sgi

Using StorHouse/FTP

Publication Number
007-6327-001

November 19, 2013

StorHouse®

Publication Number 007-6327-001

LIMITED RIGHTS LEGEND

TRADEMARKS AND ATTRIBUTIONS

Contents

Welcome

Welcome to *Using StorHouse/FTP*. StorHouse/FTP is an optimized FTP server that provides high-speed, sequential retrieval of very large (multiple gigabyte) StorHouse files from any standard FTP client. FTP retrieval capability is in addition to the native NFS and CIFS support that StorHouse/RFS Linux platforms already provide.

Purpose of This Manual

This manual explains how to configure a Linux StorHouse/RFS server for use with StorHouse/FTP. It also describes how to use the software to retrieve StorHouse files to a destination directory on a target client system.

Audience

This document is for StorHouse/RFS system administrators who are responsible for installing, configuring, and maintaining StorHouse/RFS. It assumes that administrators are familiar with StorHouse/RFS concepts and operation and know how to update the StorHouse/RFS configuration file in a Linux environment. It also assumes administrators are knowledgeable about FTP.

The document is also for client users who will be accessing StorHouse/FTP to retrieve very large files from StorHouse. It assumes that these users understand FTP concepts and operation and know how to retrieve files using a standard FTP interface.

Contents

This manual contains two chapters:

- Chapter 1, “Introduction and Setup,” defines StorHouse/FTP and explains how to set up your StorHouse/RFS Linux server prior to using the StorHouse/FTP feature.
- Chapter 2, “Retrieving Files with StorHouse/FTP,” explains how to use StorHouse/FTP to retrieve very large files from StorHouse. The chapter also provides a file retrieval example.

CHAPTER 1

Introduction and Setup

This chapter defines StorHouse/FTP and explains how to set up your StorHouse/RFS Linux server prior to using the StorHouse/FTP feature.

About StorHouse/FTP

StorHouse/RFS for Linux Version 4.0.65.1 and later support StorHouse/FTP – an optimized FTP server that enables users to retrieve very large (multiple gigabyte) StorHouse files at high speeds through any standard FTP client. By supporting rapid retrieval of very large files, StorHouse/FTP boosts StorHouse/RFS performance in environments where users frequently retrieve very large files.

StorHouse/FTP retrieval is in addition to the native NFS and CIFS support that StorHouse/RFS Linux platforms already provide. In essence, having NFS, CIFS, and FTP interfaces enables customers to optimize retrieval performance based on client operating system requirements, application use cases, and file size.

Installation of the StorHouse/FTP server is a standard part of the Linux StorHouse/RFS installation procedure. Once the StorHouse/RFS software is installed, you must perform some additional setup tasks as defined in the following section.

Setting up the StorHouse/FTP Server

To use StorHouse/FTP, you must install StorHouse/RFS and then create a new StorHouse/FTP configuration file on your Linux StorHouse/RFS server. You must also update the existing StorHouse/RFS configuration file with a new [FTPD] section and parameters for StorHouse/FTP.

Optionally, you can also:

- Change the FTP port numbers if you cannot use the defaults
- Configure Secure FTP (FTPS) for StorHouse/FTP

Creating a new StorHouse/FTP Configuration File

As part of the StorHouse/RFS installation process, the software creates a sample StorHouse/FTP configuration file called `rfsftp.v4r0` in the `/etc/pam.d` directory on the StorHouse/RFS server. This file contains information for StorHouse/FTP to handle client authentication requests. To deploy the `rfsftp.v4r0` file for live use, copy the file in the same location and rename it to `rfsftp`.

To further configure your StorHouse/FTP server to benefit from additional authentication parameters (PAM), consult your Linux system documentation.

Updating the StorHouse/RFS Configuration File with New Parameters for StorHouse/FTP

Before you can use StorHouse/FTP, you must add a new section, [FTPD], to the StorHouse/RFS configuration file. Add the new section somewhere under the [RFS] section.

The [FTPD] section has six parameters. The EnableFTP parameter is required so you must add it to the [FTPD] section. The other parameters are optional. You can add them if necessary at your installation or to override default values.

The [FTPD] configuration parameters are:

- CertsDir
- EnableFTP
- FtpDebug
- MaxFTPSMreaders
- SecureLoginsOnly
- SecureXfrsOnly

CertsDir

Format	`CertsDir=<dir_name>`
Examples	`CertsDir=/usr/local/ssl/certs`
Default	`CertsDir=/usr/local/ssl/certs`
Required	No
Dynamic	Yes
Description	Specifies the UNIX-style pathway to secure certificate and key files on the StorHouse/RFS Linux server. StorHouse/RFS uses these certificates and key files to support FTPS sessions between the StorHouse/RFS server and remote clients. If you omit this parameter or specify it without a value, StorHouse uses the default. For information about how to generate certificate files, refer to the section, “Configuring FTPS for StorHouse/FTP” at the end of this chapter.

