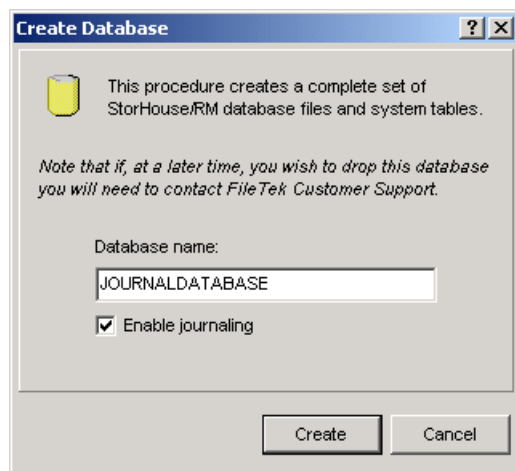
## Creating a StorHouse database

A *StorHouse database* contains StorHouse tables. Perform this procedure to create a StorHouse database. All you need to provide is the database name. You may de-select the journaling feature; however, FileTek recommends that you keep the journaling feature enabled in the event the database must be restored. Remember that the database name is case sensitive and should be descriptive.



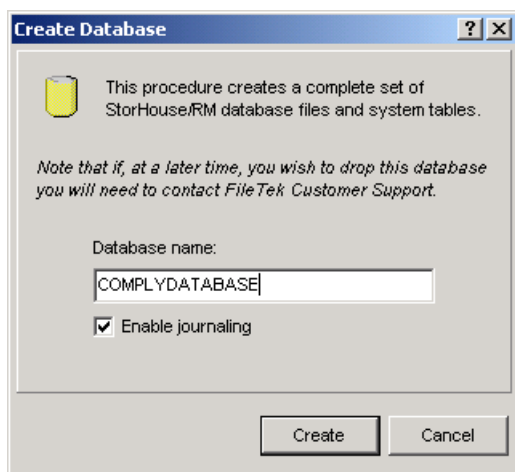


Figure 2-9 illustrates an example database for the e-mail collection set.



**Figure 2-9: Database for the e-mail collection set**

Figure 2-10 illustrates an example database for the compliance collection set.



**Figure 2-10: Database for the compliance collection set**



### ▼ To create a database with StorHouse/Admin

1. If necessary, log in to StorHouse/Admin.
2. In the folder list, click the **Databases** folder.
3. On the **Edit** menu, click **Add/Create**.
4. In the **Create Database** dialog box, type a database name in the **Database name** box.
5. Click **Create**.

## Creating StorHouse tables

Each collection set requires a set of *StorHouse tables* to hold file locator data, collection metadata, and security entries (if applicable). You use the StorHouse/RFS *tblgen utility* to create the collections table and the security table. StorHouse/RFS automatically creates file locator tables after you create a collections table. The StorHouse database and the StorHouse account and password must exist before you run this utility.

**Note:** You also use the *tblgen utility* to create statistics tables and an aliases table. See “Creating statistics tables” on page 6-11 and “Creating an aliases table” on page 4-4 for more information. You can create all of these tables at the same time.

The *tblgen utility* prompts for input, including the type of table to create, host name, database name, table name, table owner (StorHouse account), and password. The security table requires additional input: user directory (UserDir parameter in the StorHouse/RFS configuration file) and number of subdirectories (KeepSubdirectories parameter).



After you provide the input, the tblgen utility connects to the specified database on the specified StorHouse system using the provided account and password and then creates the tables. With the exception of the security table, if the tables already exist, the utility prompts you to re-run the program with a different table name.

When you create a security table, the user directory you specify must match the UserDir value in the collector definition in the StorHouse/RFS configuration file. Be sure to specify the same user directory when creating the collector definition described on page 2-42.

### ▼ To create StorHouse tables in Windows

1. Log in to the StorHouse/RFS server platform with the administrator account.
2. From the **Start** menu, select **Run**.
3. Enter **CMD** and click **OK**.
4. In the **DOS** box enter the Drive letter where StorHouse/RFS is installed, for example:

C:

5. Change to the StorHouse/RFS executable directory, for example:

```
CD "\Program Files\RFS\v4r0"
```

The quotes are necessary if StorHouse/RFS is installed in the Program Files directory.

6. Start the tblgen utility by typing  
  
tblgen
7. Respond to the prompts.



## Chapter 2 - Resource setup

Table 2-7 lists the prompts that the tblgen utility issues and the responses that you provide.

**Table 2-7: Prompts and responses for the tblgen utility**

Prompt	Response
Enter the type of RFS tables to create	Any combination of the following letters: <ul style="list-style-type: none"><li>■ S - Statistics tables</li><li>■ L - Collections table</li><li>■ A - Aliases table</li><li>■ B - Security table</li></ul> When specifying multiple letters, do not use spaces or commas. Example: SLAB
Enter the host name	The DNS name of the host (StorHouse system) where the tables will reside. Example: alpha2
Enter the database name	The name of the StorHouse database that will contain the tables used by StorHouse/RFS. The database name is case sensitive. Example: JOURNALDATABASE



**Table 2-7: Prompts and responses for the tblgen utility (cont.)**

Prompt	Response
Enter the base table name	<p>The part of the table name common to the type(s) of table(s) being created. The simplest StorHouse/RFS configuration would use the same base table name for all tables. However, you may designate any base table for any type of table by running the tblgen utility multiple times with different table types and different base table names.</p> <p>Example: If you are creating all table types (slab) and specify a base table name of RFSTABLE, then:</p> <ul style="list-style-type: none"> <li>■ The collections table would be named RFSTABLE_C and corresponding file locator tables would be named RFSTABLE_0, RFSTABLE_1, RFSTABLE_2, and so on</li> <li>■ The security table would be named RFSTABLE_S</li> <li>■ The aliases table would be named RFSTABLE_ALIASES</li> <li>■ The statistics tables would be named RFSTABLE_RFS, RFSTABLE_SYSTEMS, and RFSTABLE_TABLES</li> </ul>
Enter the owner for the tables	<p>The StorHouse account that is authorized to create the tables for StorHouse/RFS. You also specify this account name at the UserId parameter in the storage definition. At a minimum, this account must have SQLEXECUTE and RESOURCE privileges.</p> <p>Example: JACCOUNT</p>
Enter the owner password	<p>The corresponding password for the StorHouse account. The password is not displayed.</p>



## Chapter 2 - Resource setup

If you are creating a security table (table type B), the `tblgen` utility issues the additional prompts listed in Table 2-8.

**Table 2-8: Additional prompts and responses for the security table**

Prompt	Response
Enter a collector's UserDir or type 'Q' to quit	The UserDir path in a collector definition in the StorHouse/RFS configuration file.  Examples: \\journal /comply
Enter KeepSubdirectories value for this directory	The number of subdirectory levels below the staging directory on which StorHouse/RFS will retain security information for this collector's UserDir.  Example: 4

### ▼ To create StorHouse tables in UNIX

1. If necessary, log in to the StorHouse/RFS server as root. You should be running bourne shell.
2. Execute the following command to get the necessary definitions.

```
. /rfs/v4r0/bin/defs
```

3. Change to the StorHouse/RFS configuration directory:

```
cd /rfs/v4r0/conf
```

4. Start the `tblgen` utility:

```
./tblgen.exe
```

5. Respond to the prompts. See Table 2-7 on page 2-22 and Table 2-8 on page 2-24 for more information.



## Starting and exiting the StorHouse/RFS configuration file utility

In a Windows environment, you use the StorHouse/RFS configuration file utility to add, update, and delete definitions and adjust general StorHouse/RFS parameters. This utility is installed on the StorHouse/RFS server platform.

### ▼ To start the StorHouse/RFS configuration file utility

1. If necessary, log in to the StorHouse/RFS server platform with the administrator account.
2. On the Windows task bar, click **Start**.
3. Point to **Programs**, point to **RFS**, and then click **RFS Config Update**.

### ▼ To exit the StorHouse/RFS configuration file utility

In the **RFS Configuration** dialog box, click **Exit**.

## Adding a system definition

A *system definition* identifies a StorHouse system. You specify a system definition name in the **STATS** section, **ALIASES** section, and storage definition. For StorHouse collections, a system definition specifies the logical organization (group, VSET, and FSET) of data on StorHouse. You must create one system definition for each separate group/VSET/FSET to contain StorHouse collections. For StorHouse/RFS duplexing, you must create one system definition for the primary StorHouse system and a second system definition for the secondary StorHouse system.



## Chapter 2 - Resource setup

See “System definition” on page A-28 for descriptions of the system definition parameters. Some considerations are as follows:

- The STHName, DNSName, UserId, and Password are required. For UNIX, the STHName value must also be in the SMCONFIG file. For Windows, the StorHouse/RFS configuration file utility automatically updates the FILETEK.INI file (which contains the DNS and port needed by StorHouse) when you add or delete a system definition or change the DNSName parameter in an existing one.
- The UserId must be the StorHouse account with the assigned default group, VSET, and FSET for the collection set. This account must have the following StorHouse privileges to write StorHouse collections: ATF, DELETE, GET, PUT, RECORD, SETGROUP, SHOW, and VTF.
- For UNIX, whenever you enter or change a Password parameter in a definition, you must run the gen\_cfg program. See “Encrypting passwords” on page 4-11 for instructions on running the gen\_cfg program. You can run this program once after adding all definitions.
- You can omit the group, VSET, and FSET values in the system definition to use the default values for the StorHouse account.
- If you are using a set of FSETs, the FSET name is the base name, that is COMPLY instead of COMPLY0 or COMPLY1. See “Naming StorHouse/RFS resources” on page 1-23 for more information about naming FSETs in a set.
- Three ODBC parameters—DBDriver, DBPort, and DBHost—are set by default to use the StorHouse/ODBC driver. In a Windows environment, you cannot use the StorHouse/RFS configuration file utility to update these parameters. You must use a text editor if you need to change one of these parameters.
- You must restart the StorHouse/RFS service after adding definitions or updating static parameters. Or you must request a re-read of the





StorHouse/RFS configuration file if you change only a dynamic parameter. See “Summary of parameters” on page A-6 to determine the static parameters. See “Restarting the StorHouse/RFS service” on page 7-8 for instructions on stopping and starting StorHouse/RFS. See “Rereading the StorHouse/RFS configuration file” on page 7-7 for instructions after updating a dynamic parameter.

Figure 2-11 illustrates an example system definition for the e-mail collection set.

The screenshot shows the 'RFS Configuration' window with the 'System Definition' tab selected. The left sidebar contains a menu with options: Systems, Storage, Collections, Collectors, General, Statistics, Exclusions, Aliases, and Exit. The main area displays the configuration for the 'JournalSet' system. The fields are as follows:

Field	Value
Defined systems:	JournalSet
StorHouse system definition	
DNSName:	alpha2
STHName:	alpha2
UserId:	JACCOUNT
Password:	XXXXXXXXXX
Group:	
VSET:	
FSET:	
FSETSegments:	1
RetryInterval:	0
TimeoutOverride:	0
VTF:	Next
Checkpoint:	500
MailRecipient:	MJONES@cc.com

At the bottom of the window, the copyright notice 'Copyright (C) 2002 - 2005, FileTek, Inc.' and the version 'Version: 4.0.0.5' are displayed.

Figure 2-11: System definition for the e-mail collection set



## Chapter 2 - Resource setup

The values in the sample system definition for the e-mail collection set have the following meaning.

**Table 2-9: System definition for the e-mail collection set**

Parameter	Meaning
DNSName	Store and access data for the e-mail collection set on the StorHouse system with the DNS name of alpha2.
STHName	Store and access data for the e-mail collection set on the StorHouse system with the StorHouse name of alpha2.
UserId	Log in to the StorHouse system using the StorHouse account named JACCOUNT to write StorHouse collections for the e-mail collection set.
Password	Use the encrypted password to log in to the StorHouse system.
Group	Use the default group for the JACCOUNT to write StorHouse collections for the e-mail collection set.
VSET	Use the default VSET for the JACCOUNT to write StorHouse collections for the e-mail collection set.
FSET	Use the default FSET for the JACCOUNT to write StorHouse collections for the e-mail collection set.
FSETSegments	Write StorHouse collections to one FSET for the e-mail collection set (do not use a set of FSETs).
RetryInterval	Try to log in to a StorHouse/SM or StorHouse/RM system whether it is up or down.
TimeoutOverride	Use the default timeout period for StorHouse requests.
VTF	Write StorHouse collections to the resident FSET for the e-mail collection set at the next StorHouse write-back operation.
CheckPoint	Create a new data extent when a StorHouse collection is 500 MB.
MailRecipient	Send e-mail messages about the collection status to MJONES@cc.com.



The following example illustrates a system definition for the compliance collection set.

```
[ComplianceSet]
DNSName=serv01
STHName=sm
DBDriver=StorHouse
DBPort=
DBHost=
UserId=CACCOUNT
Password=^u=
PasswordType=
Group=COMPLY
VSET=COMPLY
FSET=COMPLY
FSETSegments=2
RetryInterval=0
TimeoutOverride=0
VTF=NEXT
Checkpoint=
MailRecipient=hope.forman@gwi.com
```

The values in the sample system definition for the compliance collection set have the following meaning.

**Table 2-10: System definition for the compliance collection set**

Parameter	Meaning
DNSName	Store and access data for the compliance collection set on the StorHouse system with the DNS name of serv01.
STHName	Store and access data for the compliance collection set on the StorHouse system with the StorHouse name of sm. An entry for sm is in the SMCONFIG file.
DBDriver	Use the StorHouse/ODBC driver to connect to the StorHouse database.
DBPort	Use the default port 1990 for the ODBC connection.
DBHost	Use the host serv01 (value of DNSName) for the ODBC connection.




## Chapter 2 - Resource setup

**Table 2-10: System definition for the compliance collection set**

Parameter	Meaning
UserId	Log in to the StorHouse system using the StorHouse account named CACCOUNT to write StorHouse collections.
Password	Use the specified password to log in to the StorHouse system.
Group	Write StorHouse collections to the group named COMPLY.
VSET	Write StorHouse collections to the VSET named COMPLY.
FSET	Write StorHouse collections to the FSET named COMPLY (base name).
FSETSegments	Write StorHouse collections to two FSETs (COMPLY0 and COMPLY1).
RetryInterval	Try to log in to a StorHouse/SM or StorHouse/RM system whether it is up or down.
TimeoutOverride	Use the default timeout period for StorHouse requests.
VTF	Write the collection to the performance buffer and then copy the collection to the primary FSET at the next StorHouse write-back operation.
CheckPoint	Use the default checkpoint value of 100 MB.
MailRecipient	Send e-mail messages to hope.forman@gwi.com.

▼ **To add a system definition with the StorHouse/RFS configuration file utility**

1. If necessary, start the StorHouse/RFS configuration file utility.
2. In the **RFS Configuration** dialog box, click **Systems**, and then click .
3. In the **Defined systems** list, enter a name for the system definition.
4. In the **StorHouse system definition** area, enter values for the parameters.



5. Click Save.

### ▼ To add a system definition with a text editor in UNIX

1. If necessary, log in to the StorHouse/RFS server as root.
2. Open the `/rfs/files/rfs.cfg` file with the desired text editor. The following examples uses emacs.

```
emacs /rfs/files/rfs.cfg
```

3. Insert the definition name (for example, `[ComplianceSet]`) and the system definition parameters and values. For example:

```
[ComplianceSet]
DNSName=serv01
STHName=sm
DBDriver=StorHouse
DBPort=
DBHost=
UserId=CACCOUNT
Password=^u=
PasswordType=
Group=COMPLY
VSET=COMPLY
FSET=COMPLY
FSETSegments=2
RetryInterval=0
VTF=NEXT
CheckPoint=100
MailRecipient=hope.forman@gwi.com
```

4. Save the file.
5. Run the `gen.cfg` program to encrypt the password.

See “Encrypting passwords” on page 4-11 for instructions on running the `gen_cfg` program. You can run this program now or after adding all definitions.



# Adding a storage definition

A *storage definition* specifies where to load and query file locator data (in which table array) and where to write the collection metadata in the collections (\_C) table. You associate a storage definition with a collection definition. Multiple collection definitions can use the same storage definition. See “Storage definition” on page A-38 for descriptions of the storage definition parameters. Some considerations are as follows:

- You must supply values for the following required parameters: SystemName, Database, TableName, UserId, and Password.
- The MirrorName is required if you're using StorHouse/RFS duplexing. This value identifies the system definition for the secondary StorHouse system.
- The TableName is the StorHouse table name in the format owner.base\_table\_name. This name must match the owner and table name provided when running the tblgen utility to create the L table type (collections table and corresponding file locator tables).
- The UserId must be the owner of the StorHouse table specified at the TableName parameter.
- The Database must be the StorHouse database containing the StorHouse table. This name is case sensitive and must match the database name provided when running the tblgen utility to create the table specified at the TableName parameter.
- For UNIX, if you enter or change the Password parameter in a definition, you must run the gen\_cfg program after updating the StorHouse/RFS configuration file. See “Encrypting passwords” on page 4-11 for instructions on running the gen\_cfg program.



- You must restart the StorHouse/RFS service after adding definitions or updating static parameters. Or you must request a reread of the StorHouse/RFS configuration file if you change only a dynamic parameter. See “Summary of parameters” on page A-6 to determine the static parameters. See “Restarting the StorHouse/RFS service” on page 7-8 for instructions on stopping and starting StorHouse/RFS. See “Rereading the StorHouse/RFS configuration file” on page 7-7 for instructions after updating a dynamic parameter.



## Chapter 2 - Resource setup

Figure 2-13 illustrates an example storage definition for the e-mail collection set.

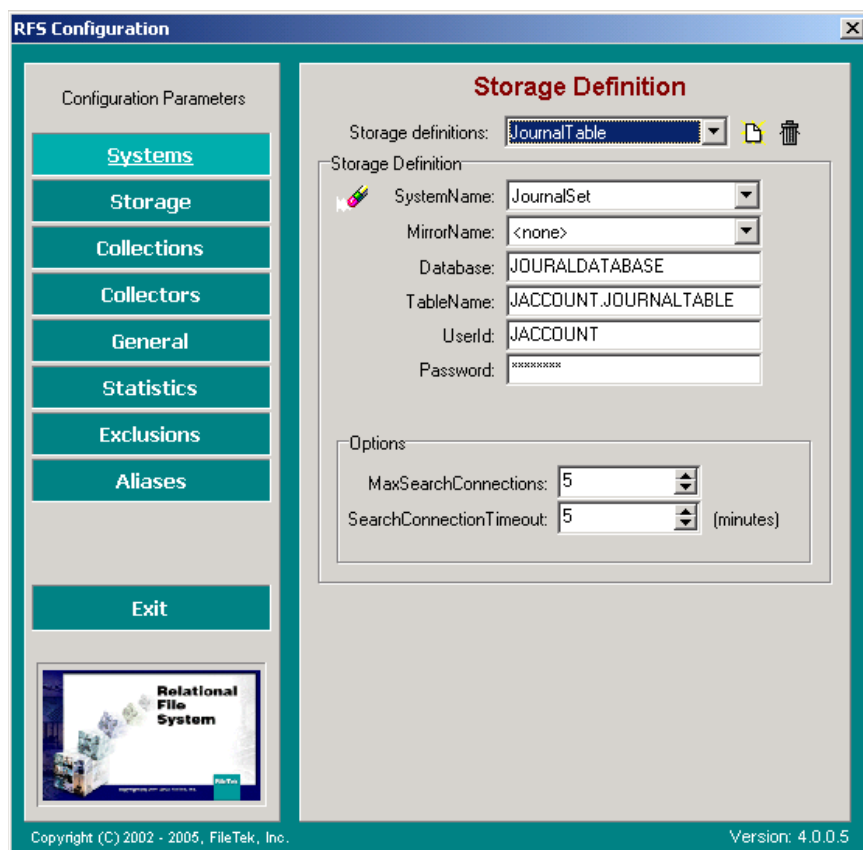


Figure 2-12: Storage definition for the e-mail collection set





The values in the sample storage definition for the e-mail collection set has the following meaning.

**Table 2-11: Storage definition for the e-mail collection set**

Parameter	Meaning
SystemName	Use the JournalSet system definition to identify the primary StorHouse system to store file locator data and collection metadata for the e-mail collection set.
MirrorName	Do not use StorHouse/RFS duplexing, that is, do not store and access data on a secondary StorHouse system for the e-mail collection set.
Database	Store the file locator data and collection metadata for the e-mail collection set in the StorHouse database called JOURNALDATABASE.
TableName	Store the file locator data and collection metadata for the e-mail collection set in StorHouse tables with a base file name of JACCOUNT.JOURNALTABLE.
UserId	Use the StorHouse account JACCOUNT to load file locator data and collection metadata into the StorHouse tables and to query the tables.
Password	Use the encrypted password to log in to the StorHouse system to access the StorHouse tables.
MaxSearchConnections	Allow up to five concurrent ODBC connections to the JACCOUNT.JOURNALTABLE.
SearchConnectionTimeout	Release an idle ODBC connection after five minutes.

The following example illustrates a storage definition for the compliance collection set.

```
[ComplianceTable]
SystemName=ComplianceSet
MirrorName=
Database=COMPLYDATABASE
TableName=CACCOUNT.COMPLYTABLE
UserId=CACCOUNT
Password=~6]
```



## Chapter 2 - Resource setup

```
PasswordType=  
MaxSearchConnections=12  
SearchConnectionTimeout=0
```


The values in the sample storage definition for the compliance collection set have the following meaning.

**Table 2-12: Storage definition for the compliance collection set**

Parameter	Meaning
SystemName	Use the ComplianceSet system definition to identify the StorHouse system to store file locator data and collection metadata for the compliance collection set.
MirrorName	Do not use StorHouse/RFS duplexing.
Database	Store the file locator data and collection metadata for the compliance collection set in the StorHouse database called COMPLYDATABASE.
TableName	Store the file locator data and collection metadata for the compliance collection set in StorHouse tables with a base name of CACCOUNT.COMPLYTABLE.
UserId	Use the StorHouse account CACCOUNT to load file locator data and collection metadata into the StorHouse tables and to query the tables.
Password	Use the encrypted password to log in to the StorHouse system to access the StorHouse tables.
MaxSearchConnections	Allow up to 12 concurrent ODBC connections to the CACCOUNT.COMPLYTABLE.
SearchConnectionTimeout	Close the ODBC connection immediately after the search is done.



### ▼ To add a storage definition with the StorHouse/RFS configuration file utility

1. If necessary, start the StorHouse/RFS configuration file utility.
2. In the **RFS Configuration** dialog box, click **Storage**, and then click .
3. In the **Storage definitions** list, enter a name for the storage definition.
4. In the **Storage Definition** and **Options** areas, enter values for the parameters.
5. Click **Save**.

### ▼ To add a storage definition with a text editor in UNIX

1. If necessary, log in to the StorHouse/RFS server as root.
2. Open the `/rfs/files/rfs.cfg` file with the desired text editor. The following examples uses emacs.

```
emacs /rfs/files/rfs.cfg
```

3. Insert the definition name (for example, `[ComplianceTable]`) and the storage definition parameters and values. For example:

```
[ComplianceTable]
SystemName=ComplianceSet
MirrorName=
Database=COMPLYDATABASE
TableName=CACCOUNT.COMPLYTABLE
UserId=CACCOUNT
...
```

4. Save the file.



5. Run the `gen.cfg` program to encrypt the password.

See “Encrypting passwords” on page 4-11 for instructions on running the `gen_cfg` program. You can run this program now or after adding all definitions.

## Adding a collection definition

A *collection definition* defines collection set options. You must create one collection definition for each collection set. Later, you assign a collection definition to a collector definition. See “Collection definition” on page A-42 for descriptions of the collection definition parameters. Some considerations are as follows:

- The `Storage` parameter and the `CollectionDir` parameter are required.
- You must restart the StorHouse/RFS service after adding definitions or updating static parameters. Or you must request a reread of the StorHouse/RFS configuration file if you change only a dynamic parameter. See “Summary of parameters” on page A-6 to determine the static parameters. See “Restarting the StorHouse/RFS service” on page 7-8 for instructions on stopping and starting StorHouse/RFS. See “Rereading the StorHouse/RFS configuration file” on page 7-7 for instructions after updating a dynamic parameter.