EnableFTP

Format	`EnableFTP=<YES|NO>`
Examples	`EnableFTP=NO`
Default	NO
Required	YES
Dynamic	Yes
Description	Controls the availability of the StorHouse/FTP feature. YES indicates the StorHouse/FTP feature is turned on. NO indicates the feature is turned off.

FtpDebug

Format	`FtpDebug=<0|1>`

Examples	`FtpDebug=0 (to disable debug logging)`
	`FtpDebug=1 (to enable debug logging)`

Default	`FtpDebug=0`
Required	No
Default	Yes
Description	Controls the level of FTP session information logging reported to the StorHouse/RFS log file. (The LogFile parameter in the [RFS] section of the StorHouse/RFS configuration file defines the log location.) You can also control FTP debugging in a session-only manner by passing the command `site debug <value>` in an established StorHouse/RFS FTP session. In this command, use a value of 1 to enable logging and a value of 0 to disable it.

MaxFTPSMReaders

Format	`MaxFTPSMReaders=<number>`
Examples	`MaxFTPSMReaders=64`
Default	5
Minimum	1
Maximum	64
Required	No
Dynamic	Yes
Description	Controls the maximum number of allowable concurrent FTP retrievals from StorHouse. If you omit this parameter or specify it without a value, StorHouse uses the default value. SGI recommends that you set the VRAM_NUM_KS parameter as least as large as MaxFTPSMReaders.

SecureLoginsOnly

Format	`SecureLoginsOnly=<YES|NO>`
Examples	`SecureLoginsOnly=NO`
Default	`SecureLoginsOnly=NO`
Required	No
Dynamic	Yes
Description	Controls whether StorHouse/RFS allows unencrypted logins (for example, clear text passwords on an unsecured connection). If you omit this parameter or specify it without a value, StorHouse/RFS uses the default.

SecureXfersOnly

Format	`SecureXfersOnly=<YES|NO>`
Examples	`SecureXfersOnly=NO`
Default	`SecureXfersOnly=NO`
Required	No
Dynamic	Yes
Description	Controls whether StorHouse/RFS allows file retrievals without a secure connection. If you omit this parameter or specify it without a value, StorHouse/RFS uses the default.

Reconfiguring the Port for StorHouse/FTP

The StorHouse/FTP server uses TCP ports 2099 and 2100. If you are already using these ports for other applications, you can assign different port numbers to StorHouse/FTP. In general, however, SGI recommends that you use the assigned values.

Use the following procedure to assign port numbers other than 2099 and 2100 to StorHouse/FTP. When you change the port assignments, the `rfsftp-data` port must be one less than the `rfsftp` port.

▼ To assign port numbers to StorHouse/FTP

1. Locate the `/etc/services` file on your StorHouse/RFS server.
2. Once you have located the file, add the following two lines, substituting actual port numbers for `<port_number>`:

   – `rfsftp-data <port_number>/tcp`

   – `rsftp <port_number>/tcp`

Both lines are required.

Configuring FTPS for StorHouse/FTP

To support FTP encryption, you must generate/create keystore and certificate files on your StorHouse/RFS Linux server. You must place these files in the directory defined by the `CertsDir` parameter in the `[FTPD]` section of the StorHouse/RFS configuration file.

You can use the following `openssl` commands (if available on your system) to generate keystore and certificate files on your StorHouse/RFS server.

- `openssl dhparam -out dh_param_512.pem -2 512`
- `openssl dhparam -out dh_param_1024.pem 02 1024`
- `openssl req -new -x509 -newkey rsa:2048 -nodes -out rfsftpd.pem -keyout rfsftpd.pem`

Note the following:

- Each file created by the `openssl` commands must be readable only by root.
- You can create these files on another system and then copy them to your StorHouse/RFS server.
- FTPS is also referred to as FTPS external or FTP over TLS/SSL.

For a higher level of security, SGI recommends that you obtain certificate file(s) signed by an accredited Certificate Authority (CA). Alternatively, you can install self-signed certificate files on any applicable client systems that will be accessing StorHouse/FTP remotely.

As always, consult your Linux system documentation for additional information about system encryption, key and certificate generation, and use of certificate files for secure connectivity through FTP.

CHAPTER 2

Retrieving Files with StorHouse/FTP

This chapter explains how to use StorHouse/FTP to retrieve very large files from StorHouse and provides a retrieval example.