Figure 2-13 illustrates an example collection definition for the e-mail collection set.

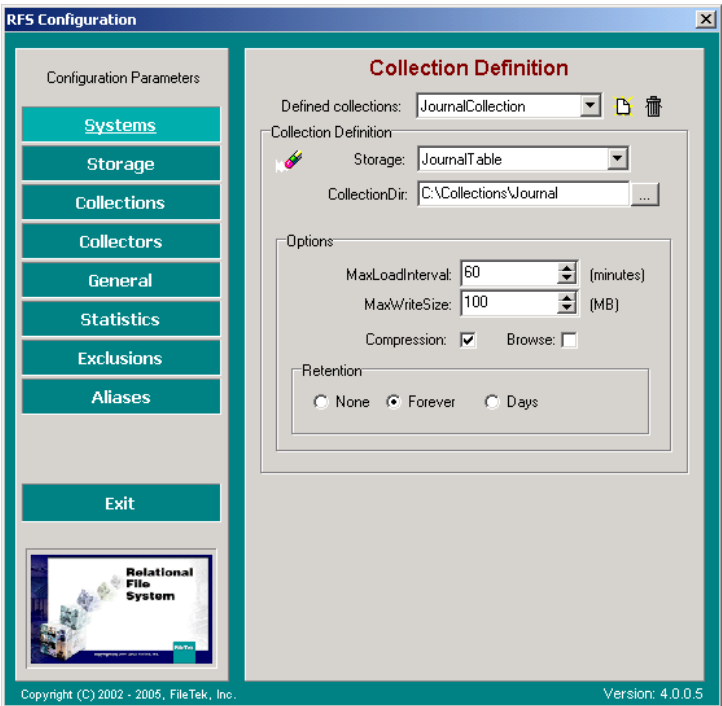


Figure 2-13: Collection definition for the e-mail collection set

The values in the sample collection definition for the e-mail collection set have the following meaning.

Table 2-13: Collection definition for the e-mail collection set	
Parameter	Meaning
Storage	Use the JournalTable storage definition to store file locator data and collection metadata for this collection set.
CollectionDir	Create the load file containing the file locator data for this collection set in C:\Collections\Journal.



## Chapter 2 - Resource setup

**Table 2-13: Collection definition for the e-mail collection set (cont.)**

Parameter	Meaning
MaxLoadInterval	Wait 60 minutes between writes and loads of e-mail collections and file locator data.
MaxWriteSize	Close a local collection for the e-mail collection set when it reaches 25 MB.
Compression	Compress the data in the e-mail collection set.
Browse	Disable browsing of the e-mail collection set.
Retention	Set the retention period for user files in the e-mail collection set to Forever.

The following example illustrates a collection definition for the compliance collection set.

```
[ComplianceCollection]
Storage=ComplyTable
CollectionDir=/rfs/control
MaxLoadInterval=1440
MaxWriteSize=400
MaxCollectionSpace=1200
Compression=No
Browse=Yes
Retention=2555
```

The values in the sample collection definition for the compliance collection set have the following meaning.


**Table 2-14: Collection definition for the compliance collection set**

Parameter	Meaning
Storage	Use the ComplyTable storage definition for storing file locator data and collection metadata for this collection set.
CollectionDir	Create the load file containing the file locator data for this collection set in /rfs/control.

**Table 2-14: Collection definition for the compliance collection set**

Parameter	Meaning
MaxLoadInterval	Wait 1440 minutes between writes and loads of compliance collections and file locator data.
MaxWriteSize	Close a local collection of the compliance collection set when it reaches 400 MB.
Compression	Do not compress the data in the compliance collection set.
Browse	Enable browsing of the compliance collection set in the virtual file system.
Retention	Set the retention period for user files in the compliance collection set to 2555 days (about 7 years).

▼ **To add a collection definition with the StorHouse/RFS configuration file utility**

1. If necessary, start the StorHouse/RFS configuration file utility.
2. In the **RFS Configuration** dialog box, click **Collections**, and then click .
3. In the **Defined collections** list, enter a name for the collection definition.
4. In the **Collection Definition** and **Options** areas, enter values for the parameters.
5. Click **Save**.
6. If prompted to create any directories, click **Yes**.



### ▼ To add a collection definition with a text editor in UNIX

1. If necessary, log in to the StorHouse/RFS server as root.
2. Open the `/rfs/files/rfs.cfg` file with the desired text editor. The following examples uses emacs.

```
emacs /rfs/files/rfs.cfg
```

3. Insert the definition name (for example, `[ComplianceCollection]`) and the collection definition parameters and values. For example:

```
[ComplianceCollection]
Storage=ComplianceTable
CollectionDir=/rfs/control
MaxLoadInterval=1440
MaxWriteSize=400
MaxCollectionSpace=1200
Compression=No
Browse=Yes
Retention=2555
```

4. Save the file.

## Adding a collector definition

A *collector definition* defines each StorHouse/RFS collector. You assign a collector to a collection definition. See “Collector definition” on page





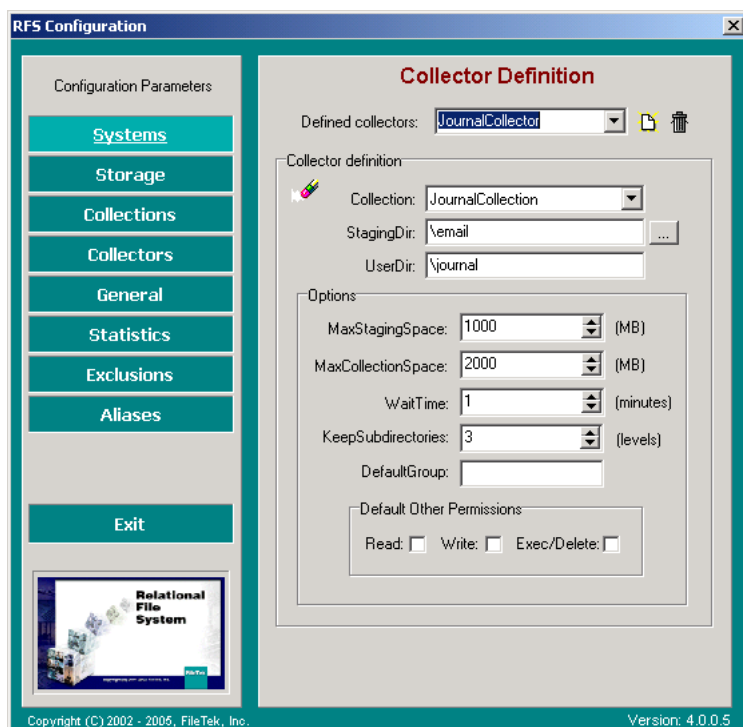
A-47 for descriptions of the collector definition parameters. Some considerations are as follows:

- The Collection definition name and the StagingDir are required.
- The StagingDir and the UserDir are the path where the collector checks for files to archive. StorHouse/RFS creates a folder for the UserDir in the virtual file system.
- You must restart the StorHouse/RFS service after adding definitions or updating static parameters. Or you must request a reread of the StorHouse/RFS configuration file if you change only a dynamic parameter. See “Summary of parameters” on page A-6 to determine the static parameters. See “Restarting the StorHouse/RFS service” on page 7-8 for instructions on stopping and starting StorHouse/RFS. See “Rereading the StorHouse/RFS configuration file” on page 7-7 for instructions after updating a dynamic parameter.



## Chapter 2 - Resource setup

Figure 2-14 illustrates an example collector definition for the e-mail collection set.



**Figure 2-14: Collector definition for the e-mail collection set**



The values in the sample collector definition for the e-mail collection set have the following meaning.

**Table 2-15: Collector definition for the e-mail collection set**

Parameter	Meaning
Collection	Assign the JournalCollector to the JournalCollection collection definition.
StagingDir	Assign the JournalCollector to the \email staging directory.
UserDir	Assign the JournalCollector to the \journal user directory. StorHouse/RFS displays this user directory in the virtual file system.
MaxStagingSpace	Set the maximum size of each staging area to 1000 MB.
MaxCollectionSpace	Set the maximum size of the rename and collection directories to 2000 MB.
WaitTime	Collect any files in the \email\journal staging area that have not been modified for 1 minute.
KeepSubdirectories	Capture and store security descriptors for three levels of subdirectories below the collector's root directory. (You set and update this value with the tblgen utility.)
DefaultGroup	Do not enable group access for files collected by the JournalCollector.
Default Other Permissions	Do not enable "other" permissions for files collected by the JournalCollector.



## Chapter 2 - Resource setup

The following example illustrates a collector definition for the compliance collection set. The `ComplianceCollection` collection definition is assigned to the `ComplianceCollector` collector definition.

```
[COLLECTORS]
ComplianceCollector=ComplianceCollection

[ComplianceCollector]
StagingDir=/stage
UserDir=/
WaitTime=1
MaxStagingSpace=1000
MaxCollectionSpace=1200
KeepSubdirectories=0
Group=
Permissions=
```


The values in the sample collector definition for the compliance collection set have the following meaning.

**Table 2-16: Collector definition for the compliance collection set**

Parameter	Meaning
StagingDir	Assign the ComplianceCollector to the /stage staging directory.
UserDir	Do not assign a UserDir for the staging area. The ComplianceCollector will collect files written to /stage.
WaitTime	Collect any files that have not been modified for 1 minute.
MaxStagingSpace	Set the maximum size of the staging area to 1000 MB.
MaxCollectionSpace	Set the maximum size of the rename and collection directories to 1200 MB.
KeepSubdirectories	This value is not used for UNIX installations.
Group	Use UNIX OS group permissions.
Permissions	Use UNIX OS “other” permissions.



### ▼ To add a collector definition with the StorHouse/RFS configuration file utility

1. If necessary, start the StorHouse/RFS configuration file utility.
2. In the **RFS Configuration** dialog box, click **Collectors**, and then click .
3. In the **Defined collectors** list, enter a name for the collector definition.
4. In the **Collector definition** and **Options** areas, enter values for the parameters.
5. Click **Save**.

### ▼ To add a collector definition with a text editor in UNIX

1. If necessary, log in to the StorHouse/RFS server as root.
2. Open the `/rfs/files/rfs.cfg` file with the desired text editor. The following examples uses emacs.

```
emacs /rfs/files/rfs.cfg
```

3. Go to the `[COLLECTORS]` section and specify a collector name (for example, `ComplianceCollector`). Then assign a collection definition (for example, `ComplianceCollection`) to the collector. For example:

```
[COLLECTORS]
ComplianceCollector=ComplianceCollection
```



4. Insert the collector definition name (for example, [ComplianceCollector]) and the collector definition parameters and values. For example:

```
[COLLECTORS]
ComplianceCollector=ComplianceCollection

[ComplianceCollector]
StagingDir=/stage
UserDir=/
WaitTime=1
MaxStagingSpace=1000
MaxCollectionSpace=1200
KeepSubdirectories=0
Group=
Permissions=
```

5. Save the file.

## Adjusting general parameters

The StorHouse/RFS configuration file contains a section for general parameters. You should periodically verify and adjust these values as



needed. Figure 2-15 illustrates the general, or RFS section, parameters for a Windows StorHouse/RFS installation.

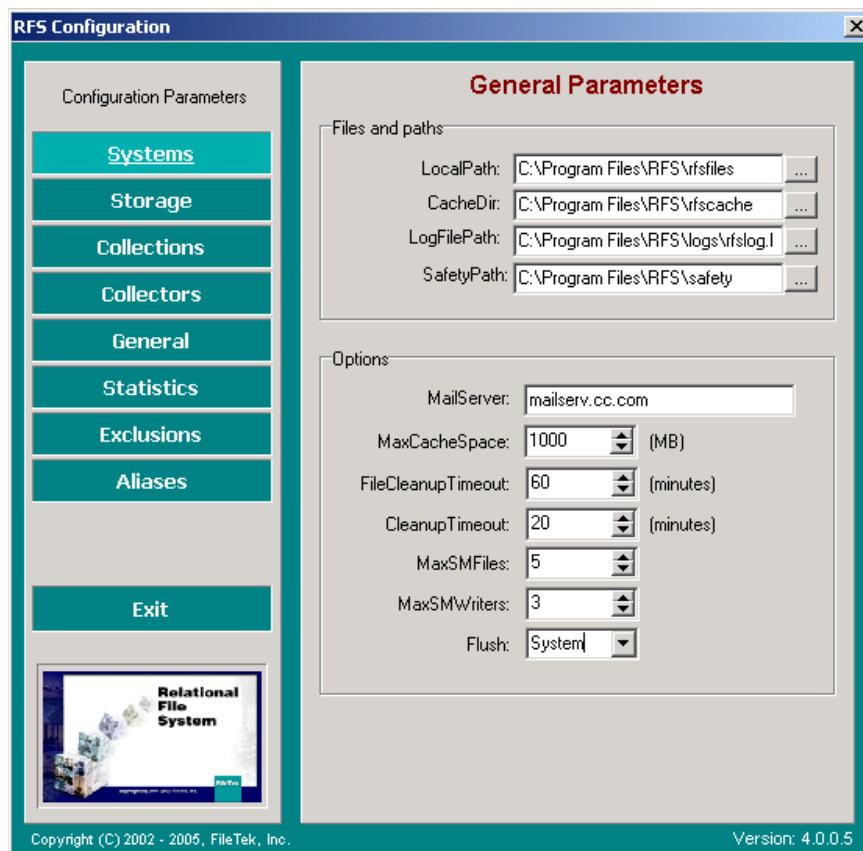


Figure 2-15: General parameters for a Windows installation



## Chapter 2 - Resource setup

The values in the sample general parameters for a Windows installation have the following meaning.

**Table 2-17: General parameters for a Windows installation**

Parameter	Meaning
LocalPath	Store the local statistics file in the path C:\Program Files\RFS\rfsfiles.
CacheDir	Cache data read from StorHouse in the path C:\Program Files\RFS\rfscache. This is the cache directory.
LogFilePath	Create log entries in the file rfslog.log located in the path C:\Program Files\RFS\logs.
SafetyPath	Create a secondary copy of metadata for user file changes in the path C:\Program\RFS\safety.
MailServer	Send e-mail messages through the mailserv.cc.com server.
MaxCacheSpace	Reserve 1000 MB for caching files read from StorHouse.
FileCleanupTimeout	Check the virtual file system every 60 minutes for unused files to remove.
CleanupTimeout	Wait 20 minutes before closing StorHouse connections and purging cache files that are no longer in use.
MaxSMFiles	Allow up to 5 connections to be used for reading files in StorHouse collections.
MaxSMWriters	Allow up to 3 connections to be used for writing StorHouse collections.
Flush	Flush the buffers by the operating system as appropriate for current system operating conditions.





The following example illustrates the general parameters for a UNIX StorHouse/RFS installation.

```
[RFS]
LocalPath=/rfs/files
CacheDir=/rfs/cache
LogFile=/rfs/logs/rfs.log
SafetyPath=
MaxCacheSpace=400
FileCleanupTimeout=60
CleanupTimeout=4
MaxSMFiles=15
MaxSMWriters=7
Flush=No
Simulate=No
```

The values in the sample general parameters for a UNIX installation have the following meaning.

**Table 2-18: General parameters for a UNIX installation**

Parameter	Meaning
LocalPath	Store the local statistics file in the path /rfs/files.
CacheDir	Cache data read from StorHouse in the path /rfs/cache.
LogFilePath	Create log entries in the file rfs.log located in the path /rfs/logs.
SafetyPath	Do not create a secondary copy of metadata for file changes.
MaxCacheSpace	Reserve 400 MB for caching files read from StorHouse.
FileCleanupTimeout	Check the virtual file system every 60 minutes for unused files to remove.
CleanupTimeout	Wait 4 minutes before closing StorHouse connections and purging cache files that are no longer in use.
MaxSMFiles	Allow up to 15 connections to be used for reading files in StorHouse collections.



## Chapter 2 - Resource setup

**Table 2-18: General parameters for a UNIX installation (cont.)**

Parameter	Meaning
MaxSMWriters	Allow up to 7 connections to be used for writing StorHouse collections.
Flush	Flush the buffers by the operating system as appropriate for the current system operating conditions.
Simulate	Do not run StorHouse/RFS in simulation mode.

▼ **To adjust general parameters with the StorHouse/RFS configuration file utility**

1. If necessary, start the StorHouse/RFS configuration file utility.
2. In the **RFS Configuration** dialog box, click **General**.
3. In the **Files and paths** and **Options** areas, enter or change any parameter values.
4. Click **Save**.
5. If you updated a static parameter, restart the StorHouse/RFS service (see page 7-8). Or if you updated a dynamic parameter only, request a reread of the StorHouse/RFS configuration file (see page 7-7).

▼ **To adjust general parameters with a text editor in UNIX**

1. If necessary, log in to the StorHouse/RFS server as root.
2. Open the `/rfs/files/rfs.cfg` file with the desired text editor. The following examples uses emacs.

```
emacs /rfs/files/rfs.cfg
```

3. Go to the `[RFS]` section and make the change.
4. Save the file.



5. If you updated a static parameter, restart the StorHouse/RFS service (see page 7-8). Or if you updated a dynamic parameter only, request a reread of the StorHouse/RFS configuration file (see page 7-7).



## **Chapter 2 - Resource setup**



## Duplexing setup

This chapter describes how to set up your environment for StorHouse/RFS duplexing. Setup includes:

- Adding ODBC data sources
- Configuring the StorHouse/Control Center server
- Configuring StorHouse/Admin
- Creating duplicate StorHouse resources
- Adding system definitions
- Specifying system definition names in other definitions



# Adding ODBC data sources

One ODBC data source is required for each StorHouse system in your environment. If you did not add an ODBC data source for the secondary StorHouse system during StorHouse/RFS installation, add one now. Refer to the *StorHouse/RFS Installation Manual for Windows* or the *StorHouse/ODBC Driver Reference Manual* for instructions on adding an ODBC data source.

# Configuring the StorHouse/Control Center server

One StorHouse/Control Center server can communicate with multiple StorHouse systems, but you must configure a StorHouse/Control Center server to communicate with each StorHouse system. You do this by adding each StorHouse system to CCAdmin. Refer to Chapter 2, “Installing the StorHouse/Control Center environment,” in *Getting Started with StorHouse/Control Center*, publication number 900138, for more information about defining StorHouse systems to CCAdmin.

# Configuring StorHouse/Admin

With StorHouse/Admin, you must:

- Configure two StorHouse data sources
- Create two shortcuts to the data sources
- Set preferences for each data source



For StorHouse/Admin, a *StorHouse data source* is the combination of a StorHouse/Control Center server and a StorHouse system. When you create the StorHouse resources for your collection sets, you log in to StorHouse/Admin through the appropriate data source shortcut. Refer to Chapter 3, “StorHouse/Admin,” in *Getting Started with StorHouse/Control Center*, publication number 900138, for more information about configuring StorHouse/Admin.

## Creating duplicate StorHouse resources

With StorHouse/Admin, you must create the same StorHouse resources to store collections on both the primary and secondary StorHouse systems. Additionally, you must create duplicate resources on both StorHouse systems to store statistics and aliases. Table 3-1 identifies the resources to create on both StorHouse systems.

**Table 3-1: Resources on primary and secondary StorHouse systems**

To store	Create these resources on both systems	See
Collections	Group VSET(s) FSET(s) StorHouse account StorHouse database StorHouse tables	Chapter 2, “Storage setup”
Aliases	StorHouse database (if in a separate database) StorHouse table	Chapter 4, “Security”
Statistics	StorHouse database (if in a separate database) StorHouse tables	Chapter 6, “Statistics”



# Adding system definitions

The StorHouse/RFS configuration file must contain system definitions for the primary and secondary StorHouse systems. The system definitions should be the same except for the StorHouse system name and the DNS name. See “Adding a system definition” on page 2-25 for instructions. For example, in Figure 3-1, the JournalSet system definition



defines the primary StorHouse system and the JournalSet2 system definition identifies the secondary StorHouse system.

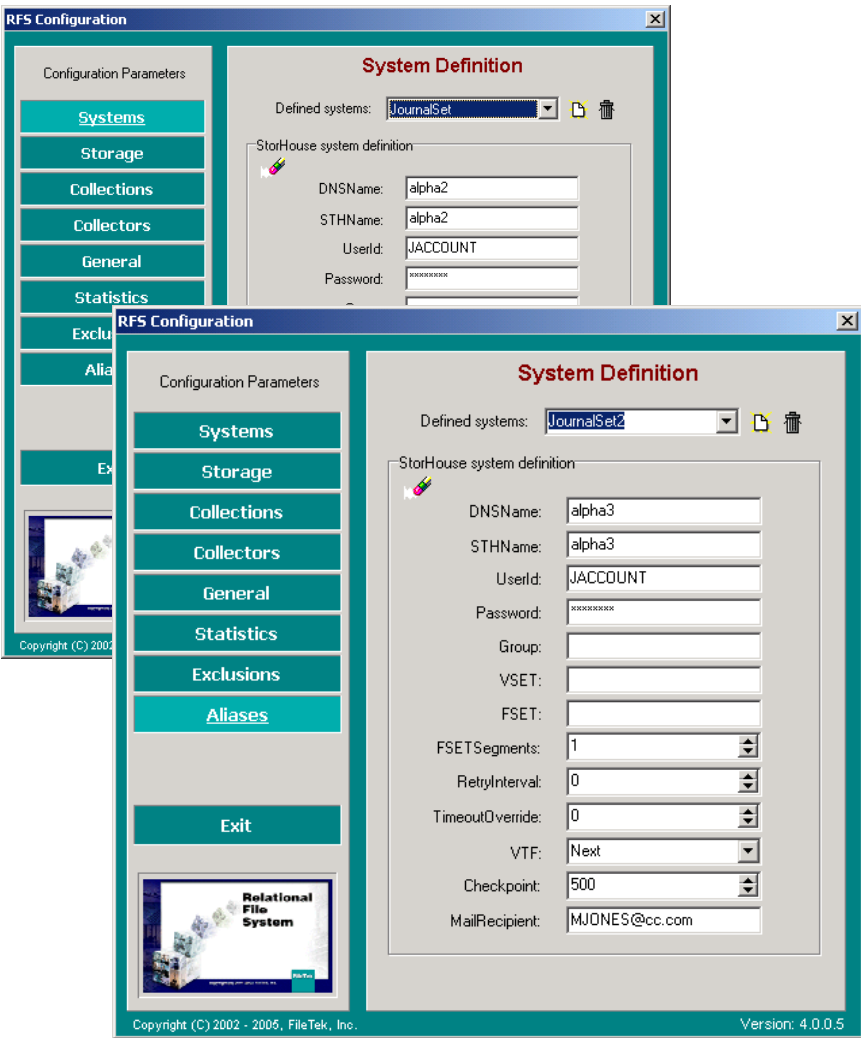


Figure 3-1: Examples of system definitions for duplexing



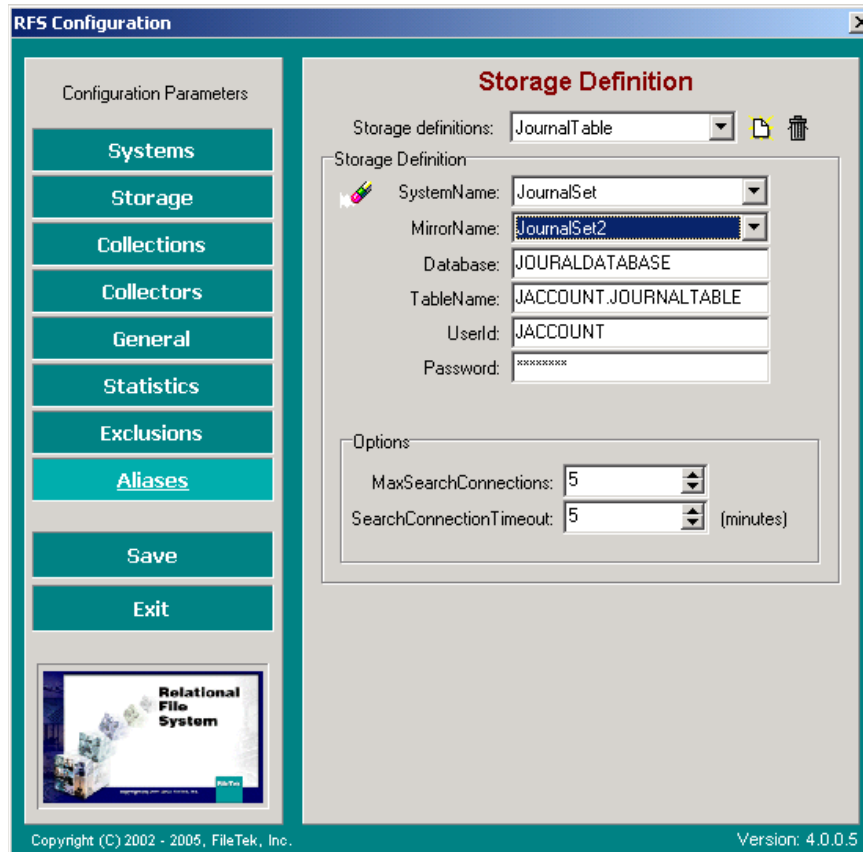
# Specifying system definition names in other definitions

For StorHouse/RFS duplexing, you must specify two system definition names—one for the primary StorHouse system (SystemName parameter) and one for the secondary StorHouse system (MirrorName parameter)—in the following definitions:

- Storage definition
- Statistics definition
- Aliases definition



For instance, in Figure 3-2, the JournalTable storage definition uses the JournalSet system definition for the primary StorHouse system and the JournalSet2 system definition for the secondary StorHouse system.



**Figure 3-2: Example of a storage definition for duplexing**



## Chapter 3 - Duplexing setup

In Figure 3-3, the example statistics definition uses the JournalSet system definition to identify the primary StorHouse system and the JournalSet2 system definition as the secondary StorHouse system to contain statistics.

The screenshot shows the 'RFS Configuration' window with the 'Statistics Definition' tab selected. On the left is a sidebar with 'Configuration Parameters' and a list of menu items: Systems, Storage, Collections, Collectors, General, Statistics, Exclusions, Aliases, and Exit. The main area contains the following settings:

- Statistics interval:** 5 (minutes)
- Output file type:** HTML: ☒ XML: ☒ TEXT: ☐
- Statistics database:**
  - Database: STATSDATABASE
  - TableName: SYSADM.STATS
  - UserId: SYSADM
  - Password: [masked]
  - SystemName: JournalSet
  - MirrorName: JournalSet2

At the bottom left is a small 'Relational File System' logo, and at the bottom right is the version '4.0.0.5'.

Figure 3-3: Example of a statistics definition for duplexing



In Figure 3-4, the example aliases definition uses the JournalSet system definition to identify the primary StorHouse system and the JournalSet2 system definition to identify the secondary StorHouse system to contain aliases.

The screenshot shows the 'RFS Configuration' window with the 'Aliases Definition' tab selected. The left sidebar contains a list of configuration parameters: Systems, Storage, Collections, Collectors, General, Statistics, Exclusions, Aliases, and Exit. The main area displays the 'Aliases database' configuration with the following fields:

- Database: ALIASES
- TableName: SYSADM.ALIASES
- UserId: ALIASES
- Password: [masked]
- SystemName: JournalSet (dropdown menu)
- MirrorName: JournalSet2 (dropdown menu)

At the bottom of the window, there is a copyright notice: 'Copyright (C) 2002 - 2005, FileTek, Inc.' and the version number: 'Version: 4.0.0.5'.

**Figure 3-4: Example of an aliases definition for duplexing**



## **Chapter 3 - Duplexing setup**



# Security

This chapter describes how to define and manage security in your StorHouse/RFS environment. Tasks include:

- Assigning permissions to StorHouse/RFS directories
- Managing aliases for user and group names
- Encrypting passwords (UNIX only)

Refer to the *StorHouse/RFS Concepts* for more information about how StorHouse/RFS implements security. See “Creating StorHouse tables” on page 2-20 for more information about creating a security table for storing security descriptors for Windows installations.



# Assigning permissions to StorHouse/RFS directories

With your standard operating system tools, assign permissions to the StorHouse/RFS directories as needed. Table 4-1 lists the StorHouse/RFS configuration file parameters you use to specify directories.

**Table 4-1: StorHouse/RFS directories**

Parameter	Contains	Number of directories	Requires read/write access
LocalPath	Statistics files	One directory per StorHouse/RFS configuration	StorHouse/RFS account and applicable users and administrators
LogFile	Log files	One directory per StorHouse/RFS configuration	StorHouse/RFS account and applicable administrators
CacheDir	Cache area for retrieving files on StorHouse	One directory per StorHouse/RFS configuration	StorHouse/RFS account
CollectionDir	Load files	Multiple directories, one in each collection definition	StorHouse/RFS account
StagingDir and UserDir	User files that are waiting to be collected	Multiple directories, one in each collector definition	StorHouse/RFS account and applicable users
SafetyPath	Duplicate copies of user file updates	One directory per storHouse/RFS configuration	StorHouse/RFS account

Additionally, assign permissions to the directory containing the StorHouse/RFS configuration file. The StorHouse/RFS account and an administrator account require access to the file.





## Managing aliases for user and group names

StorHouse/RFS permanently stores user and group names with the files on StorHouse. If a user or group name changes or is dropped, you can create an alias so that the new user or group can still access the files. For example, if Mary's login changes from mjones to msmith, in order to make sure she can access all of her files with both names, you must correlate the old and new name through alias checking.

To implement alias checking:

- Create an aliases database (or use an existing StorHouse database)
- Create the aliases table using the tblgen utility
- Set aliases parameters in the StorHouse/RFS configuration file
- Create the alias using a StorHouse/RFS-supplied HTML page

### Creating an aliases database

If you need to implement alias checking, you can use an existing StorHouse database or create a new one using StorHouse/Admin. In a StorHouse/RFS duplexing environment, create or use the same database on both the primary and secondary StorHouse systems. You also specify the name of the aliases database on the Database parameter in the aliases section of the StorHouse/RFS configuration file.

#### ▼ To create an aliases database

See “Creating a StorHouse database” on page 2-18 for instructions. You can use any unique database name.



### Creating an aliases table

StorHouse/RFS stores aliases in a StorHouse table in the system tablespace of the aliases database. You use the tblgen utility to create this table. In a StorHouse/RFS duplexing environment, you must create the aliases table on both the primary and secondary StorHouse systems.

When you run the tblgen utility, you specify the table owner, which is a StorHouse account. The minimum StorHouse privileges that the owner must have are SQLEXECUTE and RESOURCE.

#### ▼ To create an aliases table

See “Creating StorHouse tables” on page 2-20 for more information about creating StorHouse tables with the tblgen utility. Below is an example for creating an aliases table named SYSADM.GROUP in the ALIASES database on the alpha2 system.

```
tblgen
```

```
...
```

```
Enter the type of RFS tables to create.  
Choices are S - statistics,  
L - locator,  
A - aliases,  
B - security  
or combination without spaces: a
```

```
Enter the host name: alpha2
```

```
Enter the database name (case sensitive): ALIASES
```

```
Enter the base table name: GROUP
```

```
Enter the owner for the tables: SYSADM
```

```
Enter the owner password: *****
```



```
Creating alias table SYSADM.GROUP_ALIASES in database
ALIASES on alpha2
```

```
Continue? (Y or N) y
```

```
Aliases table successfully created
```

### Setting aliases parameters

The StorHouse/RFS configuration file contains an aliases section for identifying the storage resources to store aliases. You must enter values to implement alias checking. Note the following:

- The Database must be the StorHouse database containing the aliases table. This name is case sensitive.
- The TableName is the aliases table name in the format owner.base\_table\_name. This name must match the owner and table name provided on the tblgen utility.
- The UserId must be the owner of the StorHouse table specified at the TableName parameter or have the following minimum privileges: SQLEXECUTE StorHouse privilege and SELECT, INSERT, and UPDATE database component privileges on the aliases table.
- For UNIX, if you enter or change the Password parameter in a definition, you must run the gen\_cfg program after updating the StorHouse/RFS configuration file. See “Encrypting passwords” on page 4-11 for instructions on running the gen\_cfg program.
- In a StorHouse/RFS duplexing environment, specify a value for the MirrorName parameter.



## Chapter 4 - Security

Figure 4-1 illustrates an example of the aliases parameters displayed through the StorHouse/RFS configuration file utility.

The screenshot shows the 'RFS Configuration' window. On the left is a sidebar with 'Configuration Parameters' and a list of buttons: Systems, Storage, Collections, Collectors, General, Statistics, Exclusions, Aliases (highlighted), and Exit. Below the buttons is a small image of a 'Relational File System' box. The main area is titled 'Aliases Definition' and contains an 'Aliases database' section with a pencil icon. This section includes input fields for Database (ALIASES), TableName (SYSADM.GROUP), UserId (ALIASES), Password (masked with x's), SystemName (JournalSet), and MirrorName (JournalSet2). The bottom of the window shows 'Copyright (C) 2002 - 2005, FileTek, Inc.' and 'Version: 4.0.0.5'.

Figure 4-1: Example of an aliases definition



These are the same aliases parameters in the StorHouse/RFS configuration file.

```
[ALIASES]
Database=ALIASES
TableName=SYSADM.GROUP
UserId=SYSADM
Password=__)=>01
PasswordType=
SystemName=JournalSet
MirrorName=JournalSet2
```

The values in the example aliases parameters have the following meaning.

**Table 4-2: Descriptions of aliases parameters**

Parameter	Meaning
Database	Store aliases in the ALIASES database on StorHouse.
TableName	Store aliases in the SYSADM.GROUP_ALIASES table. (StorHouse/RFS adds _ALIASES to the table name.)
UserId	Use the SYSADM account to log in to StorHouse to access the SYSADM.GROUP_ALIASES table.
Password	Use the encrypted password to log in to StorHouse to access the SYSADM.GROUP_ALIASES table.
SystemName	Use the JournalSet system definition to identify the primary StorHouse system used to store aliases.
MirrorName	Use the JournalSet2 system definition to identify the secondary StorHouse system used to store aliases.

### ▼ To set aliases parameters with the StorHouse/RFS configuration file utility

1. If necessary, start the StorHouse/RFS configuration file utility.
2. In the RFS Configuration dialog box, click **Aliases**.
3. In the **Aliases database** area, enter or change any values.



4. Click Save.

### ▼ To set aliases parameters with a text editor in UNIX

1. If necessary, log in to the StorHouse/RFS server as root.
2. Open the `/rfs/files/rfs.cfg` file with the desired text editor. The following examples uses emacs.

```
emacs /rfs/files/rfs.cfg
```

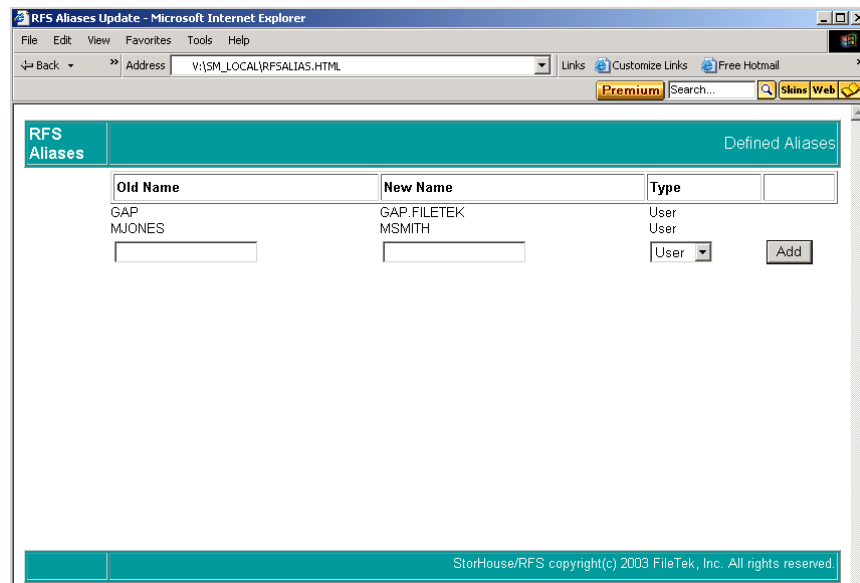
3. Go to the `[ALIASES]` section and enter or change values for the parameters.
4. Save the file.
5. Run the `gen_cfg` program to encrypt the password. See “Encrypting passwords” on page 4-11 for instructions on running the `gen_cfg` program.

## Creating or displaying an alias

You can create an alias by using a StorHouse/RFS HTML aliases page and specifying the old name, new name, and alias type (user or group). If you are creating aliases on a primary and secondary StorHouse system, both systems must be up or no aliases are accepted. This ensures both alias



databases are in sync. Figure 4-2 illustrates an example of an aliases HTML page with two aliases for users.



**Figure 4-2: Example of the aliases HTML page**

### ▼ To create or display aliases in Windows

1. Log in to the StorHouse/RFS server platform with the administrator account.
2. From the **Start** menu, click **Run**.
3. In the **Run** dialog box, type the following. (Replace V with the correct drive letter if your virtual file system is on a different drive.)

V:\sm\_local\rfsalias.html



## Chapter 4 - Security

4. Click **OK**.

StorHouse/RFS displays the aliases HTML page with any defined aliases.

5. In the **Old Name** box, type the old user or group name.
6. In the **New Name** box, type the new user or group name.
7. In the **Type** box, select **User** or **Group**.
8. Click **Add**.

### ▼ To create or display aliases in UNIX

1. Log in to the StorHouse/RFS server platform as root.
2. Access a Web browser, for instance, to access a Netscape browser, type:

```
/usr/dt/bin/netscape
```

3. Then type the following, in uppercase, to open the HTML page:

```
/RFS/SM_LOCAL/RFSALIAS.HTML
```

StorHouse/RFS displays the aliases HTML page with any defined aliases.

4. In the **Old Name** box, type the old user or group name.
5. In the **New Name** box, type the new user or group name.
6. In the **Type** box, select **User** or **Group**.
7. Click **Add**.





## Encrypting passwords

In a UNIX environment, whenever you need to add or change a Password value in system definitions, storage definitions, STATS section, and ALIASES section, you must run the `gen_cfg` program to encrypt the passwords in the StorHouse/RFS configuration file. The StorHouse/RFS configuration file utility encrypts passwords in a Windows environment.

The `gen_cfg` program has two parameters: the input `rfs.cfg` (required) and the output `rfs.cfg` (optional). The default output parameter is `/rfs/files/rfs.cfg`.

### ▼ To encrypt passwords

1. Log in to the StorHouse/RFS server as root.
2. Run the `gen_cfg` program.

```
./gen_cfg rfs.cfg
```

or

```
./gen_cfg rfs.cfg /rfs/files/rfs.cfg
```

3. Respond to the prompts by entering and then reentering the passwords. For example:

```
enter the password for user SYSADM in collection  
definition DS:
```

```
reenter the password:
```



## **Chapter 4 - Security**

## C H A P T E R

# 5



# Storage management

This chapter explains how to back up, recover, and remove StorHouse collections and StorHouse tables.



# Backing up StorHouse collections

You can back up one or more StorHouse collections on a primary VSET to either a backup or archive VSET. You can also schedule a backup to run automatically. The files on the primary VSET remain intact. To do this, you must:

- Create a backup or archive VSET
- Create a backup or archive FSET
- Specify the backup or archive VSET and FSET for the primary VSET
- Run or schedule a backup

Your FileTek customer support representative can help you develop a backup strategy for your data. This strategy should address how often to perform general backups as well as disaster protection options. You typically back up all of the StorHouse collections on a VSET or any new StorHouse collections written since the last backup. Refer to Chapter 9, “Protecting User Files and System Files,” in the *StorHouse System Administrator’s Guide* for complete information about backing up data on StorHouse and the difference between backup copies and archive copies. The tasks in this section explain how to create a backup copy of all StorHouse collections on a VSET and how to schedule a backup. Refer to the *StorHouse/Admin System Administrator’s Quick Reference* for additional procedures related to backing up files.

### ▼ To create a backup VSET

1. Log in to StorHouse/Admin.
2. In the folder list, click the VSETs folder.
3. On the **Edit** menu, click **Add/Create**.



4. In the **Create VSET** dialog box, type the name of the backup VSET.
5. In the **Library** list, click the library you want to use. And if the library has multiple media types, click the media you want to use in the **Media** list.
6. In the **Directory** list, click **Backup**.
7. Enter any other attributes as required.
8. Click **Create**, click **OK**, and then click **Done**.

### ▼ To create a backup FSET

1. In the VSETs list, open the folder for the backup VSET you created.
2. Click the FSETs folder.
3. On the **Edit** menu, click **Add/Create**.
4. In the **Create FSET** dialog box, enter the name of the backup FSET.
5. In the **In VSET** list, verify the name of the backup VSET that will contain the backup FSET.
6. Enter any other attributes as required.
7. Click **Create**, click **OK**, and then click **Done**.

### ▼ To specify the backup VSET and FSET for the primary VSET

1. Click the VSETs folder.
2. In the **Volume Sets** list, right-click the primary VSET name and click **Set**.




## Chapter 5 - Storage management

3. In the **Set VSET** dialog box, select the **Backup VSET** check box.
4. In the **Backup VSET** list, click the name of the backup VSET.
5. In the **Backup FSET** list, click the name of the backup FSET.
6. Click **Set**.

### ▼ To back up all StorHouse collections on a primary VSET

1. On the **Storage** menu, point to **File** and then click **Backup**.
2. In the **Backup** dialog box, enter \* in the **File name** box to back up all StorHouse collections on a VSET.
3. Select the **Group** box, and then select the group from the list.
4. Select the **VSET** box, and then select the VSET from the list.
5. Provide any optional information in the **Backup** dialog box.
6. Click **Create** to back up the files immediately.

### ▼ To schedule a backup of all StorHouse collections on a VSET

1. Perform steps 1 through 5 in the preceding procedure.
2. Click .
3. In the **Start Date** list, select a starting date.
4. In the **Start Time** list, select a starting time.
5. In the **Frequency** list, click how often you want the backup to run.



6. In the **Multiplier** box, enter a number if applicable.

For example, a multiplier of 1 for a Daily frequency backs up data once a day. A multiplier of 7 for a Daily frequency backs up data every 7 days.

7. Click **Schedule**.
8. Click **Done**.

## Backing up StorHouse tables

You can back up the StorHouse tables used by StorHouse/RFS. You can also schedule a backup for each StorHouse database and change that schedule when needed. For instance, you can schedule a daily backup for a journal database. Each time you run a backup, the StorHouse/RM *metadata backup utility* creates a backup file version for the specified database.

Note the following:

- The backup data for all StorHouse databases is stored in the same group, primary VSET, and primary FSET. A set of StorHouse system parameters specifies the names of these resources.
- If StorHouse/RFS is inserting file locator data into a StorHouse table, the metadata backup utility waits until the transaction completes. Likewise, if a metadata backup utility is running, StorHouse/RFS waits to insert the data into a StorHouse table until after the backup completes.
- You can specify the maximum number of backup file versions to keep for each database with a StorHouse system parameter called



## Chapter 5 - Storage management

SQL\_BKUP\_LIMIT. The default is 10. Only the latest file version is used to recover metadata.

Refer to Chapter 10, “Metadata backup” in the *StorHouse Database Administration Guide*, publication number 900108, for complete information about backing up StorHouse tables and other system files in a system tablespace. That chapter also contains error messages generated during a backup.

### ▼ To back up StorHouse tables

1. Log in to StorHouse/Admin.
2. In the folder list, click the **Databases** folder.
3. In the **Databases** list, right-click the database you want to back up, point to **Metadata**, and then click **Backup**.

The database icon must be green. If it is yellow, right-click the database name again.

4. In the **Metadata Backup** dialog box, click **Run backup now**.
5. Click **Done**.

### ▼ To schedule a backup for a StorHouse database

1. Log in to StorHouse/Admin.
2. In the folder list, click the **Databases** folder.
3. In the **Databases** list, right-click the database you want to back up, point to **Metadata**, and then click **Backup**.
4. In the **Start Date** list, select a starting date.





5. In the **Start Time** list, select a starting time.
6. In the **Frequency** list, click how often you want the backup to run.
7. In the **Multiplier** box, enter a number if applicable.

For example, a multiplier of 1 for a Monthly frequency backs up data once a month. A multiplier of 7 for a Daily frequency backs up data every 7 days.

8. Click **Schedule**.
9. Click **Done**.

## Deleting a collection set

If you no longer want to collect and store data for a collection set, you must remove all of the StorHouse storage and database resources. To do this, you must:

- Erase the collection set or uncatalog and export the collection set. You can erase data from erasable optical or magnetic media. You must uncatalog and export data on non-erasable optical or tape.
  - *Uncatalog* removes directory information for all files and FSETs in the VSET.
  - *Export* moves each uncataloged volume to the exchange station for an operator to unload and remove from StorHouse.



## Chapter 5 - Storage management

- Drop the StorHouse database and table array. Contact your FileTek customer support representative for more information about removing StorHouse database resources.
- Delete the system definition, storage definition, collection definition, and collector definition from the StorHouse/RFS configuration file.

### ▼ To uncatalog a VSET

1. Log in to StorHouse/Admin.
2. Click the **VSETs** folder.
3. In the **Volume Sets** list, right-click the VSET you want to uncatalog, and click **Uncatalog**.
4. In the **Uncatalog VSET** check box, select the **Wait for files to be available** check box if you want StorHouse to wait for any locked files to be unlocked before uncataloging the volume set.
5. Select the **Do not uncatalog if any files are found** check box if you want to prevent StorHouse from removing directory information for any files on volumes in the VSET. (StorHouse leaves these volumes in the uncataloged state.)
6. Click **Yes** to uncatalog the VSET immediately.

### ▼ To export a VSET

1. Log in to StorHouse/Admin.
2. Click the **VSETs** folder.
3. In the **Volume Sets** list, right-click the VSET from which you want to export all uncataloged volumes, and click **Export**.




4. In the **Export VSET** dialog box, click **Export uncataloged volumes only and retain directory information for the volume set**.
5. Click **Yes**.

### ▼ To erase a VSET




1. Log in to StorHouse/Admin.
2. Click the **VSETs** folder.
3. In the **Volume Sets** list, right-click the VSET you want to erase and then click **Erase**.
4. In the **Erase VSET** dialog box, click **Yes** to erase the VSET immediately.
5. Repeat steps 3 and 4 to erase other VSETs for the collection set.

### ▼ To delete a system, storage, collection, and collector definition with the StorHouse/RFS configuration file utility

1. Log in to the StorHouse/RFS server platform with the administrator account.
2. Start the StorHouse/RFS configuration file utility.
3. Delete the system definition as follows:
  - a. In the **Configuration Parameters** area, click **Systems**.
  - b. In the **Defined systems** list, select the system definition to delete.
  - c. Click .



## Chapter 5 - Storage management

4. Delete the storage definition as follows:
  - a. In the **Configuration Parameters** area, click **Storage**.
  - b. In the **Storage definitions** list, select the storage definition to delete.
  - c. Click .
5. Delete the collection definition as follows:
  - a. In the **Configuration Parameters** area, click **Collections**.
  - b. In the **Defined tables** list, select the collection definition to delete.
  - c. Click .
6. Delete the collector definition as follows:
  - a. In the **Configuration Parameters** area, click **Collectors**.
  - b. In the **Defined collectors** list, select the collector definition to delete.
  - c. Click .
7. Click **Exit**.



# Recovering StorHouse collections

If a StorHouse collection is unreadable, you can recover it from the backup or archive copy. To do this, you must:

- Delete and remove the unreadable primary file
- Create a primary file from the backup or archive copy

You need the name of the file to remove it. Refer to Chapter 10, “Recovering User Files and System Files,” in the *StorHouse System Administrator’s Guide* for complete information about recovering file data on StorHouse.