Using StorHouse/FTP

Once you have configured your Linux environment and your Linux StorHouse/RFS server, you can use any FTP client with a graphical or command line interface to retrieve files using StorHouse/FTP.

FTP Commands

When using a command line FTP client, StorHouse/FTP supports Linux commands such as `cd` and `ls` for navigation. While graphical user interface FTP clients have their own way to navigate directories, retrieve files, and perform other functions, they typically also allow users to send server-level commands directly.

Table 2-1 defines the format of some frequently used FTP commands and provides command examples.

Table 2-1: Frequently Used FTP Commands

FTP Commands	Description
`get <filename> <destination_file_path>`	Retrieves the specified file name `(<filename>)` to the specified directory location `(<destination_file_path>)` on the target system. If you omit `destination_file_path`, StorHouse/FTP retrieves the file to your default directory. Example: `get file0.0 /coll1/file1/file_0.0`
`opts mode z level n`	Compresses the file before sending it to the target system. The optional modifier, `level n`, specifies the compression level for retrievals. Valid values for `n` are: ▪ Minimum of 1 ▪ Maximum of 9 ▪ Default of 6 Example: `opts mode z level 5`
`site idle n`	Controls the number of seconds that FTP sessions can sit idle before timing out. Valid values for `n` are: ▪ Minimum of 1 ▪ Maximum of 86400 ▪ Default of 86400 Example: `site idle 600`
`site debug <value>`	Controls whether debugging of StorHouse/FTP activity will be written to the StorHouse/RFS log at the session level. Valid values are: ▪ 1 to indicate logging is on ▪ 0 to indicate no logging Example: `site debug 1`

In addition, Help commands such as `rhelp` provide information about server commands. For example, to access help for the `get` command, type:

```
rhelp RETR
```

Additional Information

Note the following about StorHouse/FTP:

- Initially, StorHouse/FTP tries to retrieve the requested file from the local StorHouse/RFS collection space. If the file does not exist in the local collection space, the software retrieves the file from StorHouse.
- A file may be written to StorHouse/RFS but not yet collected by the StorHouse/RFS software. If a file has not been collected, you cannot retrieve it using StorHouse/FTP. Only collected files can be retrieved.
- StorHouse/FTP performs CRC checking on file retrieval.
- StorHouse/RFS does not support FTP actions such as restart and the use of multiple sessions to retrieve a file more quickly.
- StorHouse/RFS versions prior to version 4.0.65.2 do not support reading links.
- StorHouse/FTP supports only BINARY mode data connections, not ASCII.

Retrieving Files

This section explains how to retrieve files using StorHouse/FTP. The example uses the standard Linux RedHat/CentOS FTP command line utility to:

- Connect to a StorHouse/RFS server called `HPRFS1` using the FTP listing port `2100`

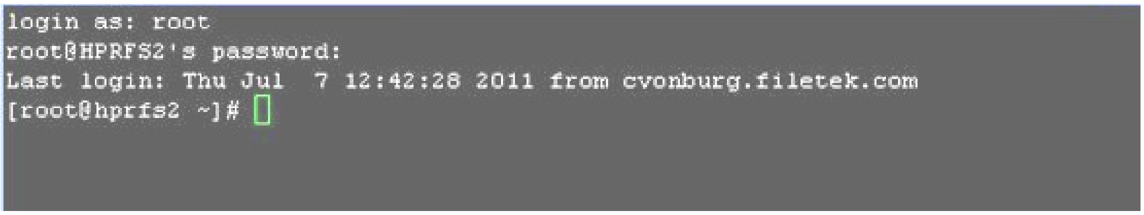
- Change to the `/rfs_1/June-14-2001-11_10_42/dir_0` directory where the source file resides
- Retrieve (`get`) the source file `file_0.0` and transfer it to the `/coll1/file1/file_0.0` directory on the target machine

Use the following procedure to retrieve `file_0.0` using StorHouse/FTP.

▼ To retrieve `file_0.0` using StorHouse/FTP

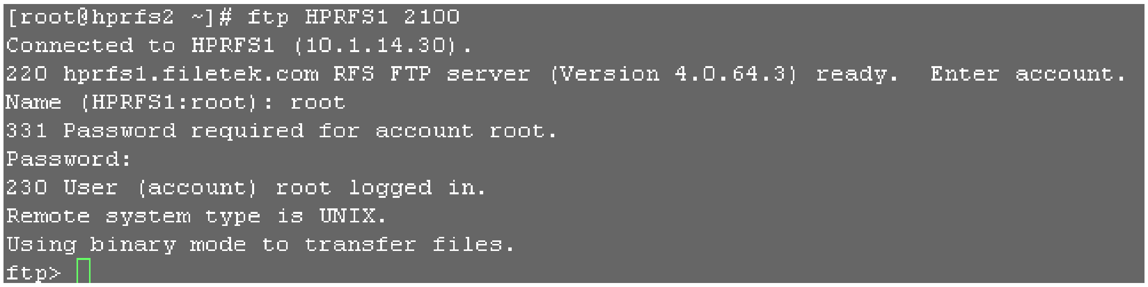
1. Log in to the target client platform to establish a terminal session to the StorHouse/RFS server using PuTTY or a similar application.