## ▼ To delete and remove an unreadable primary file

1. Log in to StorHouse/Admin.
2. On the **Storage** menu, point to **File** and then click **Search**.
3. In the **File name** box, accept the default \*.
4. In the **Group** list, click the group that contains the file.
5. In the **VSET** list, click the VSET that contains the file.
6. Click **Find**.
7. In the **Files** list, click the file you want to delete.
8. On the **Edit** menu, click **Delete/Drop**.
9. In the **Delete** dialog box, select the **Wait for locked files to be available check box** if you want StorHouse to wait for a locked file to be unlocked before attempting to delete that file.



10. Click Yes.

### ▼ To create a primary file

1. On the **Storage** menu, point to **File** and then click **Create Primary**.
2. In the **Create Primary** dialog box, click **File name** or **File ID**, depending on which you want to specify.
3. Type the name or ID of the file you want to recover in the **File name/ File ID** box.
4. Provide any optional information in the **Create Primary** dialog box.
5. Click **Create**.

## Recovering StorHouse tables

You can restore the StorHouse tables in a system tablespace should the magnetic disks fail. The following utilities recover tables used by StorHouse/RFS:

- StorHouse/RM metadata restore utility
- StorHouse/RM redo journaling utilities
- StorHouse/RFS rfsrestore utility

The StorHouse/RM *metadata restore utility* restores the data from the most recent metadata backup file. You can run this utility only if you ran the metadata backup utility described on page 4-5. Refer to the *StorHouse Database Administration Manual* for instructions on running the metadata restore utility. Contact your FileTek customer support representative for assistance.



The StorHouse/RM *redo journaling utilities* available with StorHouse/RM release 3.3 and later capture transactions (for instance, inserts and updates) since the last metadata backup and apply, or replay, the cycled and archived transactions when needed. Refer to Chapter 12, “Redo journaling,” in the *StorHouse Database Administration Guide* for more information about redo journaling utilities.

The StorHouse/RFS *rfsrestore utility* identifies StorHouse collections created since the last metadata backup, extracts the load data files (containing file locator data) from the collections, and re-loads the file locator data into the StorHouse tables. Contact your FileTek customer support representative for more information about the *rfsrestore* utility.



## **Chapter 5 - Storage management**



## C H A P T E R

# 6



# Statistics

This chapter describes the statistics that StorHouse/RFS can capture and store and the setup procedures you perform to store statistics in a local statistics file, a StorHouse database, or both. If your environment is configured for StorHouse/RFS duplexing, you can also store statistics on both the primary and secondary StorHouse systems.



# About statistics

StorHouse/RFS can optionally capture and store general statistics as well as statistics for each collection definition and system definition.

## General statistics

*General statistics* can help you monitor and manage staging and cache space. Table 6-1 lists the general statistics captured at each configured statistics interval.

**Table 6-1: General statistics**

Statistic	Description
Sample Date	Time and date when StorHouse/RFS created the statistics record
Sample GMT	Time and date (Greenwich Mean Time) when StorHouse/RFS created the statistics record
Uptime	Number of seconds since StorHouse/RFS was started
Interval Type	Type of record: <ul style="list-style-type: none"><li>■ Startup – First record written after StorHouse/RFS is started</li><li>■ Shutdown – Record written when StorHouse/RFS is stopped gracefully</li><li>■ Interval – Record written during the previous statistics interval</li><li>■ Current – Record written since the previous statistics interval to the current time</li></ul>
Interval Time	Length (in seconds) of the statistics interval
Staging Space Defined	Value (in MB) of the MaxStagingSpace (RFS) parameter
Staging Space Free	Amount of free space (in MB) in the staging areas



Table 6-1: General statistics (continued)

Statistic	Description
Cache Space Defined	Value (in MB) of the MaxCacheSpace parameter in the RFS section
Cache Space Free	Amount of free space (in MB) in the cache directory
SM Connections Defined	Value of the MaxSMFiles parameter in the RFS section
SM Connections in Use	Number of StorHouse file connections in use
SM Limit Errors	Number of times the MaxSMFiles (RFS section) parameter was reached during an interval
SM Unavailable Errors	Number of times StorHouse was unavailable during the interval

## Collection definition statistics

*Collection definition statistics* can help you assess and manage collection activity, search activity, and local collection space. Table 6-2 lists the collection definition statistics captured at each configured statistics interval.

Table 6-2: Collection definition statistics

Statistic	Description
Collections written	Number of collections that have been written to StorHouse at the time the statistics record was written
Collections unwritten	Number of collections that have not been written to StorHouse at the time the statistics record was written
Collection Space Defined	Value (in MB) of the MaxCollectionSpace parameter in a collector definition
Collection Space Free	Amount of free space (in MB) in the rename and collection directories
Collection Space Written	Amount of space (in MB) in the rename and collection directories used by collections written to StorHouse



## Chapter 6 - Statistics

**Table 6-2: Collection definition statistics (continued)**

Statistic	Description
Collection Space Unwritten	Amount of space (in MB) in the rename and collection directories used by collections not yet written to StorHouse
Files Collected	Number of files collected during the statistics interval
Uncollected Files	Number of files not yet collected as of the end of the interval
MB Collected	Amount (in MB) of data collected
Collection Rate	Rate (in MB per second) that files were collected
ODBC Connection Limit	Value of the MaxSearchConnections parameter in a storage definition
ODBC Connections in Use	Number of ODBC connections in use as of the end of the interval
Searches Completed	Number of searches completed during the interval (a search may have started in a prior interval)
Average Search Time	Average time in seconds for a search that completed during the interval
Files Found	Number of files returned for searches that completed during the interval
Directories Found	Number of directories returned for searches that completed during the interval
Loads completed	Number of loads completed during the interval
Database Rows Loaded	Number of files loaded into the database for loads that completed during the interval
Database Load Rate	Number of rows per second loaded into the database for loads that completed during the interval
MB Read from Local Collections	Amount (in MB) of data read from local collections during the interval



## System definition statistics

*System definition statistics* can help you monitor StorHouse writes and reads. Table 6-3 lists the system definition statistics captured at each configured statistics interval.

**Table 6-3: System definition statistics**

Statistic	Description
MB Written	Amount (in MB) written to StorHouse for writes that completed during the interval (a file write could begin in a prior interval)
Write Rate	Rate (in MB per second) for writes that completed during the interval
Writes Completed	Number of StorHouse writes completed during the interval
MB Read from StorHouse	Amount (in MB) read from StorHouse during the interval
Collections Accessed	Number of StorHouse collections accessed during the interval
MB Read from Cache	Amount (in MB) read from cache or the current record returned from StorHouse

## Ways to store statistics

You can configure StorHouse/RFS to store statistics in the following ways:

- In a local statistics file only
- In a StorHouse database only
- In both a local statistics file and a StorHouse database
- All of the above on both the primary and secondary StorHouse systems for StorHouse/RFS duplexing



### Local statistics file

StorHouse/RFS can provide statistics in a file located in the path specified at the LocalPath parameter in the General, or RFS section, of the StorHouse/RFS configuration file. You choose the file format: HTML, XML, text, or any combination. For example, you can request a local statistics file in both HTML and XML formats, only text format, or all formats. StorHouse/RFS updates the file(s) at a configured interval, replacing the statistics from the previous interval. Therefore, the local statistics file in the LocalPath location contains statistics from the previous statistics interval. The file names are RFSSTATS.HTML, RFSSTATS.XML, and RFSSTATS.TXT. See “Displaying local statistics from the previous interval” on page 6-18 for instructions on accessing these files.

StorHouse/RFS also maintains the current statistics (from the previous statistics interval to the current time) in a text, HTML, or XML file in a hidden directory on the virtual file system. This file is named RFSSTATSNOW.TXT, RFSSTATSNOW.HTML, or RFSSTATSNOW.XML. You can display the current statistics file by accessing the virtual file system and navigating to the hidden directory. See “Displaying the current statistics” on page 6-20 for instructions.

Each time you stop the StorHouse/RFS server, StorHouse/RFS saves the statistics file, appending a date and time extension to the statistics file name. When you start the StorHouse/RFS service, StorHouse/RFS creates a new statistics file. The LocalPath location then may contain multiple statistics files.



Figure 6-2 illustrates an example of a local statistics file in text format.

```

RFS Computer Name..... DEV-ELF.FILETEK.COM
Sample Date..... 2006-01-05 13:59:15
Sample GMT..... 2006-01-05 18:59:15
Uptime..... 0 days 0 hours 2 min 17 secs
Interval Type..... Current
Interval Time (seconds)..... 134
Staging Space Defined (MB)..... 18000
Staging Space Free (MB)..... 18000
Cache Space Defined (MB)..... 499
Cache Space Free (MB)..... 0
SM Connections Defined..... 8
SM Connections In Use..... 0
SM Limit Errors..... 0
SM Unavailable Errors..... 0
-----
Collection Definition..... DIRECT
Collections Written..... 0
Collections Unwritten..... 0
Collection Space Defined (MB).... 1000
Collection Space Free (MB)..... 1000
Collection Space Written (MB).... 0
Collection Space Unwritten (MB).. 0
Files Collected..... 0
Uncollected Files..... 0
MB Collected..... 0.000
Collection Rate (MB/second)..... 0.000
ODBC Connection Limit..... 5
ODBC Connections in Use..... 0
Searches Completed..... 0
Average Search Time (seconds).... 0.000
Files Found..... 0
Directories Found..... 0
Database Loads Completed..... 0
Database Rows Loaded..... 0
Database Load Rate (rows/second). 0.000
MB Read from Local Collections... 0.000
-----
System Definition..... sthr
Writes Completed..... 1
MB Written..... 1.071
Write Rate (MB/second)..... 0.094
MB Read From StorHouse..... 0.000
MB Read From Cache..... 0.000
-----

```

Figure 6-1: Example of a local statistics file in text format



## Chapter 6 - Statistics

Figure 6-2 illustrates an example of a local statistics file in XML format.

```
- <STATS>
<RFS_NAME>DEV-ELF.FILETEK.COM</RFS_NAME>
<STATS_DATE>2005-03-31 16:26:33</STATS_DATE>
<STATS_GMT>2005-03-31 21:26:33</STATS_GMT>
<UP_TIME_SECS>10</UP_TIME_SECS>
<INTERVAL_TYPE>Current</INTERVAL_TYPE>
<INTERVAL_TIME>5</INTERVAL_TIME>
<STAGING_DEF>17450</STAGING_DEF>
<STAGING_FREE>17449</STAGING_FREE>
<CACHE_DEF>99</CACHE_DEF>
<CACHE_FREE>0</CACHE_FREE>
<SM_CON_DEF>8</SM_CON_DEF>
<SM_CON_USED>0</SM_CON_USED>
<LIMIT_ERRORS>0</LIMIT_ERRORS>
<AVAIL_ERRORS>0</AVAIL_ERRORS>
- <TABLE>
<TABLE_DEF>DIRECT</TABLE_DEF>
<COLL_WRITTEN>0</COLL_WRITTEN>
<COLL_UNWRITTEN>0</COLL_UNWRITTEN>
<COLL_SPC_DEF>10800</COLL_SPC_DEF>
<COLL_SPC_FREE>10800</COLL_SPC_FREE>
<COLL_SPC_WRITTEN>0</COLL_SPC_WRITTEN>
<COLL_SPC_UNWRITTEN>0</COLL_SPC_UNWRITTEN>
<FILES_COLLECTED>0</FILES_COLLECTED>
<FILES_UNCOLLECTED>0</FILES_UNCOLLECTED>
<MB_COLLECTED>0.000</MB_COLLECTED>
<MB_COLL_RATE>0.000</MB_COLL_RATE>
<ODBC_CON_LIMIT>5</ODBC_CON_LIMIT>
<ODBC_CON_USED>0</ODBC_CON_USED>
<SEARCHES_COMPL>0</SEARCHES_COMPL>
<AVG_SEARCH_TIME>0.000</AVG_SEARCH_TIME>
<FILES_FOUND>0</FILES_FOUND>
<DIRS_FOUND>0</DIRS_FOUND>
<LOADS_COMPLETED>0</LOADS_COMPLETED>
<FILES_LOADED>0</FILES_LOADED>
<LOAD_RATE>0.000</LOAD_RATE>
<MB_LOCAL_READ>0.000</MB_LOCAL_READ>
</TABLE>
- <SYSTEM>
<SYSTEM_DEF>sthr</SYSTEM_DEF>
<WRITES_COMPLETED>0</WRITES_COMPLETED>
<MB_WRITTEN>0.000</MB_WRITTEN>
<MB_WRITE_RATE>0.000</MB_WRITE_RATE>
<COLLECTIONS_ACCESSED>0</COLLECTIONS_ACCESSED>
<MB_STH_READ>0.000</MB_STH_READ>
<MB_CACHE_READ>0.000</MB_CACHE_READ>
</SYSTEM>
</STATS>
```

**Figure 6-2: Example of a local statistics file in XML format**





## Statistics database

In addition to capturing statistics in a local statistics file, StorHouse/RFS can write statistics records in a statistics database on StorHouse at each configured interval. StorHouse/RFS uses ODBC to write a statistics record in three statistics tables (where you provide the owner.base\_table\_name when running the tblgen utility):

- The owner.base\_table\_name\_RFS table contains general statistics.
- The owner.base\_table\_name\_TABLES table contains collection definition statistics.
- The owner.base\_table\_name\_SYSTEMS table contains system definition statistics.

Specifically, at the statistics interval, StorHouse/RFS writes one row in the \_RFS table, one row in the \_TABLES for each collection definition, and one row in the \_SYSTEMS table for each system definition. StorHouse/RFS also writes these rows when StorHouse/RFS is stopped and started.

You can query these statistics tables to obtain data from the last interval or previous intervals. For example, you could query all rows in the \_RFS table to determine staging and cache space usage over time. Or you could query the \_TABLES table for a specific collection definition to obtain collection and retrieval activity for a specific collection set. If you choose to store statistics in a StorHouse database, you must create the database (or use an existing one) and the statistics tables. See “Creating a statistics database” on page 6-11 for more information.

## Statistics duplexing

If your enterprise is configured for StorHouse/RFS duplexing, you can store statistics on both the primary and secondary StorHouse systems. This provides a secondary copy of the statistics. If either system is down



during an attempt to write statistics to the database, StorHouse/RFS holds the old values that would be written to the down system and continues to accumulate information so that a complete but large interval can be written when the StorHouse system becomes available.

**Caution:** StorHouse/RFS does not retain the statistics interval values in non-volatile storage. Should StorHouse/RFS go down for any reason while any StorHouse system is not available, the normal shutdown statistics record is not written to the down system and the interval statistics being held are lost.

In order to duplicate the statistics on the secondary system, you must provide a value for the MirrorName parameter in the statistics definition and create the same StorHouse database and statistics tables on the primary and secondary systems.

### Statistics interval

StorHouse/RFS captures statistics at an interval (in minutes) that you set with the StatsInterval parameter in the StorHouse/RFS configuration file. The default value is 10 minutes. StorHouse/RFS does not store statistics if this value is 0 or omitted from the StorHouse/RFS configuration file.

When the interval expires, StorHouse/RFS:

- Updates the statistics in the local statistics file
- Writes statistics records in the statistics tables on StorHouse

The statistics files and records contain the interval time, which may be less than the value of the StatsInterval parameter. Here's why. StorHouse/RFS writes statistics when the StorHouse/RFS service is started, when it is stopped gracefully, and during the normal statistics interval. The interval type—Startup, Shutdown, Interval, and Current—in the statistics file or record identifies when StorHouse/RFS wrote the



record. For example, the Interval Type is Shutdown in the sample statistics file in Figure 6-1 and Figure 6-2 and the Interval Seconds are 27.

The interval time in a Startup record is always zero seconds. The statistics in a Startup record include only those values defined in the StorHouse/RFS configuration file and any statistics based on local collections recovered during start up. The first interval record after the startup record may be shorter than the StatsInterval because StorHouse/RFS adjusts the timer to run by the clock. For example, if the StatsInterval is 5 minutes and StorHouse/RFS starts at 10:13:57, the first interval would finish at 10:15:00 and would only be 63 seconds long. Each subsequent Interval record would then show an interval time of approximately 300 seconds.

## **Creating a statistics database**

If you want to store statistics on StorHouse, you must create a StorHouse database (or use an existing one) using StorHouse/Admin. In a StorHouse/RFS duplexing environment, you must create or use the same database on both the primary and secondary StorHouse systems.

### **▼ To create a statistics database**

See “Creating a StorHouse database” on page 2-18 for instructions. You can use any unique database name.

## **Creating statistics tables**

You use the StorHouse/RFS tblgen utility to create the three statistics tables. When you run the tblgen utility, you specify the table owner,



## Chapter 6 - Statistics

which is a StorHouse account. The minimum StorHouse privileges that owner must have are SQLEXECUTE and RESOURCE.

### ▼ To create statistics tables

See “Creating StorHouse tables” on page 2-20 for more information about creating StorHouse tables with the tblgen utility. Below is an example for creating statistics tables with a base table name of SYSADM.STATS in the STATSDATABASE database on the alpha2 system. The utility will create three statistics tables: SYSADM.STATS\_RFS, SYSADM.STATS\_TABLES, and SYSADM.STATS\_SYSTEMS.

```
tblgen
```

```
...
```

```
Enter the type of RFS tables to create.
```

```
Choices are S - statistics,
```

```
L - locator,
```

```
A - aliases,
```

```
B - security
```

```
or combination without spaces: s
```

```
Enter the host name: alpha2
```

```
Enter the database name (case sensitive): STATSDATABASE
```

```
Enter the base table name: STATS
```

```
Enter the owner for the tables: SYSADM
```

```
Enter the owner password: *****
```

```
Creating statistics tables SYSADM.STATS in database  
STATSDATABASE on alpha2
```

```
Continue? (Y or N) y
```

```
Statistics tables successfully created
```



## Setting statistics parameters

The StorHouse/RFS configuration file contains a statistics or STATS section for setting the statistics interval and identifying the storage resources to store statistics. Note the following:

- If you do not want to capture statistics, specify 0 for the StatsInterval parameter.
- If you want to store statistics in a local file, specify a value for the FileType parameter.
- If you want to store statistics in a StorHouse database, specify values for the DSNName, Database, TableName, UserId, and Password parameters.
- The TableName is the base table name in the format owner.base\_table\_name. This name must match the table name specified on the tblgen utility to create the statistics tables.
- The UserId must be the owner of the StorHouse table specified at the TableName parameter or have the following minimum privileges: SQLEXECUTE StorHouse privilege and INSERT database component privilege on the statistics tables.
- For UNIX, if you enter or change the Password parameter in a definition, you must run the gen\_cfg program after updating the StorHouse/RFS configuration file. See “Encrypting passwords” on page 4-11 for instructions on running the gen\_cfg program.
- If you want to store a copy of the statistics on a secondary StorHouse system, specify a value for the MirrorName parameter.

The StatsInterval parameter is dynamic, that is, the change takes effect immediately after you request StorHouse/RFS to reread the



## Chapter 6 - Statistics

StorHouse/RFS configuration file. The other statistics parameters may be static or dynamic. For instance, assume you decide not to use statistics. You set the StatsInterval to 0 and request StorHouse/RFS to reread the file. StorHouse/RFS immediately stops capturing statistics. Later, you change the StatsInterval to 5 minutes and request a reread. StorHouse/RFS immediately reads all of the statistics parameters. In this case, all of the statistics parameters are dynamic. On the other hand, if you are using statistics and just change the Password, then that change takes effect after you restart the StorHouse/RFS service. In this case, the parameter is static.



Figure 6-3 illustrates an example statistics definition displayed with the StorHouse/RFS configuration file utility.

The screenshot shows the 'RFS Configuration' window with the 'Statistics Definition' tab selected. The left sidebar contains a list of configuration parameters: Systems, Storage, Collections, Collectors, General, **Statistics**, Exclusions, Aliases, Save, and Exit. The main area displays the following settings:

- Statistics interval:** 5 (minutes)
- Output file type:** HTML: ☒ XML: ☒ TEXT: ☐
- Statistics database:**
  - Database: STATSDATABASE
  - TableName: SYSADM.STATS
  - UserId: SYSADM
  - Password: [masked]
  - SystemName: JournalSet
  - MirrorName: <none>

At the bottom left is a logo for 'Relational File System' and at the bottom right is the text 'Version: 4.0.0.5'.

**Figure 6-3: Example of a statistics definition**



## Chapter 6 - Statistics

These are the same statistics parameters in the StorHouse/RFS configuration file.

```
[STATS]
StatsInterval=5
FileType=HTML XML
Database=STATSDATABASE
TableName=SYSADM.STATS
UserId=SYSADM
Password=^u=
PasswordType=
SystemName=JournalSet
MirrorName=
```

The values in the sample statistics definition have the following meaning.

**Table 6-4: Descriptions of statistics parameters**

Parameter	Meaning
StatsInterval	Write statistics every 5 minutes.
FileType	Create local statistics files in HTML and XML formats.
Database	Store statistics in the STATSDATABASE on StorHouse.
TableName	Store statistics in the StorHouse tables owned by SYSADM and with names that start with the base table name of STATS.
UserId	Use the SYSADM account to log in to StorHouse to access the statistics tables in the STATSDATABASE.
Password	Use the encrypted password to log in to StorHouse to access the statistics tables in the STATSDATABASE.
SystemName	Use the JournalSet system definition to identify the primary StorHouse system to contain statistics.
MirrorName	Do not store a duplicate copy of the statistics on a secondary StorHouse system.





### ▼ To set statistics parameters with the StorHouse/RFS configuration file utility

1. If necessary, start the StorHouse/RFS configuration file utility.
2. In the **RFS Configuration** dialog box, click **Statistics**.
3. Enter or change any values on the **Statistics Definition** page.
4. Click **Save**.
5. If you changed only the StatsInterval parameter, request StorHouse/RFS to reread the StorHouse/RFS configuration file (see page 7-7). If you specified or changed any other parameter, restart the StorHouse/RFS service (see page 7-8).

### ▼ To set statistics parameters with a text editor in UNIX

1. If necessary, log in to the StorHouse/RFS server as root.
2. Open the `/rfs/files/rfs.cfg` file with the desired text editor. The following examples uses emacs.

```
emacs /rfs/files/rfs.cfg
```

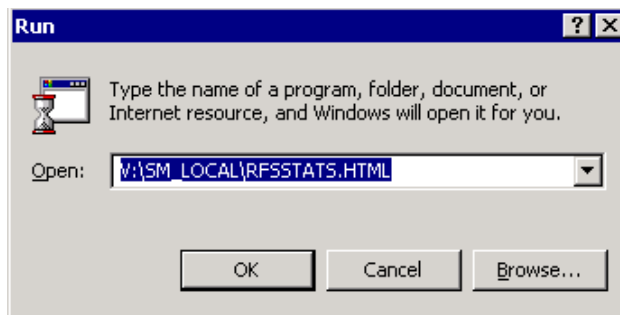
3. Go to the [STATS] section and enter or change values for the parameters.
4. Save the file.
5. Run the `gen_cfg` program to encrypt the password. See “Encrypting passwords” on page 4-11 for instructions on running the `gen_cfg` program.



6. If you changed only the StatsInterval parameter, request StorHouse/RFS to reread the StorHouse/RFS configuration file (see page 7-7). If you specified or changed any other parameter, restart the StorHouse/RFS service (see page 7-8).

## Displaying local statistics from the previous interval

If you selected an output file type (FileType parameter) in the Statistics, or STATS, section of the StorHouse/RFS configuration file, then you can display a local statistics file containing statistics from the previous interval. The local statistics file is located in the path specified on the LocalPath (RFS) parameter in the StorHouse/RFS configuration file. You can navigate to that path, or you can display the file in the SM\_LOCAL hidden directory on the virtual file system. Figure 6-4 illustrates how to access the previous statistics file (in HTML format) in a Windows environment by using Run from the Start menu.



**Figure 6-4: Displaying previous statistics in Windows**



### ▼ To display the previous statistics in Windows

1. Log in to the StorHouse/RFS server platform with the administrator account.
2. From the **Start** menu, click **Run**.
3. In the **Run** dialog box, type one of the following. (Replace V with the correct drive letter if your virtual file system is on a different drive.)

V:\SM\_LOCAL\RFSSTATS.HTML

V:\SM\_LOCAL\RFSSTATS.TXT

V:\SM\_LOCAL\RFSSTATS.XML

4. Click **OK**.
5. If the file is in HTML format, click **Refresh** as needed to update the statistics when the StatsInterval expires. If the file is in text or XML formats, close and open the file to update the statistics.

### ▼ To display the previous statistics in UNIX

1. Log in to the StorHouse/RFS server platform as root.
2. Type the following to display the statistics in text format:

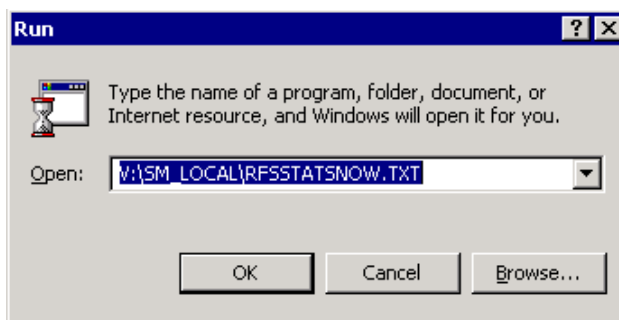
```
cat /RFS/SM_LOCAL/RFSSTATS.TXT
```

Or you can display the statistics in HTML format by using a Web browser (/usr/dt/bin/netscape) and opening the RFSSTATS file.



# Displaying the current statistics

You can display the current statistics (those captured since the previous interval to the current time) in a text, HTML, or XML file located in the SM\_LOCAL hidden directory on the virtual file system. Figure 6-5 illustrates how to access the current statistics file (in text format) in a Windows environment by using Run from the Start menu.



**Figure 6-5: Displaying current statistics in a Windows environment**

### ▼ To display the current statistics in Windows

1. Log in to the StorHouse/RFS server platform with the administrator account.
2. From the **Start** menu, click **Run**.
3. In the **Run** dialog box, type one of the following. (Replace V with the correct drive letter if your virtual file system is on a different drive.)

V:\SM\_LOCAL\RFSSTATSNOW.HTML

V:\SM\_LOCAL\RFSSTATSNOW.TXT

V:\SM\_LOCAL\RFSSTATSNOW.XML



4. Click OK.

### ▼ To display the current statistics in UNIX

1. Log in to the StorHouse/RFS server platform as root.
2. Type the following to display the statistics in text format:

```
cat /RFS/SM_LOCAL/RFSSTATSNOW.TXT
```

Or you can display the statistics in HTML format by using a Web browser (/usr/dt/bin/netscape) and opening the RFSSTATSNOW file.

## Querying statistics in the statistics database

You can use the StorHouse/Admin ISQL facility to display statistics in the statistics database. Refer to the *StorHouse/Admin Database Administrator's Guide* for instructions on using ISQL.



## Chapter 6 - Statistics

Figure 6-6 lists the column names in the three statistics tables.





















































<b>_SYSTEMS table</b>		<b>_RFS table</b>		<b>_TABLES table</b>	
Column Name		Column Name		Column Name	
 RFS_NAME		 RFS_NAME		 RFS_NAME	
 STATS_DATE		 STATS_DATE		 STATS_DATE	
 STATS_GMT		 STATS_GMT		 STATS_GMT	
 INTERVAL_TIME		 INTERVAL_TIME		 INTERVAL_TIME	
 INTERVAL_TYPE		 INTERVAL_TYPE		 INTERVAL_TYPE	
 SYSTEM_DEF		 UP_TIME_SECS		 TABLE_DEF	
 WRITES_COMPL		 STAGING_DEF		 COLL_WRITTEN	
 MB_WRITTEN		 STAGING_FREE		 COLL_UNWRITTEN	
 MB_WRITE_RATE		 CACHE_DEF		 COLL_SPC_DEF	
 COLL_ACCESSED		 CACHE_FREE		 COLL_SPC_FREE	
 MB_STH_READ		 SM_CON_DEF		 COLL_SPC_WRITTEN	
 MB_CACHE_READ		 SM_CON_USED		 COLL_SPC_UNWRITTEN	
		 LIMIT_ERRORS		 FILES_COLLECTED	
		 AVAIL_ERRORS		 FILES_UNCOLLECTED	
				 MB_COLLECTED	
				 MB_COLL_RATE	
				 ODBC_CON_LIMIT	
				 ODBC_CON_USED	
				 SEARCHES_COMPL	
				 AVG_SEARCH_TIME	
				 FILES_FOUND	
				 DIRS_FOUND	
				 LOADS_COMPL	
				 FILES_LOADED	
				 LOAD_RATE	
				 MB_LOCAL_READ	

Figure 6-6: Column names in the statistics tables



Table 6-5 lists query examples.

**Table 6-5: Sample queries to obtain statistics**

To display	SQL statement
All general statistics	SELECT * FROM SYSADM.STATS_RFS;
All collection definition statistics	SELECT * FROM SYSADM.STATS_TABLES;
All system definition statistics	SELECT * FROM SYSADM.STATS_SYSTEMS;
Amount of data written for the JOURNALSET system definition	SELECT MB_WRITTEN FROM SYSADM.STATS_SYSTEMS WHERE SYSTEM_DEF='JOURNALSET';
Amount of data read from StorHouse for the JOURNALSET system definition	SELECT MB_STH_READ FROM SYSADM.STATS_SYSTEMS WHERE SYSTEM_DEF='JOURNALSET';
Number of files collected for the COMPLYTABLE storage definition	SELECT FILES_COLLECTED FROM SYSADM.STATS_TABLES WHERE TABLE_DEF='COMPLYTABLE';
Number of searches completed and average search time for the JOURNALTABLE storage definition	SELECT SEARCHES_COMPL, AVG_SEARCH_TIME FROM SYSADM.STATS_TABLES WHERE TABLE_DEF='JOURNALTABLE';

Note that the ISQL facility returns only the first 100 rows. FileTek recommends that you filter a result set by specifying a date or time.



## Chapter 6 - Statistics

Figure 6-7 illustrates an example of a result set with the Interval records for a collection definition called DIRECT.

Enter SQL statements:

```
select * from sysadm.stats tables  
where INTERVAL_TYPE='Interval' and TABLE_DEF='DIRECT' ;
```

RFS_NAME	STATS_DATE	INTERVAL_TIME	INTERVAL_TYPE	TABLE_DEF	COLL_WRITTEN	COLL_UNWRITTEN	COLL_SPC_DEF
DEVSELF FILETEK.COM	2002-11-11 14:27:00.000000	19	Interval	DIRECT	0	0	250
DEVSELF FILETEK.COM	2002-11-11 14:30:00.000000	17	Interval	DIRECT	0	0	250
DEVSELF FILETEK.COM	2002-11-12 11:06:00.000000	97	Interval	DIRECT	0	0	250
DEVSELF FILETEK.COM	2002-11-12 11:36:01.000000	79	Interval	DIRECT	0	1	250
DEVSELF FILETEK.COM	2002-11-12 11:39:00.000000	179	Interval	DIRECT	0	1	250
DEVSELF FILETEK.COM	2002-11-12 11:42:00.000000	180	Interval	DIRECT	0	2	250
DEVSELF FILETEK.COM	2002-11-12 11:45:00.000000	180	Interval	DIRECT	1	1	250
DEVSELF FILETEK.COM	2002-11-12 11:48:00.000000	180	Interval	DIRECT	2	0	250
DEVSELF FILETEK.COM	2002-11-12 11:51:00.000000	180	Interval	DIRECT	2	1	250
DEVSELF FILETEK.COM	2002-11-12 11:54:03.000000	183	Interval	DIRECT	2	2	250
DEVSELF FILETEK.COM	2002-11-12 11:57:00.000000	177	Interval	DIRECT	3	1	250
DEVSELF FILETEK.COM	2002-11-12 12:00:00.000000	180	Interval	DIRECT	4	0	250
DEVSELF FILETEK.COM	2002-11-12 12:03:00.000000	180	Interval	DIRECT	4	0	250
DEVSELF FILETEK.COM	2002-11-12 12:06:00.000000	180	Interval	DIRECT	4	0	250
DEVSELF FILETEK.COM	2002-11-12 12:09:00.000000	180	Interval	DIRECT	4	0	250
DEVSELF FILETEK.COM	2002-11-12 12:27:03.000000	68	Interval	DIRECT	4	1	250
DEVSELF FILETEK.COM	2002-11-12 12:30:01.000000	178	Interval	DIRECT	3	1	250
DEVSELF FILETEK.COM	2002-11-12 12:33:00.000000	179	Interval	DIRECT	3	2	250
DEVSELF FILETEK.COM	2002-11-12 12:45:00.000000	67	Interval	DIRECT	5	1	250
DEVSELF FILETEK.COM	2002-11-12 12:48:02.000000	182	Interval	DIRECT	3	1	250
DEVSELF FILETEK.COM	2002-11-12 12:51:04.000000	182	Interval	DIRECT	3	2	250
DEVSELF FILETEK.COM	2002-11-12 13:30:00.000000	86	Interval	DIRECT	0	3	250
DEVSELF FILETEK.COM	2002-11-12 13:42:00.000000	49	Interval	DIRECT	1	1	250
DEVSELF FILETEK.COM	2002-11-12 13:45:00.000000	180	Interval	DIRECT	1	2	250
DEVSELF FILETEK.COM	2002-11-12 14:12:00.000000	87	Interval	DIRECT	3	1	250
DEVSELF FILETEK.COM	2002-11-12 14:18:00.000000	118	Interval	DIRECT	4	1	250
DEVSELF FILETEK.COM	2002-11-12 14:21:03.000000	183	Interval	DIRECT	2	2	250
DEVSELF FILETEK.COM	2002-11-13 14:24:00.000000	177	Interval	DIRECT	1	1	250

100 rows returned, elapsed time: 00:00:01

Figure 6-7: Example of a result set containing statistics





### ▼ To submit an SQL statement with StorHouse/Admin ISQL

1. If necessary, log in to StorHouse/Admin.
2. In the folder list, expand the **Databases** folder.
3. Expand the statistics database.
4. On the **System** menu, click **ISQL**.
5. On the Interactive SQL working window, type the SQL statement in the **Enter ISQL statements** area.

Be sure to include the semi-colon at the end of the SQL statement.

6. Click .



## **Chapter 6 - Statistics**



# Operations

This chapter describes some of the daily or periodic procedures to perform when StorHouse/RFS is running. These procedures include:

- Working with log files
- Excluding file types from searches and collections
- Rereading the StorHouse/RFS configuration file
- Restarting StorHouse/RFS
- Validating the system
- Running StorHouse/RFS in simulation mode
- Reporting information about a user file



# Working with log files

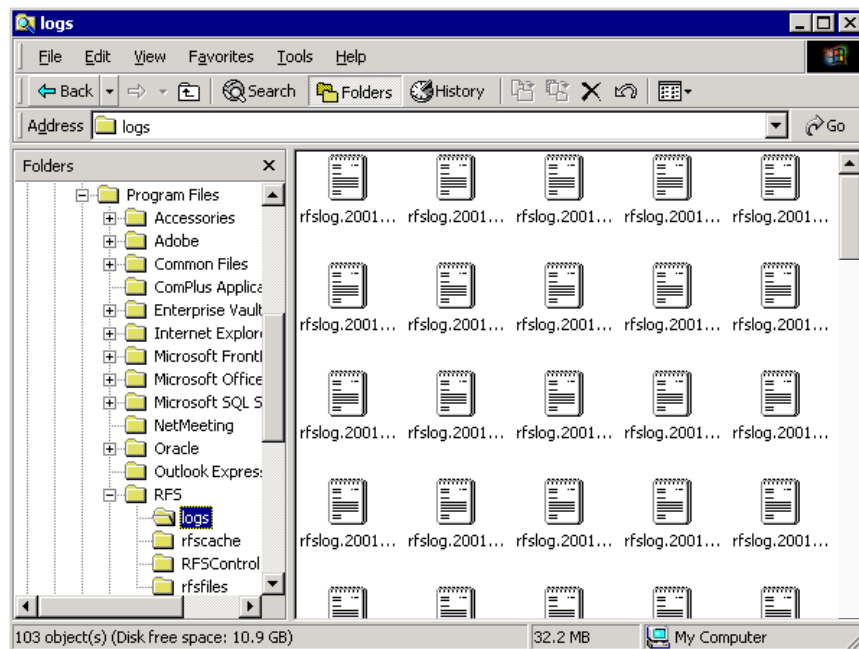
StorHouse/RFS creates entries in a log file when writing StorHouse collections, loading file locator data, and performing local and StorHouse searches. You can work with a log file like any text file.

After midnight, StorHouse/RFS saves the current log file and creates a new one. StorHouse/RFS also saves the current log file and creates a new one each time you restart the StorHouse/RFS service. StorHouse/RFS appends a date and time extension to the log file name. For example, in the file name `rfslog.20011205000002`, 20011205 is the date (YYYYMMDD) and 000002 is the time (HHMMSS). StorHouse/RFS does not purge log files.

The log files are located in the fully qualified path and file name specified on the `LogFile` parameter in the StorHouse/RFS configuration file.



Figure 7-1 illustrates an example of log files in a log file directory on a Windows-based StorHouse/RFS server platform.



**Figure 7-1: Example of log files in a log file directory**

▼ **To check the log**

1. Log in to the StorHouse/RFS server platform with the administrator account (in Windows) or as root (in UNIX).
2. Navigate to the directory specified at the LogFile parameter in the StorHouse/RFS configuration file.
3. Open a log file with a text editor.



# Excluding file types

The EXCLUSIONS section in the StorHouse/RFS configuration file contains file masks, such as AUX\*, MIDI\*, WAVE\*, to be excluded from searches and collections. For instance, StorHouse/RFS checks the exclusions list to determine whether a search request matches any of the masks provided. If it does and the file is not in the staging area, StorHouse/RFS reports FILE NOT FOUND. Using an exclusions list speeds processing for those applications that request support files (such as DLLs) when reading files. If you do not want to exclude any files from searches or collections, do not provide any file masks in the EXCLUSIONS section. If you do, simply add the file masks.

**Note:** Certain Windows applications, such as Media Player, will have extensive delays starting if you omit the following file masks from the exclusions list: MIXER\*, AUX\*, MIDI\*, WAVE\*, \*.DRV, \*.INI, \*.DLL, \*.EXE, \*.INF, THUMBS\*, and ~\*.TMP.



Figure 7-2 illustrates an example of the file masks displayed with the StorHouse/RFS configuration file utility.

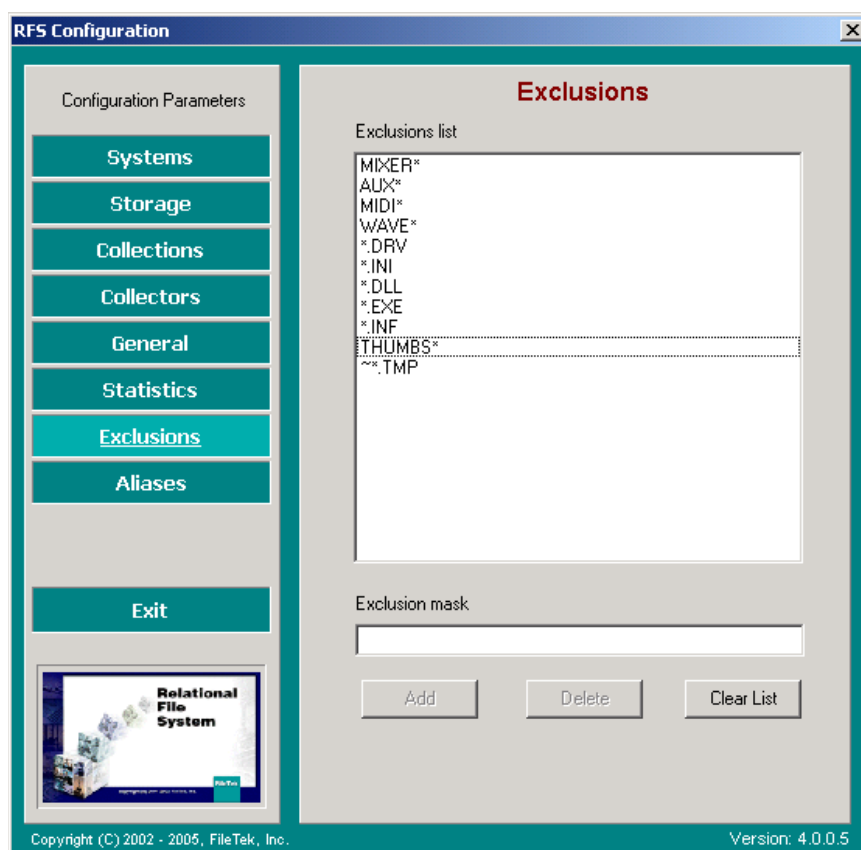


Figure 7-2: Example of file masks in the exclusions list



## Chapter 7 - Operations

These are the same file masks in the EXCLUSIONS section of the StorHouse/RFS configuration file.

```
[ EXCLUSIONS ]
MIXER*
AUX*
MIDI*
WAVE*
*.DRV
*.INI
*.DLL
*.EXE
*.INF
THUMBS*
~*.TMP
```

### ▼ To add or delete a file mask using the StorHouse/RFS configuration file utility

1. If necessary, start the StorHouse/RFS configuration file utility.
2. In the **RFS Configuration** dialog box, click **Exclusions**.
3. Add or delete file masks as follows:
  - To add a file mask, enter it the **Exclusions mask** box and click **Add**.
  - To delete a file mask, click the mask in the **Exclusions** list and then click **Delete**.
  - To delete all file masks in the **Exclusions** list, just click **Clear List**.
4. Click **Save**.





### ▼ To add or delete a file mask using a text editor in UNIX

1. If necessary, log in to the StorHouse/RFS server as root.
2. Open the `/rfs/files/rfs.cfg` file with the desired text editor. The following examples uses emacs.

```
emacs /rfs/files/rfs.cfg
```

3. Go to the [EXCLUSIONS] section.
4. To add a file mask, enter it on a separate line. To delete a file mask, remove the line with the editor.
5. Save the file.

## Rereading the StorHouse/RFS configuration file

Whenever you update dynamic parameters in the StorHouse/RFS configuration file, you may request StorHouse/RFS to reread the file. The changes take effect immediately after StorHouse/RFS rereads the file. The user ID you use requires read and write permission to the `rfs.cfg` file. StorHouse/RFS checks these permissions before rereading the file. StorHouse/RFS displays a Web page indicating successful processing; otherwise, StorHouse/RFS returns a file not found condition and the operating system displays the appropriate error message.

### ▼ To reread the StorHouse/RFS configuration file in Windows

1. Log in to the StorHouse/RFS server platform with a user login ID that has read and write permission to the `rfs.cfg` file.
2. On the **Start** menu, click **Run**.



3. In the **Run** dialog box, type the following information. (Replace V with the correct drive letter if your virtual file system is on a different drive.)

```
V:\SM_LOCAL\RFSCONFIG.HTML
```

4. Click **OK**.

### ▼ To reread the StorHouse/RFS configuration file in UNIX

1. Log in to the StorHouse/RFS server platform as root.
2. Type the following (uppercase required):

```
ls /RFS/SM_LOCAL/RFSCONFIG.HTML
```

## Restarting the StorHouse/RFS service

Whenever you update static parameters in the StorHouse/RFS configuration file, you must stop and then start the StorHouse/RFS service to implement the changes. Stopping StorHouse/RFS terminates any in-process local collections, StorHouse transfers, and StorHouse writes at the first convenient point. Starting StorHouse/RFS resumes accumulating files to the current local collection and starts StorHouse transfers and writes from the interruption point.

**Caution:** In a Windows environment, do not use the Restart and Resume options on the Server Properties dialog box. For StorHouse/RFS, use Stop and Start only.



### ▼ To restart the StorHouse/RFS service in Windows

1. Log in to the StorHouse/RFS server platform with the administrator account.
2. On the **Start** menu, click **Programs, Administrative Tools, Component Services**, and **Services**.
3. In the **Services** dialog box, locate and click **RFS 4.n Server**.
4. Right-click and then click **Properties**.
5. In the **Server Properties** dialog box:
  - a. Click **Stop**, and then click **OK**.
  - b. Click **Start**, and then **OK**.

### ▼ To restart the StorHouse/RFS service in UNIX

1. Log in to the StorHouse/RFS server platform as root.
2. Stop the StorHouse/RFS server by typing:
3. Start the StorHouse/RFS server by typing:

```
/etc/init.d/rfs stop
```

```
/etc/init.d/rfs start
```



# Validating the system

You can validate the system to verify configuration parameter settings, connections to StorHouse, and storage resource setup. StorHouse/RFS reports any suspicious parameter settings, invalid directory paths, and inaccessible resources. Validating the system is helpful during installation and any time you create a new set of storage resources for a collection set. StorHouse/RFS produces a Validation and Health Report that you can check to determine the result. StorHouse/RFS lists warnings or errors for any problems encountered.

Figure 7-3 illustrates a sample Validation and Health Report.

```
Validation and Health Report for RFS on DEV-ELF.FILETEK.COM
as-of: 2005-12-06 13:27:30
----- GENERAL INFORMATION -----
RFS Version: 4.0.0.8 Special Debug created: Dec 6 2005 at 12:04:06
FSD Version: RFSFSD 1.0.0.4 created: Oct 13 2005 at 14:11:39
The current log file is C:\rfslogs\rfslog.log
Data returned from StorHouse will be cached in C:\rfscache\
(CacheDir).
Up to 99 MB (MaxCacheSpace) may be used for StorHouse cache.
No more than 8 collection files on StorHouse (MaxSMFiles) may be
open concurrently.
Of these no more than 3 can be for writing collections concurrently
(MaxSMWriters).
Unused files will be removed from virtual directories after 5
minutes (FileCleanupTimeout).
General cleanup will occur every 3 minutes (CleanupTimeout).
----- STATISTICS -----
Statistics will be produced every 30 minutes (StatsInterval).
Statistics will be available as RFSSTATS.HTML.
Statistics will be available as RFSSTATS.TXT.
Successfully connected to primary statistics database RFSSTATS using
System Definition sthr and DSN alpha2
The SYSADM.STATS31_RFS table is valid and accessible.
The SYSADM.STATS31_TABLES table is valid and accessible.
The SYSADM.STATS31_SYSTEMS table is valid and accessible.
Completed STATS section validation.
----- ALIASES -----
Aliases will be read from table SYSADM.RFSALIAS_ALIASES in database
RFSALIASES
Successfully connected to primary system database RFSALIASES using
```



```
DSN alpha2
The SYSADM.RFSALIAS_ALIASES table is valid and accessible.
Completed ALIASES section validation.
----- COLLECTOR DEFINITION: BIGFILES -----
Files collected will be added to collections in collection
definition DIRECT.
Files will be collected from each user's \bigfile directory
(UserDir) in D:\LocalRWBig (StagingDir).
Files must be unmodified for 1 minutes before they will be
collected.
Up to 10000 MB of disk space can be used by this collector to stage
files.
Up to 10000 MB of disk space can be used by this collector to
collect files.
Subdirectories 2 levels or more below the virtual root directory
will be cleaned up when they are empty.
-- Warning: This collector has not yet collected any files and its
operability can not be verified.
Completed Collector Definition BIGFILES section validation.
----- COLLECTOR DEFINITION: DFILES -----
Files collected will be added to collections in collection
definition DIRECT.
Files will be collected from each user's \General directory
(UserDir) in d:\localrw (StagingDir).
Files must be unmodified for 1 minutes before they will be
collected.
Up to 450 MB of disk space can be used by this collector to stage
files.
Up to 400 MB of disk space can be used by this collector to collect
files.
Subdirectories 3 levels or more below the virtual root directory
will be cleaned up when they are empty.
-- Warning: This collector has not yet collected any files and its
operability can not be verified.
Completed Collector Definition DFILES section validation.
----- STORAGE DEFINITION: ST_RFS241LT -----
No more than 5 concurrent searches using ODBC (MaxSearchConnections)
will be allowed.
If ODBC connections are idle for 10 minutes, they will be
terminated.
Successfully connected to primary system database RFSRECOVER using
DSN alpha2
The SYSADM.RFS241LT_0 table is valid and accessible.
The SYSADM.RFS241LT_1 table is valid and accessible.
The SYSADM.RFS241LT_2 table is valid and accessible.
The SYSADM.RFS241LT_3 table is valid and accessible.
...
The SYSADM.RFS241LT_63 table is valid and accessible.
```



## Chapter 7 - Operations

```
The SYSADM.RFS241LT_S table is valid and accessible.
The SYSADM.RFS241LT_C table is valid and accessible.
Table set 0 contains 13201824 rows.
Table set 1 contains 10665851 rows.
SYSADM.RFS241LT contains 23867675 total rows.
Completed storage definition ST_RFS241LT section validation.
----- SYSTEM DEFINITION: sthr -----
SM connections will be considered down after 10 consecutive failed
attempts and will not be retried for 2 minutes.
ODBC connections will be considered down after 5 consecutive failed
attempts and will not be retried for 2 minutes.
Collections are written to the StorHouse performance buffer and
copied to media at the next backup (VTF=NEXT).
Collections written to StorHouse will have a checkpoint taken every
500MB (Checkpoint).
RFS will write collections to a single FSET.
Using default VSET RFSDEV
Using default FSET RFSDEV
Group RFSREC is valid and accessible.
VSET RFSDEV is valid and accessible.
VSET media type is TDA
FSET RFSDEV is valid and accessible.
Completed System Definition sthr section validation.
-----
***** There were 6 warnings reported *****
```

**Figure 7-3: Sample of a Validation and Health Report**

### ▼ To validate the system in Windows

1. Log in to the StorHouse/RFS server platform with a user login ID that has read and write permission to the rfs.cfg file.
2. On the **Start** menu, click **Run**.
3. In the **Run** dialog box, type the following. (Replace V with the correct drive letter if your virtual file system is on a different drive.)