2. Type `ftp HPRFS1 2001` to establish an FTP session with the StorHouse/RFS server, and then press `Enter`.



```
login as: root
root@HPRFS2's password:
Last login: Thu Jul  7 12:42:28 2011 from cvonburg.filetek.com
[root@hprfs2 ~]# 
```

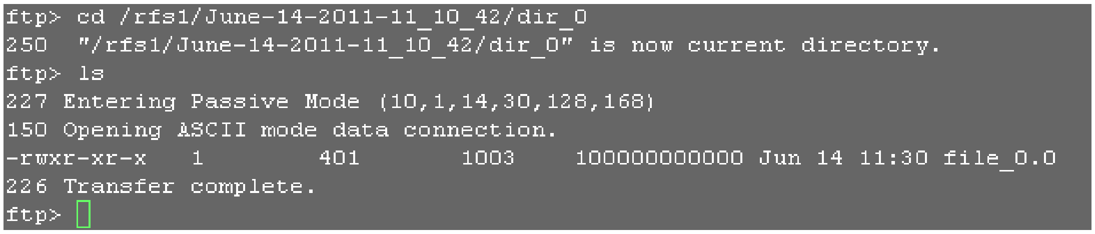
3. Type your Linux user name and password for authentication, and press `Enter` after each.



```
[root@hprfs2 ~]# ftp HPRFS1 2100
Connected to HPRFS1 (10.1.14.30).
220 hprfs1.filetek.com RFS FTP server (Version 4.0.64.3) ready.  Enter account.
Name (HPRFS1:root): root
331 Password required for account root.
Password:
230 User (account) root logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

After authenticating and opening a StorHouse/ FTP connection, you are at the root directory (/) of the StorHouse/RFS virtual file system. Within StorHouse/RFS, you may browse any virtual file system level that you are authorized to view. You may not navigate to a location outside the StorHouse/RFS virtual file system while connected to the StorHouse/FTP server.

4. Type `cd /rfs1/June-14-2011_10_42/dir_0` to change to the directory that contains the source file you want to retrieve, and press `Enter`.

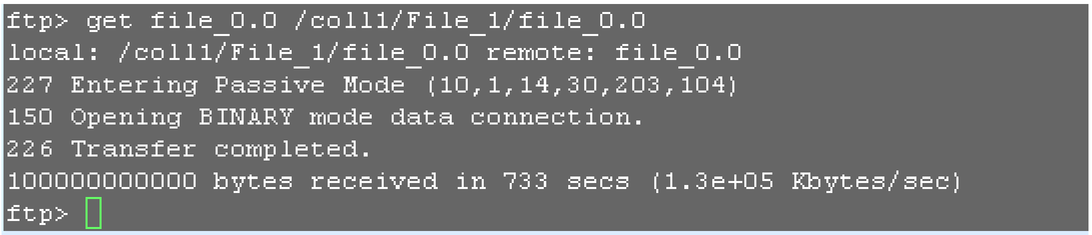


```
ftp> cd /rfs1/June-14-2011-11_10_42/dir_0
250  "/rfs1/June-14-2011-11_10_42/dir_0" is now current directory.
ftp> ls
227 Entering Passive Mode (10,1,14,30,128,168)
150 Opening ASCII mode data connection.
-rwxr-xr-x    1         401        1003     100000000000 Jun 14 11:30 file_0.0
226 Transfer complete.
ftp> 
```

5. To retrieve `file_0.0` and transfer it to the target directory `/coll1/File_1/file_0.0`, type:

   `get file_0.0 /coll1/File_1/file_0.0`

   and press `Enter`.



```
ftp> get file_0.0 /coll1/File_1/file_0.0
local: /coll1/File_1/file_0.0 remote: file_0.0
227 Entering Passive Mode (10,1,14,30,203,104)
150 Opening BINARY mode data connection.
226 Transfer completed.
100000000000 bytes received in 733 secs (1.3e+05 Kbytes/sec)
ftp> 
```

When the transfer is completed, you can use standard procedures to access the file on your destination system.