Notepad V:\SM\_LOCAL\RFSHEALTH.TXT



4. Click OK.

StorHouse/RFS generates and displays the Validation and Health Report.

### ▼ To validate the system in UNIX

1. Log in to the StorHouse/RFS server platform as root.
2. Type the following (uppercase required):

```
cat /RFS/SM_LOCAL/RFSHEALTH.TXT
```

StorHouse/RFS displays the Validation and Health Report.

## Running StorHouse/RFS in simulation mode

You can run StorHouse/RFS in simulation mode, which is helpful to test the collection and storage process. In simulation mode, StorHouse/RFS runs without a StorHouse system. All searches that are not satisfied locally result in File Not Found and all writes and loads are reported as successful.

**Warning:** After running StorHouse/RFS in simulation mode, be sure to turn the simulation off. Otherwise, you could lose data if you archive production data while the simulation is running.

### ▼ To run StorHouse/RFS in simulation mode in Windows

1. Log in to the StorHouse/RFS server platform with a user login ID that has read and write permission to the rfs.cfg file.
2. Open the rfs.cfg file with the desired text editor.



## Chapter 7 - Operations

3. Go to the [RFS] section.
4. Change the Simulate parameter to Yes.
5. Save the rfs.cfg file and minimize it.
6. Display the log to monitor the activity. See “Working with log files” on page 7-2 for instructions.
7. When you are done working in simulation mode:
  - a. Maximize, or open, the rfs.cfg file
  - b. Change the Simulate parameter to No to stop the simulation.
  - c. Save the file.

### ▼ To run StorHouse/RFS in simulation mode in UNIX

1. If necessary, log in to the StorHouse/RFS server as root.
2. Open the `/rfs/files/rfs.cfg` file with the desired text editor. The following examples uses emacs.

```
emacs /rfs/files/rfs.cfg
```
3. Go to the [RFS] section.
4. Change the Simulate parameter to yes.
5. Save the file.
6. Display the log to monitor the activity. See “Working with log files” on page 7-2 for instructions.





7. When you are done working in simulation mode:
  - a. Open the rfs.cfg file.
  - b. Change the Simulate parameter to no to stop the simulation.
  - c. Save the file.

## **Reporting information for a user file**

StorHouse/RFS provides a command that uses the SM\_LOCAL keyword and an encoded file name to report metadata only or metadata and media location for a user file. The command can be a parameter to a utility such as Notepad, Wordpad, or cat, or it can also be run programmatically by using the encoded file name with the fopen( ) and fread( ) functions.

The user file reporting command syntax for Windows is:

```
"virtual_drive_letter:\SM_LOCAL\command_verb^complete_file_path"
```

The user file reporting command syntax for UNIX is:

```
"mount_point/SM_LOCAL/command_verb^complete_file_path"
```

The entire command syntax must be enclosed by double quotation marks.



## Chapter 7 - Operations

**Table 7-1: Command syntax for the user file reporting command**

Command components	Description
For Windows: virtual_drive_letter:	The letter that represents the StorHouse/RFS virtual drive. Example: v:
For UNIX: mount_point:	The name of the StorHouse/RFS mount point. Example: RFS:
SM_LOCAL	Required keyword.  SM_LOCAL must be preceded and followed by a back slash (\) in Windows or a forward slash (/) in UNIX.
command_verb	One of the following: <ul style="list-style-type: none"><li>■ XMLFIND - generates XML output that includes metadata and media location for the file.</li><li>■ XMLFINDB - generates XML output that includes only metadata for the file.</li><li>■ FIND - generates text output that includes metadata and media location for the file.</li><li>■ FINDB - generates text output that includes only metadata for the file.</li></ul>
complete_file_path	The complete path to the file name for which information will be reported.  You must use the ^ symbol for all path delimiters positioned after the command_name.

Figure 7-4 illustrates the XML output (metadata and media location) generated by the following command:

```
"w:\SM_LOCAL\XMLFIND^GENERAL^dates^RFSFILE_00000010.TXT"
```

```
FILE>\general\dates\rfsfile_00000010.txt
<SYSTEM>
<TYPE>PRIMARY</TYPE>
<DNS>alpha2</DNS>
<INSTANCE>
<STATUS>ACTIVE</STATUS>
```



```
<MODIFIED>2005-02-04 10:27:54</MODIFIED>
<SIZE>100000000</SIZE>
<OWNER>ELF.FILETEK</OWNER>
<LBN>59</LBN>
<COLLECTION>DIRECT20050203170548(A).DEV-ELF.FILETEK.COM</
COLLECTION>
<VOLUME>TDA"010161":A</VOLUME>
<LIB>L00</LIB>
<MEDIA>WRITTEN</MEDIA>
</INSTANCE>
<INSTANCE>
<STATUS>ACTIVE</STATUS>
<MODIFIED>2005-02-04 11:29:13</MODIFIED>
<SIZE>100000000</SIZE>
<OWNER>ELF.FILETEK</OWNER>
<LBN>59</LBN>
<COLLECTION>DIRECT20050203170548(A).DEV-ELF.FILETEK.COM</
COLLECTION>
<VOLUME>TDA"010161":A</VOLUME>
<LIB>L00</LIB>
<MEDIA>WRITTEN</MEDIA>
</INSTANCE>
<INSTANCE>
<STATUS>ACTIVE</STATUS>
<MODIFIED>2005-02-04 11:10:55</MODIFIED>
<SIZE>100000000</SIZE>
<OWNER>ELF.FILETEK</OWNER>
<LBN>59</LBN>
<COLLECTION>DIRECT20050203170548(A).DEV-ELF.FILETEK.COM</
COLLECTION>
<VOLUME>TDA"010161":A</VOLUME>
<LIB>L00</LIB>
<MEDIA>WRITTEN</MEDIA>
</INSTANCE>
<INSTANCE>
<STATUS>ACTIVE</STATUS>
<MODIFIED>2005-02-04 10:54:14</MODIFIED>
<SIZE>100000000</SIZE>
<OWNER>ELF.FILETEK</OWNER>
<LBN>59</LBN>
<COLLECTION>DIRECT20050203170548(A).DEV-ELF.FILETEK.COM</
COLLECTION>
<VOLUME>TDA"010161":A</VOLUME>
<LIB>L00</LIB>
```



## Chapter 7 - Operations

```
<MEDIA>WRITTEN</MEDIA>
</INSTANCE>
</SYSTEM>
</FILE>
```

**Figure 7-4: Sample user file report in XML format**

Table 7-2 describes the XML tag definitions in the sample report.

**Table 7-2: Tag definitions for the user file report**

Tag	Description
FILE>	The name of the file that the report information describes.
<SYSTEM>	The beginning of system information for all reported instances of the file.
<TYPE>	The type of system where the file resides. Values are: <ul style="list-style-type: none"><li>■ LOCAL - the file resides in a local collection and has not been written to StorHouse. If &lt;TYPE&gt; is LOCAL, the &lt;LIB&gt;, &lt;MEDIA&gt;, and &lt;VOLUME&gt; tags will not appear in the report, because they only pertain to StorHouse files.</li><li>■ PRIMARY - the file resides on the primary StorHouse system.</li><li>■ MIRROR - the file resides on the duplex StorHouse system.</li></ul>
<DNS>	The DNS name of the system where the file resides.
<INSTANCE>	The beginning of information for a particular file instance. Each instance represents one occurrence of a user file in a row in the StorHouse database.
STATUS	The status of the file instance. Values are: <ul style="list-style-type: none"><li>■ ACTIVE - the file is accessible to users.</li><li>■ DELETED - StorHouse/RFS has marked the file as deleted.</li><li>■ UNCOLLECTED - the file still resides in the staging area. It has not been renamed to a local collection.</li><li>■ UNWRITTEN - the file has not been written to a StorHouse collection.</li><li>■ UNWRITTEN DELETE - StorHouse/ RFS marked the file as deleted but has not updated the metadata on StorHouse.</li></ul>

**Table 7-2: Tag definitions for the user file report (continued)**

Tag	Description
MODIFIED	The date when the file or the file metadata was last modified.
SIZE	The size of the file in bytes.
OWNER	The file owner.
LBN	The unique location of the file in a collection. LBN stands for logical block number.
COLLECTION	The name of the StorHouse collection that contains the file.
VOLUME	The volume identification code (VID) of the StorHouse volume where the file resides.
LIB	The device identification code (DID) of the StorHouse device where the volume containing the file resides.
MEDIA	The status of the file on StorHouse. Values are: <ul style="list-style-type: none"><li>■ WRITTEN - the file has been written to its resident file set.</li><li>■ NOT WRITTEN - the file is still in the performance buffer.</li></ul>

For example, if a file is in the staging area, the `TYPE` will be `LOCAL`, the `STATUS` will be `UNCOLLECTED`, and the `DNS` will be the DNS of the local computer. If the file is collected, the `TYPE` will be `LOCAL`, and the `STATUS` will be either `UNWRITTEN` or `UNWRITTEN DELETE`. The `VOLUME`, `LIB`, and `MEDIA` tags will not be reported because the file does not reside on StorHouse.



## Chapter 7 - Operations

### ▼ To generate metadata only or metadata and media location for a user file from a Windows StorHouse/RFS client

1. On the Start menu, click **Run**.
2. In the **Run** dialog box, type the utility that will open the generated report (for example, Notepad) followed by the file locator command.

For example:

```
Notepad "V:\SM_LOCAL\FIND^GENERAL^DATES^RFSFILE_00000010.TXT"
```

In this example, V is the virtual drive.

3. Click **OK**.

StorHouse/RFS generates and displays metadata and media information for RFSFILE\_000000010.TXT in text format.

### ▼ To generate metadata only or metadata and media location for a user file from a UNIX StorHouse/RFS client

1. Log into the UNIX StorHouse/RFS client with an account that has access to the directory containing StorHouse/RFS files.
2. Type the utility that will open the generated report (for example, cat) followed by the file locator command.

For example:

```
cat "/RFS/SM_LOCAL/XMLFIND^ACCT^MAY^RFSFILE_00000010.TXT"
```

In this example, RFS is the mount point. StorHouse/RFS generates metadata and media location information about RFSFILE\_000000010.TXT in XML format.



## **The StorHouse/RFS configuration file**

Each StorHouse/RFS server has one StorHouse/RFS configuration file that provides the operating parameters for the server. The file, named `rfs.cfg`, must reside in the same directory as the `rfs.exe` (for Windows) or in the root `/etc` directory (for UNIX). This appendix describes:

- Ways to update the file
- The format of the file
- The parameters and values



# File maintenance

A sample StorHouse/RFS configuration file with default values is provided with each StorHouse/RFS installation. If you're running StorHouse/RFS in a Windows environment, you can use the StorHouse/RFS configuration file utility supplied on the StorHouse/RFS installation CD to maintain file parameters through a graphical user interface. If you're running StorHouse/RFS in a UNIX environment, you can update the file with any text editor, such as emacs or vi.

# File format

The StorHouse/RFS configuration file consists of *sections* and *definitions*. Each configuration file contains a set number of sections and any number of definitions. Table A-1 describes the sections.

**Table A-1: Sections in the StorHouse/RFS configuration file**

Section	Description
RFS	Specifies general operating parameters and defaults
COLLECTORS	Specifies collector names and identifies corresponding collection definitions
STATS	Specifies parameters used to generate and store statistics
EXCLUSIONS	Identifies file masks to exclude from searches and collections
ALIASES	Specifies resources used to manage file owner and group name changes





Table A-2 describes the definitions.

**Table A-2: Definitions in the StorHouse/RFS configuration file**

Definition	Description
System	Defines where collections, statistics, and aliases are stored on StorHouse
Storage	Specifies where file locator data and collection metadata are stored on StorHouse and identifies the corresponding system definition
Collection	Defines specifications for each collection set and identifies the corresponding storage definition
Collector	Specifies directories where each collector looks for data and security requirements for files in those directories and identifies the corresponding collection definition

The sections and definitions may be located anywhere, in any order, in the StorHouse/RFS configuration file. Note that if you use the StorHouse/RFS configuration file utility to add definitions, the utility places new definitions at the end of the file.

If you are using a text editor to maintain the StorHouse/RFS configuration file, the following information describes the file format and conventions. If you are using the StorHouse/RFS configuration file utility, some of this information does not apply because the utility provides a graphical interface for maintaining parameters.

## Section names

In the StorHouse/RFS configuration file, a section begins with a section name delimited by brackets. Similar section names are displayed by the



## Appendix A - The StorHouse/RFS configuration file

StorHouse/RFS configuration file utility. Table A-3 lists the section names.

**Table A-3: Section names**

Configuration file	Configuration file utility
[RFS]	General
[COLLECTORS]	Collectors
[STATS]	Statistics
[EXCLUSIONS]	Exclusions
[ALIASES]	Aliases

### Definition names

Each definition has a name, which must be unique within a file. See “Naming StorHouse/RFS resources” on page 1-23 for conventions on naming definitions with the StorHouse/RFS configuration file utility. In a StorHouse/RFS configuration file, definition names must be delimited by brackets, for example:

- [ JOURNAL ]
- [ MAILCOLLECTOR ]

### Parameters

The StorHouse/RFS configuration file utility displays each parameter name. You simply provide or omit values. In a StorHouse/RFS configuration file, each section or definition contains a series of parameters expressed as *keyword-value pairs*, in any order, according to the following rules:

- An equal sign separates each keyword and value.
- No spaces are allowed before or after the equal sign.



- Each keyword-value pair appears on a separate line in the file.

In the following example, `FileCleanupTimeout` is the keyword and 60 is the value:

```
FileCleanupTimeout=60
```

### Required and optional parameters

Parameters may be required or optional. A required parameter must be present with a value. An optional parameter may be omitted from the file, or the keyword may be present without a value.

### Omitted parameters or values

An omitted parameter or a parameter without a value takes the default value. For example, no statistics are collected because the value for the following parameter is missing:

```
StatsInterval=
```

### Ignored parameters

You can place a special character—such as a semi-colon—before a keyword to indicate that StorHouse/RFS should ignore any value. This has the same effect as omitting the value. For instance, StorHouse/RFS ignores the following parameter because it is preceded by a semi-colon:

```
;StatsInterval=60
```

### Dynamic and static parameters

Parameters are dynamic or static. When you change a *dynamic parameter*, the value takes effect immediately after you request a reread of the StorHouse/RFS configuration file. When you change a *static parameter*,



## Appendix A - The StorHouse/RFS configuration file

the value takes effect after you restart the StorHouse/RFS service. Table A-4 identifies the dynamic and static parameters. Note that the parameters in the STATS section may be dynamic and static. See “Setting statistics parameters” on page 6-13 for more information about when those values take effect.

### Summary of parameters

Table A-4 lists the parameters in alphabetical order and identifies the section and whether each parameter is required or optional and dynamic or static.

**Table A-4: Alphabetical list of parameters**

Parameter	Location	Required	Dynamic
Browse	Collection definition	No	Yes
CacheDir	RFS section	Yes	No
CheckPoint	System definition	No	Yes
CleanupTimeout	RFS section	No	Yes
CollectionDir	Collection definition	Yes	No
Compression	Collection definition	No	No
Database	ALIASES section	Yes if section present	No
Database	Storage definition	Yes	No
Database	STATS section	No, unless desired	Yes and No
DBDriver	System definition	No	No
DBHost	System definition	No	No
DBPort	System definition	No	No
DNSName	System definition	Yes	No
FileCleanupTimeout	RFS section	No	Yes



Table A-4: Alphabetical list of parameters (continued)

Parameter	Location	Required	Dynamic
FileType	STATS section	Yes if section present	Yes and No
Flush	RFS section	No	No
FSET	System definition	No	No
FSETSegments	System definition	No	No
Group	System definition	No	No
Group	Collector definition	No	No
KeepSubdirectories	Collector definition	No	Yes
LocalPath	RFS section	Yes	No
LogFile	RFS section	Yes	No
MailRecipient	System definition	No	No
MailServer	RFS section	No	No
MaxCacheSpace	RFS section	Yes	Yes
MaxCollectionSpace	Collector definition	Yes	Yes
MaxLoadInterval	Collection definition	No	Yes
MaxSearchConnections	Storage definition	No	No
MaxSMFiles	RFS section	No	Yes
MaxSMWriters	RFS section	No	Yes
MaxStagingSpace	Collector definition	Yes	Yes
MaxWriteSize	Collection definition	No	Yes
MirrorName	ALIASES section	No	No
MirrorName	Storage definition	No	No
MirrorName	STATS section	No	Yes and No
Password	ALIASES section	Yes if section present	No



## Appendix A - The StorHouse/RFS configuration file

**Table A-4: Alphabetical list of parameters (continued)**

Parameter	Location	Required	Dynamic
Password	System definition	Yes	No
Password	Storage definition	Yes	No
Password	STATS section	No	Yes and No
PasswordType	ALIASES section	No	Not applicable
PasswordType	System definition	No	Not applicable
PasswordType	Storage definition	No	Not applicable
PasswordType	STATS section	No	Not applicable
Permissions	Collector definition	No	No
Retention	Collector definition	No	No
RetryInterval	System definition	No	Yes
SafetyPath	RFS section	No	No
SearchConnectionTimeout	Storage definition	No	Yes
Simulate	RFS section	No	No
StagingDir	Collector definition	Yes	No
StatsInterval	STATS section	No	Yes
STHName	System definition	Yes	No
Storage	Collection definition	Yes	No
SystemName	ALIASES section	Yes if section present	No
SystemName	Storage definition	Yes	No
SystemName	STATS section	No	Yes and No
TableName	ALIASES section	Yes if section present	No



Table A-4: Alphabetical list of parameters (continued)

Parameter	Location	Required	Dynamic
TableName	Storage definition	Yes	No
TableName	STATS section	No	Yes and No
TimeoutOverride	System definition	No	Yes
UserDir	Collector definition	No	No
UserId	ALIASES section	Yes if section present	No
UserId	System definition	Yes	No
UserId	Storage definition	Yes	No
UserId	STATS section	No	Yes and No
Version	RFS section	Yes	No
VSET	System definition	No	No
VTF	System definition	No	No
WaitTime	Collector definition	Yes	Yes

## File example

The following example illustrates a text version of the StorHouse/RFS configuration file.

```

Section — [RFS]
        LocalPath=C:\rfssearch
Parameter — CacheDir=C:\rfscache
and value   LogFile=C:\rfslogs\rfslog.log
        Debug=No
        MaxCacheSpace=50
        MailServer=mailserv.filetek.com
        FileCleanupTimeout=5
        MaxSMFiles=8
        MaxSMWriters=3

```



## Appendix A - The StorHouse/RFS configuration file

```
CleanupTimeout=3
SafetyPath=D:\Safety
Version=4.0
```

### Section — [ EXCLUSIONS ]

```
MIXER*
AUX*
MIDI*
WAVE*
*.DRV
*.INI
*.DLL
*.INF
THUMBS*
~*.TMP
~$*.DOC
```

### Section — [ STATS ]

```
StatsInterval=30
FileType=TXT HTML
SystemName=sthr
Database=RFSSTATS
TableName=SYSADM.STATS31
UserId=SYSADM
Password=_)=>01
PasswordType=0
```

### Section — [ ALIASES ]

```
SystemName=sthr ——— System definition name
Database=RFSALIASES
TableName=SYSADM.RFSALIAS
UserId=SYSADM
Password=_)=>01
PasswordType=0
```

### Section — [ COLLECTORS ]

```
DFILES=DIRECT
BIGFILES=DIRECT
RFS31=DIRECT31 ——— Collection definition name
|
| Collector definition name
```





## System definition — [ sthr ]

```
STHName=alpha2
DNSName=alpha2
RetryInterval=2
UserId=LVR0M
Password=X<LX
PasswordType=0
Group=RFSREC
Checkpoint=500
MailRecipient=elm@filetek.com
DBDriver=StorHouse
TimeoutOverride=2
```

## Collector definition — [ BIGFILES ]

```
StagingDir=d:\localrwBig
WaitTime=1
UserDir=\bigfile
MaxStagingSpace=10000
MaxCollectionSpace=10000
Permissions=R
KeepSubdirectories=0
```

## Collector definition — [ DFILES ]

```
StagingDir=d:\localrw
UserDir=\General
WaitTime=1
MaxStagingSpace=450
MaxCollectionSpace=400
Group=TECHDS.FILETEK
Permissions=R
KeepSubdirectories=0
```

## Collector definition — [ RFS31 ]

```
StagingDir=d:\localrw
UserDir=\RFS31
WaitTime=1
MaxStagingSpace=6000
Group=TECHDS.FILETEK
Permissions=R
MaxCollectionSpace=600
KeepSubdirectories=0
```



## Appendix A - The StorHouse/RFS configuration file

Collection definition — [ DIRECT ]  
Storage=STORAGE\_RFS241LT ——— Storage definition name  
CollectionDir=C:\RFSControl  
MaxLoadInterval=5  
MaxWriteSize=200  
Browse=Yes  
Compression=No

Collection definition — [ DIRECT31 ]  
Storage=STORAGE\_RFS31  
CollectionDir=D:\RFSControl  
MaxLoadInterval=5  
MaxWriteSize=500  
Browse=Yes

Storage definition — [ STORAGE\_RFS241LT ]  
Database=RFSRECOVER  
TableName=SYSADM.RFS241LT  
UserId=SYSADM  
Password=\_)=>01  
PasswordType=0  
SystemName=sthr  
MaxSearchConnections=5  
SearchConnectionTimeout=10

Storage definition — [ STORAGE\_RFS31 ]  
Database=RFSRECOVER  
TableName=SYSADM.RFS31  
UserId=SYSADM  
Password=\_)=>01  
PasswordType=0  
SystemName=sthr  
;MirrorName=sthr ——— Ignored parameter  
MaxSearchConnections=5  
SearchConnectionTimeout=10



## RFS section

The RFS, or General, section contains system-wide values and defaults. The parameters are as follows.

### LocalPath

Format	LocalPath=<fully qualified path>
Examples	LocalPath=C:\rfsfiles LocalPath=/rfs/files
Required	Yes
Dynamic	No
Description	Fully qualified path to contain the following files: <ul style="list-style-type: none"> <li>■ Local statistics files in HTML, XML, text, or any combination of formats</li> <li>■ RFSCONFIG.HTML (for re-reading the StorHouse/RFS configuration file)</li> <li>■ RFSALIAS.HTML (for managing aliases)</li> </ul>

### CacheDir

Format	CacheDir=<fully qualified path>
Examples	CacheDir=C:\Program Files\RFS\rfscache CacheDir=/rfs/cache
Required	Yes
Dynamic	No
Description	Fully qualified path where StorHouse/RFS caches data that it reads from StorHouse. This directory can be on the StorHouse/RFS server or on a device accessible to the server. In UNIX, the root account owns the cache directory. The ID under which StorHouse/RFS runs requires access to this cache directory.



## Appendix A - The StorHouse/RFS configuration file

### LogFile

Format	LogFile=<fully qualified file name>
Examples	LogFile=C:\Program Files\RFS\logs\rfslog.log LogFile=/rfs/logs/rfslog.log
Required	Yes
Dynamic	No
Description	Fully qualified path and file name of the StorHouse/RFS log file. StorHouse/RFS creates log entries when accumulating files to local collections, writing StorHouse collections, and performing local searches and StorHouse searches.

### MailServer

Format	MailServer=<any string>
Example	MailServer=mailserve.gwi.com
Required	No
Dynamic	No
Description	Name of the mail server to use for sending e-mail to the address specified at the MailRecipient parameter in a system definition. This parameter does not apply to UNIX environments. If you omit this value and/or the MailRecipient value, no e-mail is sent.



## MaxCacheSpace

Format	MaxCacheSpace=<MB>
Example	MaxCacheSpace=500
Required	Yes
Dynamic	No
Description	Maximum amount of storage (in MB) to use for caching files retrieved from StorHouse. Cache uses a least recently used/most recently used scheme to manage data. When space is needed to cache a record read from StorHouse, StorHouse/RFS replaces the least recently used record with the new one. The minimum value is 100 MB. The maximum value is 1000 MB. If you specify 0 or a value under 100 MB, StorHouse/RFS reserves 100 MB.

## FileCleanupTimeout

Format	FileCleanupTimeout=<minutes>
Example	FileCleanupTimeout=90
Default	FileCleanupTimeout=60
Required	No
Dynamic	Yes
Description	Frequency that StorHouse/RFS checks the virtual file system for unused files to remove. A file is unused when it hasn't been touched (for example, opened) since the last file cleanup. Once removed, the file no longer appears in the virtual file system but it is still available for access and automatically added back to the virtual file system when requested. The minimum value (and default) is 60.



## Appendix A - The StorHouse/RFS configuration file

### MaxSMFiles

Format	MaxSMFiles=<number>
Example	MaxSMFiles=25
Default	MaxSMFiles=32
Required	No
Dynamic	Yes
Description	Maximum number of StorHouse connections that can be opened at the same time for reading files. One StorHouse connection is used for reading any number of user files in a StorHouse collection, that is, one connection per StorHouse collection. When the number of connections is exceeded and a StorHouse connection is needed, StorHouse/RFS closes the least recently used connection even if the CleanupTimeout has not expired. In a UNIX environment, this value should not be higher than 32. In a multiple StorHouse/RFS server environment, the value of MaxSMFiles should be less than or equal to the StorHouse VRAM_NUM_KU system parameter.

### MaxSMWriters

Format	MaxSMWriters=<number>
Example	MaxSMWriters=15
Default	MaxSMWriters= which means use one half the value of the MaxSMFiles parameter
Required	No
Dynamic	Yes
Description	Maximum number of StorHouse connections that may be used for writing collections to StorHouse. This parameter helps prevent a busy system from using all connections for writing data when some are needed for retrieving data. This value cannot exceed the MaxSMFiles value. The default value is one half of the MaxSMFiles parameter. For example, if MaxSMFiles=20, then the default value is MaxSMWriters=10.



## Flush

Format	Flush=<WRITE CLOSE NO or SYSTEM>
Example	Flush=WRITE
Default	Flush=NO or Flush=SYSTEM
Required	No
Dynamic	No
Description	<p>Option that determines when StorHouse/RFS sends the flush buffers command to the operating system for files written to the staging area. This command causes the operating system's file cache buffers to be flushed to disk.</p> <ul style="list-style-type: none"><li>■ WRITE flushes the buffers after each write request. This setting is the safest (improves the chances of recovering a file in the event of a failure) but the slowest.</li><li>■ CLOSE flushes the buffers when a file is closed. This setting is faster but provides a window where data in the system cache could be lost should the StorHouse/RFS server crash.</li><li>■ NO or SYSTEM flushes the buffers by the operating system as appropriate for current system operating conditions. This setting provides the fastest performance but the longest window of vulnerability.</li></ul> <p>With all settings, once StorHouse/RFS collects a file, it is flushed to the collection no later than two minutes after it is collected. StorHouse/RFS removes the file from the staging area only after the file is safely flushed with the collection.</p>



## Appendix A - The StorHouse/RFS configuration file

### CleanupTimeout

Format	CleanupTimeout=<minutes>
Example	CleanupTimeout=4
Default	CleanupTimeout=10
Required	No
Dynamic	Yes
Description	Frequency that StorHouse/RFS waits before closing StorHouse connections and purging cache files that are no longer in use.

### Simulate

Format	Simulate=<YES NO>
Example	Simulate=YES
Default	Simulate=NO
Required	No
Dynamic	No
Description	<p>Option to run StorHouse/RFS in simulation mode for development or evaluation use only.</p> <p><b>Warning:</b> Do not use Simulate=YES in a production environment. Such use will result in the loss of data.</p> <p>In simulation mode, StorHouse/RFS runs without a StorHouse system. All StorHouse/RM searches result in File Not Found and all writes and loads are reported as successful. See “Running StorHouse/RFS in simulation mode” on page 7-13 for more information about the Simulate parameter.</p>





## SafetyPath

Format	SafetyPath=<fully qualified path>
Examples	SafetyPath=C:\rfs\safety_files SafetyPath=/rfs/files/safety_files
Required	No
Dynamic	No
Description	Fully qualified path to contain a secondary copy of metadata for user file changes (such as renames, deletes, security updates) that are not associated with the current local collection. StorHouse/RFS uses these safety copies in the event it detects a corrupt load (.ldr) file and has to rebuild it. StorHouse/RFS removes the entries from the safety path directory after successfully loading the entries into a StorHouse table. StorHouse/RFS fully manages the content of this directory.

## Version

Format	Version=<StorHouse/RFS version number>
Example	Version=4.0
Required	Yes
Dynamic	No
Description	Software version number of the StorHouse/RFS system. For a Windows installation, you cannot update this parameter with the StorHouse/RFS configuration file utility. You must update it with a text editor.



# STATS section

The STATS section contains parameters that StorHouse/RFS uses to create and store statistics in a local statistics file, in a StorHouse database, or both.

## StatsInterval

Format	<code>StatsInterval=&lt;minutes&gt;</code>
Example	<code>StatsInterval=60</code>
Default	<code>StatsInterval=10</code>
Required	No, unless you want to capture statistics in a local statistics file or in a StorHouse database
Dynamic	Yes
Description	Frequency that StorHouse/RFS captures and writes statistics. If omitted or set to 0, no statistics are written. This interval is relative to the clock. For example, if you set the interval to 10 minutes and you start at 2:11, then StorHouse/RFS writes the next statistics at 2:20. If you start at 2:05, StorHouse/RFS writes the next statistics at 2:10.



## FileType

Format	FileType=<HTML   XML   TXT>
Examples	FileType=XML FileType=HTML TXT
Default	None
Required	Yes if section present
Dynamic	Yes and No
Description	Option to store statistics in a local statistics file in HTML, XML, text, or any combination of formats. Use a space to separate multiple formats, as shown in the example. StorHouse/RFS stores the local statistics file in the location specified by the LocalPath parameter. If omitted, no local statistics file is created.

## Database

Format	Database=<database name on StorHouse>
Example	Database=STATSDATABASE
Required	No, unless you want to store statistics in a StorHouse database
Dynamic	Yes and No
Description	Name of the StorHouse database to store statistics. You create this database with StorHouse/Admin. The database name is case sensitive.



## Appendix A - The StorHouse/RFS configuration file

### TableName

Format	TableName=<owner.base_table_name>
Example	TableName=SYSADM.STATS
Required	No, unless you want to store statistics in a StorHouse database
Dynamic	Yes and No
Description	<p>Owner and base name of the StorHouse tables used to store statistics. StorHouse/RFS adds a suffix (RFS, TABLES, and SYSTEMS) to the table name. For example, if the owner is SYSADM and the base table name is STATS, then the complete table names are:</p> <ul style="list-style-type: none"><li>■ SYSADM.STATS_RFS</li><li>■ SYSADM.STATS_TABLES</li><li>■ SYSADM.STATS_SYSTEMS</li></ul> <p>You use the tblgen utility to create the statistics tables. The TableName in the StorHouse/RFS configuration file must match the table name provided when running the tblgen utility. See "Creating statistics tables" on page 6-11 for more information about using the tblgen utility.</p>

### UserId

Format	UserId=<StorHouse account>
Example	UserId=SYSADM
Required	No, unless you want to store statistics in a StorHouse database
Dynamic	Yes and No
Description	<p>StorHouse account ID used to log in to the StorHouse system to load the statistics tables. This account ID must be the owner of the statistics tables or have the following minimum privileges: SQLEXECUTE StorHouse privilege and INSERT database component privilege on the statistics tables.</p>



## Password

Format	Password=<StorHouse password>
Example	Password=* )&^*!
Required	No, unless you want to store statistics in a StorHouse database
Dynamic	Yes and No
Description	StorHouse account password (maximum 32 characters) associated with the UserId that is used to load the statistics tables. The StorHouse/RFS configuration file utility encrypts the display of the password. In a UNIX environment, when you specify or change this password, you must run the gen_cfg program. See page 4-11 for instructions.

## PasswordType

The PasswordType parameter is used internally by StorHouse/RFS. Do not change this parameter or your system may not work properly.

## SystemName

Format	SystemName=<system definition name>
Example	SystemName=ComplianceSet
Required	No, unless you want to store statistics in a StorHouse database
Dynamic	Yes and No
Description	Name of the system definition that identifies the primary StorHouse system for storing statistics.



## Appendix A - The StorHouse/RFS configuration file

### MirrorName

Format	MirrorName=<system definition name>
Example	MirrorName=ComplianceSet2
Required	No, unless using a secondary StorHouse system for storing statistics
Dynamic	Yes and No
Description	Name of the system definition that identifies the secondary StorHouse system for storing statistics. StorHouse/RFS writes statistics to this StorHouse system after writing the data to the primary StorHouse system.

## EXCLUSIONS section

The EXCLUSIONS section lists file masks to be excluded from searches and collections. There are no parameters in this section, only values, for example AUX\*, MIDI\*, WAVE\*, \*.DRV, \*.INI, \*.DLL, \*.EXE, \*.INF, THUMBS\*, and ~\*.TMP and -\*.DOC. See “Excluding file types” on page 7-4 for more information.

## ALIASES section

The ALIASES section identifies resources used for alias checking. If you omit this section, no alias checking occurs. See “Managing aliases for user and group names” on page 4-3 for more information about aliases.



**Database**

Format	Database=<database name on StorHouse>
Example	Database=ALIASES
Required	No, unless you want to use alias checking
Dynamic	No
Description	Name of the StorHouse database to store aliases. You create this database with StorHouse/Admin. The database name is case sensitive.

**TableName**

Format	TableName=<owner.base_table_name>
Example	TableName=SYSADM.GROUP
Required	No, unless you want to use alias checking
Dynamic	No
Description	Owner and base table name of the StorHouse table to store aliases. StorHouse/RFS adds the ALIASES suffix to the table name. For example, if the owner is SYSADM and the base table name is GROUP, then the complete table name is SYSADM.GROUP_ALIASES. You create this table with the tblgen utility. The TableName value in the StorHouse/RFS configuration file must match the table name supplied when running the tblgen utility. See “Creating an aliases table” on page 4-4 for more information about using the tblgen utility.



## Appendix A - The StorHouse/RFS configuration file

### UserId

Format	UserId=<StorHouse account>
Example	UserId=SYSADM
Required	No, unless you want to use alias checking
Dynamic	No
Description	StorHouse account ID used to log in to the StorHouse system to check the aliases table. This account ID must be the owner of the aliases table specified or have the following minimum privileges: SQLEXECUTE StorHouse privilege and SELECT, INSERT, and UPDATE database component privileges on the aliases table.

### Password

Format	Password=<StorHouse password>
Example	Password=* )&^*!
Required	No, unless you want to use alias checking
Dynamic	No
Description	StorHouse account password (maximum 32 characters) associated with the UserId that is used to check the aliases table. The StorHouse/RFS configuration file utility encrypts the display of the password. In a UNIX environment, when you specify or change this password, you must run the gen_cfg program. See page 4-11 for instructions.

### PasswordType

The PasswordType parameter is used internally by StorHouse/RFS. Do not change this parameter or your system may not work properly.





## SystemName

Format	<code>SystemName=&lt;system definition name&gt;</code>
Example	<code>SystemName=ComplianceSet</code>
Required	No, unless you want to use alias checking
Dynamic	No
Description	Name of the system definition that identifies the primary StorHouse system for storing aliases.

## MirrorName

Format	<code>MirrorName=&lt;system definition name&gt;</code>
Example	<code>MirrorName=ComplianceSet2</code>
Required	No, unless using a secondary StorHouse system for alias checking
Dynamic	No
Description	Name of the system definition that identifies the secondary StorHouse system for storing aliases.

# COLLECTORS section

The COLLECTORS section identifies one or more collector definitions. This section simply lists each collector definition name and corresponding collector definition. The format of each line in this section is as follows:

```
CollectorDefinitionName=CollectionDefinitionName
```



## Appendix A - The StorHouse/RFS configuration file

For example, the following COLLECTORS section identifies three collectors: `DDFILES`, `BIGFILES`, and `RFS31`. The same collection definition—`DIRECT`—is assigned to the first two collectors. The `DIRECT31` collection definition is assigned to the `RFS31` collector.

```
[COLLECTORS]
DDFILES=DIRECT
BIGFILES=DIRECT
RFS31=DIRECT31
```

See page 1-23 for more information about collector and collector definition names.

## System definition

A *system definition* identifies the StorHouse system to use for storing StorHouse collections, statistics, and aliases. For a StorHouse collection, each system definition is a unique destination defined by StorHouse system name, group name, VSET name, and FSET name. Any data destined for different VSETs and/or FSETs on a single StorHouse system or any data destined for different StorHouse systems requires a unique system definition in StorHouse/RFS. A system definition consists of the following parameters.



## DNSName

Format	DNSName=<any string>
Example	DNSName=alpha2
Required	Yes
Dynamic	No
Description	DNS name of the StorHouse system. This name must be the same as the Data Source Name (DSN) name of the ODBC driver instance as defined in the ODBC driver manager during StorHouse/RFS installation. This value can also be an IP address if the DBDriver is not specified.

## STHName

Format	STHName=<any string>
Example	STHName=sth1
Required	Yes
Dynamic	No
Description	Name of the StorHouse system. This name cannot exceed six characters. StorHouse/RFS uses this name to locate the complete system name in the SMCONFIG file (for UNIX) or the FILETEK.INI file (for Windows) for this StorHouse system.



## Appendix A - The StorHouse/RFS configuration file

### DBDriver

Format	DBDriver=<any string>
Example	DBDriver=StorHouse
Required	No
Dynamic	No
Description	<p>Name of the ODBC driver to use to access StorHouse databases. For a Windows environment, the configuration file utility automatically specifies the StorHouse/ODBC driver, or DBDriver=StorHouse. You must use a text editor to change or remove the value in a Windows environment. If you omit the DBDriver parameter and/or value, then StorHouse/RFS assumes you are using a DSN for the ODBC connection.</p> <p>You can specify all or part of the driver name. If you include part of the name, StorHouse/RFS matches as much of the driver name as provided. For example, the default "StorHouse" corresponds to "StorHouse Server 2.x (32 bit)". If your enterprise has multiple StorHouse drivers installed with different names, then specify more of the driver name. StorHouse/RFS uses the first driver that matches the DBDriver specified.</p>

### DBPort

Format	DBPort=<port number>
Example	DBPort=1990
Default	DBPort=1990
Required	No
Dynamic	No
Description	<p>Port number used by StorHouse/RFS to connect to the StorHouse database through ODBC. You must use a text editor to change the value in a Windows environment. This value is ignored if the DBDriver parameter and value are omitted.</p>



## DBHost

Format	DBHost=<any string>
Example	DBHost=
Default	DBHost=DSNName
Required	No
Dynamic	No
Description	Name or IP address of the host to use for the ODBC connection. You must use a text editor to change the value in a Windows environment. If you omit the DBHost, StorHouse/RFS uses the name specified for the DSNName parameter. This value is ignored if the DBDriver parameter and value are omitted.

## UserId

Format	UserId=<any string>
Example	UserId=RFSUSER
Required	Yes
Dynamic	No
Description	StorHouse account ID used by StorHouse/RFS for logging in to this StorHouse system to write StorHouse collections. This account must have the following StorHouse privileges to write StorHouse collections: ATF, DELETE, GET, PUT, RECORD, SETGROUP, SHOW, and VTF.



## Appendix A - The StorHouse/RFS configuration file

### Password

Format	Password=<any string>
Example	Password=* )&^*!
Required	Yes
Dynamic	No
Description	StorHouse account password (maximum 32 characters) associated with the UserId that is used to write StorHouse collections. The StorHouse/RFS configuration file utility encrypts the display of the password. In a UNIX environment, when you specify or change this password, you must run the gen_cfg program. See page 4-11 for instructions.

### PasswordType

The PasswordType parameter is used internally by StorHouse/RFS. Do not change this parameter or your system may not work properly.

### Group

Format	Group=<any string>
Example	Group=RFSGRP
Default	Default group for the account specified at the UserId parameter
Required	No
Dynamic	No
Description	Name of the StorHouse file access group used by StorHouse/RFS to access files on this StorHouse system.



## VSET

Format	VSET=<any string>
Example	VSET=RFSVSET
Default	Default VSET for the account specified at the UserId parameter
Required	No
Dynamic	No
Description	Name of the StorHouse volume set that will contain StorHouse collections written to this StorHouse system.

## FSET

Format	FSET=<any string>
Example	FSET=RFSFSET
Default	Default FSET for the account specified at the UserId parameter
Required	No
Dynamic	No
Description	Name of the StorHouse file set that will contain StorHouse collections written to this StorHouse system. If you are using a set of FSETs, that is, you specify a value for the FSETSegments parameter, then you must specify a value for the FSET parameter.



## Appendix A - The StorHouse/RFS configuration file

### FSETSegments

Format	FSETSegments=<number>
Example	FSETSegments=10
Default	FSETSegments=1
Required	No
Dynamic	No
Description	<p>Number of FSETs in a set of FSETs. You create the FSETs with StorHouse/Admin. Using a set of FSETs is beneficial when writing multiple collections to the same VSET. StorHouse/RFS writes to the first FSET in the set and then to the second FSET, and so on. Consider using the FSET segments feature if your organization requires write throughput greater than 8 MB per second. If you omit values for the FSET name and the FSETSegments parameters, then StorHouse/RFS uses the default FSET for the account. See “Using a set of FSETs for faster writing” on page 1-9 for more information. If you specify a value of 0, StorHouse/RFS assumes 1 FSET, or FSETSegments=1.</p>





## RetryInterval

Format	<code>RetryInterval=&lt;minutes&gt;</code>
Example	<code>RetryInterval=15</code>
Default	<code>RetryInterval=0</code> , which means StorHouse/RFS always tries to log in to StorHouse/SM and StorHouse/RM whether they are up or down
Required	No
Dynamic	Yes
Description	<p>Number of minutes that StorHouse/RFS should wait before attempting to connect to a StorHouse/SM system or to a StorHouse/RM system that is unresponsive, or down. StorHouse/RFS performs separate checks for StorHouse/SM and for StorHouse/RM. In other words, if StorHouse/SM is up but StorHouse/RM is down, then the retry interval applies to StorHouse/RM. If both systems are down, then the retry interval applies to both systems.</p> <p>StorHouse/RFS marks a system as down after five consecutive failed write or retrieval requests. After the <code>RetryInterval</code> interval has expired, StorHouse/RFS attempts to connect to the unresponsive system and if still down, marks the system as down and tries again at the next <code>RetryInterval</code> interval. The minimum value (if not 0) is 1. The maximum value is 2.1 billion.</p>



## Appendix A - The StorHouse/RFS configuration file

### TimeoutOverride

Format	TimeoutOverride=<minutes>
Example	TimeoutOverride=2
Default	Dynamic timeout value based on the media type or if StorHouse/RFS can't determine the media, it uses a timeout value of four minutes.
Required	No
Dynamic	Yes
Description	Number of minutes to override the dynamic timeout performed by StorHouse/RFS on reads, writes, opens, creates, and closes of StorHouse collections.

### VTF

Format	VTF=<DIRECT   NOW   NEXT>
Example	VTF=DIRECT
Default	VTF=NEXT
Required	No
Dynamic	No
Description	<p>Option that determines when StorHouse/RFS writes collections to their resident FSETs as well as whether to create performance copies of collections in the performance buffer. VTF stands for Vulnerability Time Factor.</p> <ul style="list-style-type: none"><li>■ DIRECT – Bypass the performance buffer and write the collection directly to the primary FSET.</li><li>■ NOW – Write the collection to the performance buffer and then copy the collection to the primary FSET immediately.</li><li>■ NEXT – Write the collection to the performance buffer and then copy the collection to the primary FSET at the next StorHouse write-back operation.</li></ul>



## CheckPoint

Format	CheckPoint=<MB>
Example	CheckPoint=500
Default	CheckPoint=100
Required	No
Dynamic	Yes
Description	<p>Maximum checkpoint value, in megabytes, for writing StorHouse collections. A checkpoint value specifies the maximum data extent size of a StorHouse collection. When a data extent reaches the maximum checkpoint size, StorHouse/RFS creates a new data extent. Should a write process fail, StorHouse/RFS can recover data extents to the last successful checkpoint. This value cannot exceed 1800 and cannot be less than the current StorHouse/RFS default, which may vary by StorHouse/RFS release. You can determine the default checkpoint value by restarting StorHouse/RFS or rereading the configuration file. During a restart or reread, StorHouse/RFS logs a message indicating the default checkpoint value.</p>



## Appendix A - The StorHouse/RFS configuration file

### MailRecipient

Format	MailRecipient=<e-mail address>
Example	MailRecipient=mjones@cc.com
Required	No
Dynamic	No
Description	<p>E-mail address where StorHouse/RFS sends messages when the following events occur:</p> <ul style="list-style-type: none"><li>■ StorHouse/RM goes down and comes up</li><li>■ StorHouse/SM goes down and comes up</li><li>■ A particular collection cannot be written to StorHouse/SM for the first time</li><li>■ The collection that could not be written is successfully written</li><li>■ The file locator data for a particular collection cannot be loaded into StorHouse/RM for the first time</li><li>■ The file locator data that could not be loaded is successfully loaded</li><li>■ The staging space is full</li><li>■ The collection space is full</li></ul> <p>If you omit this value and/or the MailServer value in the RFS section, no e-mail is sent.</p>

### Storage definition

A *storage definition* contains information for storing file locator data and collection metadata. Multiple collection sets can use the same storage definition.



## SystemName

Format	SystemName=<system definition name>
Example	SystemName=mailboxset
Required	Yes
Dynamic	No
Description	Name of the system definition that identifies the primary StorHouse system to contain file locator data and collection metadata.

## MirrorName

Format	MirrorName=<system definition name>
Example	MirrorName=mailboxset2
Required	No, unless using a secondary StorHouse system for StorHouse/RFS duplexing
Dynamic	No
Description	Name of the system definition that identifies the secondary StorHouse system. StorHouse/RFS writes data to this StorHouse system after writing the data to the primary StorHouse system.

## Database

Format	Database=<database name on StorHouse>
Example	Database=RFSDATA
Required	Yes
Dynamic	No
Description	Name of the StorHouse database that contains the set of StorHouse tables identified at the TableName parameter. You create the database with StorHouse/Admin. The database name is case sensitive.



## Appendix A - The StorHouse/RFS configuration file

### TableName

Format	TableName=<owner.base_table_name>
Example	TableName=MACCOUNT.MAILBOXTABLE
Required	Yes
Dynamic	No
Description	Owner and base table name of the StorHouse tables to contain file locator data and collection metadata. You use the tblgen utility to create the tables. The TableName parameter in the StorHouse/RFS configuration file must match the base table name you provide when running the tblgen utility. See “Creating StorHouse tables” on page 2-20 for more information about using the tblgen utility.

### UserId

Format	UserId=<StorHouse account>
Example	UserId=SYSADM
Required	Yes
Dynamic	No
Description	StorHouse account ID used to log in to the StorHouse system to load file locator data and collection metadata and to query the StorHouse tables. This account must be the owner of the StorHouse table specified at the TableName parameter.



## Password

Format	Password=<StorHouse password>
Example	Password=* )&^*!
Required	Yes
Dynamic	No
Description	StorHouse account password (maximum 32 characters) associated with the UserId that is used to load and query the StorHouse tables. The StorHouse/RFS configuration file utility encrypts the display of the password. In a UNIX environment, when you specify or change this password, you must run the gen_cfg program. See page 4-11 for instructions.

## PasswordType

The PasswordType parameter is used internally by StorHouse/RFS. Do not change this parameter or your system may not work properly.

## MaxSearchConnections

Format	MaxSearchConnections=<number>
Example	MaxSearchConnections=5
Default	MaxSearchConnections=0, which means unlimited number of ODBC connections
Required	No
Dynamic	No
Description	Maximum number of concurrent ODBC connections that can be used at any time for accessing the StorHouse tables to load or query file locator data and collection metadata.



## Appendix A - The StorHouse/RFS configuration file

### SearchConnectionTimeout

Format	<code>SearchConnectionTimeout=&lt;minutes&gt;</code>
Example	<code>SearchConnectionTimeout=5</code>
Default	<code>SearchConnectionTimeout=0</code> , which means StorHouse/RFS closes the connection when the search is done
Required	No
Dynamic	Yes
Description	Length of time an ODBC connection must be idle before it is released. This parameter is used only if <code>MaxSearchConnections</code> is specified.

### Collection definition

A *collection definition* defines collection options. Each collection set requires a collection definition. A collection definition contains the following parameters.

#### Storage

Format	<code>Storage=&lt;storage definition name&gt;</code>
Example	<code>Storage=STORAGE_RFS31</code>
Required	Yes
Dynamic	No
Description	Name of the storage definition that contains specifications for storing file locator data and collection metadata. Multiple collections can use the same storage definition.





## CollectionDir

Format	CollectionDir=<fully qualified path>
Example	CollectionDir=C:\collections\mailboxes CollectionDir=/rfs/control
Required	Yes
Dynamic	No
Description	Fully qualified path where StorHouse/RFS creates file locator data, or load (.ldr) files. This directory can be on the StorHouse/RFS server or on a device accessible to the server.

## MaxLoadInterval

Format	MaxLoadInterval=<minutes>
Example	MaxLoadInterval=600
Default	MaxLoadInterval=1440 (one day)
Required	No
Dynamic	Yes
Description	<p>Longest time to wait between loads (writing StorHouse collections and inserting file locator data into StorHouse tables), regardless of how much data has been collected. StorHouse/RFS uses this parameter in conjunction with MaxWriteSize to determine when it writes StorHouse collections. The interval starts when a collector collects the first file in the local collection.</p> <p>The minimum value is 30 minutes. If you specify a value less than 30 (for instance, 10 or 20), StorHouse/RFS sets the MaxLoadInterval to 30. If you specify 0, StorHouse/RFS uses the default value (1440).</p>



## Appendix A - The StorHouse/RFS configuration file

### MaxWriteSize

Format	MaxWriteSize=<MB>
Example	MaxWriteSize=1000
Default	MaxWriteSize=500
Required	No
Dynamic	Yes
Description	<p>Largest size a local collection can get before StorHouse/RFS closes it and starts a new one. The minimum value is 100 MB and the maximum value is 2 GB. StorHouse/RFS places any file larger than the MaxWriteSize into its own collection. For example, StorHouse/RFS writes a 100 GB file as a single collection to StorHouse. StorHouse/RFS places any file smaller than MaxWriteSize into a local collection along with the other files. The maximum number of files allowed in a collection, however, is 200,000. If a local collection has not exceeded the MaxWriteSize but contains more than 200,000 files, StorHouse/RFS closes the current collection and starts a new one.</p>



## Retention

Format	Retention=<FOREVER   number of days>
Example	Retention=10
Default	Retention=0, which means files may be deleted after they are collected and file versions may be created
Required	No
Dynamic	No
Description	<p>Number of days to retain a file before it may be deleted or overwritten. When a file is retrieved, StorHouse/RFS determines the expiration date by adding the retention days to the file's last modified time provided by the operating system. If the last modified time is less than the current time, the retention has expired. For example, if a user archives a file with a last modified time of noon on February 15 and the Retention=10, then the retention expires at noon on February 25. A user may delete, modify, or overwrite a file after the expiration.</p> <p>The retention number may be 0 through 65000 days (or 0 years and 0 days through 178 years).</p> <ul style="list-style-type: none"><li>■ If you specify 0 (the default), no retention period applies to files in the collection set. Files may be deleted after they are collected and file versions may be created.</li><li>■ If you specify a number greater than 0, no file versions are allowed. A file may be deleted, modified, or overwritten only after the retention period expires.</li><li>■ If you specify FOREVER, the file may not be deleted or overwritten.</li><li>■ If you specify a value less than 0, StorHouse/RFS assumes FOREVER and prohibits a file from being deleted, modified, or overwritten.</li></ul> <p>You can change the retention at any time. The new value applies only to uncollected files. In other words, any file collected after the change has the new retention and any file collected before the change has the old retention.</p> <p>You can also remove the retention, that is, change a retention period to 0. In this case, StorHouse/RFS may create file versions and allow file versions to be deleted with the exception of the original file version.</p>



## Appendix A - The StorHouse/RFS configuration file

### Compression

Format	Compression=<YES NO>
Example	Compression=YES
Default	Compression=NO, which means the StorHouse collections are not compressed
Required	No
Dynamic	No
Description	Option to compress the StorHouse collections. Compression rates vary by the type of source data written to StorHouse/RFS. Compression costs CPU time but reduces storage costs and network bandwidth requirements. If you have available CPU time and want to reduce the amount of storage required on StorHouse, then specify COMPRESSION=YES. No compression occurs if the value is NO or if you omit the parameter from the StorHouse/RFS configuration file.

### Browse

Format	Browse=<YES NO>
Example	Browse=YES
Default	Browse=NO
Required	No
Dynamic	Yes
Description	Option to enable or disable file browsing in the virtual file system for this collection set. When enabled, authorized users can browse files in the collection set using standard operating system tools, such as Windows Explorer or the ls command. When disabled, no browsing is allowed.



## Collector definition

A *collector definition* defines each StorHouse/RFS collector, including where the collector looks for files to collect. You assign each collector to a collection definition. You can assign multiple collectors to the same collection definition. A collector definition consists of the following parameters.

### StagingDir

Format	StagingDir=<fully qualified path>
Example	StagingDir=c:\email StagingDir=/stage
Required	Yes
Dynamic	No
Description	<p>Path where StorHouse/RFS stages files to be collected and where a collector checks for files to collect. The StagingDir can be the same for all collectors, or different for each collector, or any combination. This directory can be on the StorHouse/RFS server or on a device accessible to the server. It must be located at least one level below the root level of a file system or drive. A user or application must have read and write permission to access the StagingDir to archive files.</p> <p>The full collection path is composed of the StagingDir and the UserDir. For example, assume the following:</p> <ul style="list-style-type: none"><li>■ StagingDir is c:\email</li><li>■ UserDir is \mailboxes</li></ul> <p>A collector then collects files in c:\email\mailboxes. StorHouse/RFS displays the UserDir, but not the StagingDir, in the virtual file system.</p>



## Appendix A - The StorHouse/RFS configuration file

### UserDir

Format	UserDir=<subdirectory under StagingDir>
Examples	UserDir=\mailboxes UserDir=/journal
Default	UserDir=\ (for Windows) or UserDir=/ (for UNIX), which means collect all files written to the StagingDir
Required	No
Dynamic	No
Description	Subdirectory under the StagingDir where StorHouse/RFS stages files to be collected and where a collector checks for files to collect. Wildcard characters are not allowed in the UserDir, for instance, \mailboxes* is invalid. You can use the default (/ or \) once per StorHouse/RFS configuration. The UserDir value must be unique. StorHouse/RFS creates a folder or directory for the UserDir in the virtual file system.



## MaxStagingSpace

Format	MaxStagingSpace=<MB>
Example	MaxStagingSpace=500
Required	Yes
Dynamic	Yes
Description	<p>Maximum amount of storage (in MB) to use for writing files to the staging area. When the maximum amount is reached, StorHouse/RFS returns an out of space condition appropriate for the operating system. Files may resume being written to the staging area only when space permits. If you specify 0, StorHouse/RFS reserves 100 MB of staging space.</p> <p>You set a MaxStagingSpace for each StagingDir and UserDir combination. For example, you set different MaxStagingSpace values for these two staging areas:</p> <p>c:\email\mailboxes c:\email\journal</p>



## Appendix A - The StorHouse/RFS configuration file

### MaxCollectionSpace

Format	MaxCollectionSpace=<MB>
Example	MaxCollectionSpace=500
Default	Value of the MaxWriteSize parameter
Required	Yes
Dynamic	Yes
Description	Maximum amount of disk space, in megabytes, that StorHouse/RFS can use to collect files for this collector. The sum of all MaxCollectionSpace parameters is the amount of rename space that can be used by all collectors regardless of staging directory. All files are collected by renaming them from the staging area to a rename directory, which StorHouse/RFS creates automatically. The rename directory appears at the same level as the staging directory but includes the name of the collection that it is creating. If the rename space becomes full, StorHouse/RFS performs cleanup routines or stops renaming files to the rename directory until it writes the local collection to StorHouse. The minimum value is the value of the MaxWriteSize parameter.

### WaitTime

Format	WaitTime=<minutes>
Example	WaitTime=60
Required	Yes
Dynamic	Yes
Description	Number of minutes a file must be idle in a staging area before it is eligible for collection. This parameter prevents files that are in the process of being written from being collected prematurely. The collector checks the last modified time of each file in the staging area, and when the file has aged, the collector renames the file to the rename directory. If this value is set to 0, StorHouse/RFS changes it to 10. The minimum value is 1.





## Group

Format	Group=<name>
Example	Group=admin
Default	None, which means no group access in a Windows environment
Required	No
Dynamic	No
Description	In a Windows environment, name of the default primary group to use for security. If you omit this value, StorHouse/RFS assumes only the file owner can access files collected by this collector. In a UNIX environment, you can omit this value because StorHouse/RFS obtains the primary group name during the collection process. The group name is stored with the file locator data if group access is allowed.



## Appendix A - The StorHouse/RFS configuration file

### Permissions

Format	Permissions=<R   W   X>
Examples	Permissions=R Permissions=R W X
Default	None, which means no other permissions are allowed, that is, only the file owner or group member may access files and directories collected by this collector
Required	No
Dynamic	No
Description	<p>Permissions to share files across StorHouse/RFS systems that have access to the same StorHouse table array. The values have the same meaning as the UNIX “Other” permissions.</p> <ul style="list-style-type: none"><li>■ R – read a file and list files in a directory</li><li>■ W – modify a file and create and delete files in a directory</li><li>■ X – execute a file (for instance, when the file is a program) or access a directory (for instance, use cd)</li></ul> <p>In a UNIX environment, these permissions override any “other” permissions set with standard UNIX commands.</p>



## KeepSubdirectories

Format	KeepSubdirectories=<number>
Example	KeepSubdirectories=4
Default	KeepSubdirectories=10000
Required	No
Dynamic	Yes
Description	<p>Number of levels of subdirectories, below the collector root, that must exist at all times. If your operating system supports security such as Windows security descriptors or ACLs, then you will provide this value when running the tblgen utility to create a security table. Once you create a security table with the tblgen utility, StorHouse/RFS ignores any value in the StorHouse/RFS configuration file. The minimum value is 0 (if no user directory, or UserDir) or the level at which the user directory exists.</p> <p>For UNIX systems with standard security, you can omit this parameter or set the value to 0. StorHouse/RFS already captures all basic UNIX file permissions and ownership information.</p>



## **Appendix A - The StorHouse/RFS configuration file**

# I N D E X



## Symbols

- \_0 table 1-4, 2-22
- \_ALIASES table 1-4, 4-4
- \_C table 1-4, 2-22
- \_RFS table 1-4, 6-9
- \_S table 1-4, 2-22
- \_SYSTEMS table 1-4, 6-9
- \_TABLES table 1-4, 6-9

## A

- account, StorHouse A-22, A-26, A-31, A-40
- adding
  - collection definition 2-38
  - collector definition 2-42
  - ODBC data source 3-2
  - storage definition 2-32
  - system definition 2-25
- aliases
  - database 4-3
  - definition 1-3, 1-5, 4-7
  - setting parameters 4-5
  - table 4-4

## ALIASES section

- Database A-25
- MirrorName A-27
- Password A-26
- SystemName A-27
- TableName A-25
- UserId A-26

## B

- backing up
  - StorHouse collections 5-2
  - StorHouse tables 5-5
- Browse parameter A-46

## C

- CacheDir parameter A-13
- CheckPoint parameter A-37
- collection
  - intervals 1-14
  - local 1-2
  - path A-47
  - StorHouse 1-2
- collection definition



## Index

- adding 2-38
  - Browse A-46
  - CollectionDir A-43
  - Compression A-46
  - deleting 5-10
  - description 1-5, 2-39, 2-40
  - example 2-39, 2-40
  - MaxLoadInterval A-43
  - MaxWriteSize A-44
  - naming 1-25
  - purpose A-42
  - Retention A-45
  - statistics 6-3
  - Storage A-42
  - collection metadata 1-2
  - collection set 1-2
  - collection strategy 1-13
  - CollectionDir parameter A-43
  - collections table 1-4
  - collector definition
    - adding 2-42
    - deleting 5-10
    - description 1-5
    - example 2-44
    - Group A-51
    - KeepSubDirectories A-53
    - MaxCollectionSpace A-50
    - MaxStagingSpace A-49
    - naming 1-25
    - Permissions A-52
    - purpose 1-13, A-47
    - StagingDir A-47
    - UserDir A-48
    - WaitTime A-50
  - Compression parameter A-46
  - creating
    - account 2-14
    - aliases 4-8
    - aliases database 4-3
    - aliases table 4-4
    - database 2-18
    - FSET 2-11
    - group 2-6
    - primary file 5-12
    - statistics database 6-11
    - statistics tables 6-11
    - table 2-20
    - VSET 2-8
  - current statistics interval type 6-2
- ## D
- data source
    - ODBC 3-2
    - StorHouse 3-3
  - Database parameter A-21, A-39
  - database, StorHouse 2-22
  - DBDriver parameter A-30
  - DBHost parameter A-31
  - DBPort parameter A-30
  - definition
    - collection A-42
    - collector A-47
    - names A-4
    - storage A-38
    - system A-28
  - deleting
    - collection definition 5-10
    - collection set 5-7
    - collector definition 5-10
    - storage definition 5-10
    - system definition 5-9
    - unreadable primary file 5-11
  - displaying
    - aliases 4-8
    - current statistics 6-20
    - local statistics 6-18
  - DNS 2-22
  - DNSName parameter A-29
  - duplexing 3-1



dynamic parameter A-5

## E

exclusions list 7-4  
EXCLUSIONS section A-24

## F

file  
    format A-2  
    maintenance A-2  
    sections A-2  
file locator data 1-2, 1-10  
FileCleanupTimeout parameter A-15  
FILETEK.INI file A-29  
FileType parameter A-21  
FIND command 7-16  
FINDB command 7-16  
Flush parameter A-17  
FSET  
    creating 2-11  
    definition 1-3  
    example 2-12, 2-13  
    naming 1-23  
FSET parameter A-33  
FSETSegments parameter A-34

## G

general statistics 6-2  
general StorHouse/RFS parameters 2-48  
group  
    creating 2-6  
    definition 1-3  
    naming 1-23  
Group parameter A-32, A-51

## I

interval statistics interval type 6-2  
interval type 6-2, 6-10  
ISQL 6-25

## K

KeepSubDirectories parameter A-53

## L

load file 1-2  
local collection 1-2  
local statistics file 6-6  
LocalPath parameter A-13  
log 7-2  
log entries A-14  
LogFile parameter A-14  
logging in to StorHouse/Admin 2-5  
logging out of StorHouse/Admin 2-6

## M

MailRecipient parameter A-38  
MailServer parameter A-14  
MaxCacheSpace parameter A-15  
MaxCollectionSpace 1-19  
MaxLoadInterval parameter A-43  
MaxSearchConnections parameter A-41  
MaxSMFiles parameter A-16  
MaxSMWriters parameter A-16  
MaxStagingSpace parameter A-49  
MaxWriteSize parameter A-44  
metadata  
    backup utility 5-5  
    restore utility 5-12  
    scheduling backups 5-6  
MirrorName parameter 3-6, A-24,



## Index

A-27, A-39

## N

naming conventions 1-23

## O

ODBC driver data sources 3-2

## P

parameters

- Browse A-46
- CacheDir A-13
- Checkpoint A-37
- CleanupTimeout A-18
- CollectionDir A-43
- Compression A-46
- Database A-21, A-25, A-39
- DBDriver A-30
- DBHost A-31
- DBPort A-30
- definition A-4
- DNSName A-29
- dynamic A-5
- FileCleanupTimeout A-15
- FileType A-21
- Flush A-17
- FSET A-33
- FSETSegments A-34
- Group A-32, A-51
- KeepSubDirectories A-53
- LocalPath A-13
- LogFile A-14
- MailRecipient A-38
- MailServer A-14
- MaxCacheSpace A-15
- MaxCollectionSpace A-50
- MaxLoadInterval A-43

MaxSearchConnections A-41

MaxSMFiles A-16

MaxSMWriters A-16

MaxStagingSpace A-49

MaxWriteSize A-44

MirrorName A-24, A-27, A-39

Password A-23, A-26, A-32, A-41

Permissions A-52

Retention A-45

RetryInterval A-35

SafetyPath A-19

SearchConnectionTimeout A-42

Simulate A-18

StagingDir A-47

static A-5

StatsInterval A-20

STHName A-29

Storage A-42

SystemName A-23, A-27, A-39

TableName A-22, A-25, A-40

TimeoutOverride A-36

UserDir A-48

UserId A-22, A-26, A-31, A-40

Version A-19

VSET A-33

VTF A-36

WaitTime A-50

Password parameter A-23, A-26, A-32, A-41

password, account 1-24, 2-15

Permissions parameter A-52

planning reference 1-2

## Q

querying statistics tables 6-21

## R

recovering





- StorHouse collections 5-11
- StorHouse tables 5-12
- reporting information for a user file 7-15
- restarting StorHouse/RFS 7-8
- Retention parameter A-45
- RetryInterval parameter A-35
- RFS section
  - CacheDir A-13
  - Cleanup Timeout A-18
  - FileCleanupTimeout A-15
  - Flush A-17
  - LocalPath A-13
  - LogFile A-14
  - MailServer A-14
  - MaxCacheSpace A-15
  - MaxSMFiles A-16
  - MaxSMWriters A-16
  - SafetyPath A-19
  - Simulate A-18
  - Version A-19
- rfconfig.exe A-2

## S

- SafetyPath parameter A-19
- scheduling
  - collection backups 5-4
  - database backups 5-6
- SearchConnectionTimeout parameter A-42
- section names A-3
- security entries 1-3
- security table 1-4
- server, StorHouse/Control Center 3-2
- setup procedures 2-2
- shutdown statistics interval type 6-2
- Simulate parameter A-18
- SM\_LOCAL keyword 7-16
- SMCONFIG file A-29

- SQL
  - statistics queries 6-23
- StagingDir parameter A-47
- starting
  - StorHouse/Admin 2-5
  - StorHouse/RFS configuration file utility 2-25
  - StorHouse/RFS service 7-8
- startup statistics interval type 6-2
- static parameter A-5
- statistics
  - collection definition 6-3
  - current 6-6, 6-20
  - database 6-9, 6-11
  - definition 1-5
  - description 6-2
  - displaying local statistics 6-18
  - example text file 6-7
  - example XML file 6-8
  - general 6-2
  - interval 6-10
  - interval time 6-11
  - interval type 6-2, 6-10
  - local file 6-6
  - parameters 6-16
  - setting parameters 6-13
  - system definition 6-5
  - tables 1-4, 6-9
- STATS section
  - Database A-21
  - FileType A-21
  - MirrorName A-24
  - Password A-23
  - StatsInterval A-20
  - SystemName A-23
  - TableName A-22
  - UserId A-22
- StatsInterval parameter A-20
- STHName parameter A-29



## Index

- stopping
    - StorHouse/RFS configuration file utility 2-25
    - StorHouse/RFS service 7-8
  - storage definition
    - adding 2-32
    - Database A-39
    - deleting 5-10
    - description 1-5, 2-35, 2-36
    - example 2-34, 2-35
    - MaxSearchConnections A-41
    - MirrorName A-39
    - naming 1-25
    - Password A-41
    - SearchConnectionTimeout A-42
    - SystemName A-39
    - TableName A-40
    - UserId A-40
  - Storage parameter A-42
  - storage strategy 1-7
  - StorHouse account
    - creating 2-14
    - definition 1-3
    - naming 1-24
    - privileges 2-14
  - StorHouse collection 1-2
  - StorHouse database 2-22
    - concurrent access 1-11
    - creating 2-18
    - definition 1-4
    - example 2-18, 2-19, 6-11
    - naming 1-24
  - StorHouse table
    - creating 2-20
    - definition 1-4
    - naming 1-24
  - StorHouse/Admin xiii, 2-5
  - StorHouse/Control Center 3-2
  - StorHouse/RFS
    - configuration file example A-9
    - configuration file utility 2-25, A-2
    - general parameters 2-48
    - log 7-2
    - restarting 7-8
  - submitting SQL statements 6-25
  - system definition
    - adding 2-25
    - CheckPoint A-37
    - DBDriver A-30
    - DBHost A-31
    - DBPort A-30
    - deleting 5-9
    - description 1-5, 2-28, 2-29
    - DNSName A-29
    - DNSName parameter A-29
    - example 2-27, 2-29
    - FSET A-33
    - FSETSegments A-34
    - Group A-32
    - MailRecipient A-38
    - naming 1-24
    - password A-32
    - RetryInterval A-35
    - statistics 6-5
    - STHName A-29
    - TimeoutOverride A-36
    - UserId A-31
    - VSET A-33
    - VTF A-36
  - SystemName parameter 3-6, A-23, A-27, A-39
- ## T
- table array 1-4
  - TableName parameter A-22, A-25, A-40
  - tblgen utility 2-20, 6-11
  - TimeoutOverride parameter A-36



## **U**

- unused file A-15
- user file 1-2
- UserDir parameter A-48
- UserId parameter A-22, A-26, A-31, A-40

## **V**

- validating the system 7-10
- Validation and Health Report 7-10
- Version parameter A-19
- virtual file system xii
- volumes 1-3
- VSET
  - creating 2-8
  - definition 1-3
  - erasing 5-9
  - example 2-9, 2-10
  - exporting 5-8
  - naming 1-23
  - uncataloging 5-8
- VSET parameter A-33
- VTF parameter A-36

## **W**

- WaitTime parameter A-50

## **X**

- XMLFIND command 7-16
- XMLFINDB command 7-16