Secure FTP Server

User's Guide

GlobalSCAPE

Table Of Contents

Secure FTP Server

GlobalSCAPE's Secure FTP Server is a hardened file/data transfer server that provides secure data transactions over standard Internet protocols. It extends beyond standard FTP servers by providing support for:

- Multi-protocol support: FTP, FTP/S (SSL/TLS), optionally SFTP (SSH2) and HTTP/S (SSL)
- Post-transaction processing using highly configurable event rules
- Data reliability and integrity guarantees
- Automation of complex and time-consuming tasks
- Local and remote administration of multiple servers and/or sites
- Flexible authentication choices
- Highly configurable user, account, and site settings

Secure FTP Server provides:

Data Protection and Encryption

GlobalSCAPE Secure FTP (Secure FTP) Server protects intellectual property, trade secrets, and customer files transferred over the Internet using secure protocols including FTPS (SSL/TLS), and optionally SFTP (SSH2) and HTTP/S (SSL).

Guaranteed Delivery and Data Integrity

Secure FTP Server extends the industry standard FTP with strong reliability features, including post transmission integrity verification, mid-file recovery, and automatic restart

Tracking and Auditing

Secure data delivery requires strong audit trails for tracking and non-repudiation. Secure FTP Server provides industry standard logging (W3C, NCSA, Microsoft IIS Extended), E-mail notification of completed transactions, and digital certificates for proof of identity.

Programmatic Interface

Secure FTP Server can be controlled through its Windows Administrator Interface, or through its Component Object Model (COM) interface. The COM API is a programmatic interface that lets you control the server from your own custom applications using any COM enabled programming language.

Accelerated Transfers

Secure FTP Server supports Multi-part (segmented) transfers for faster delivery of large files over large geographical distances. Multi-part transfers require the use of compatible clients such as CuteFTP Professional.

Life-Cycle Management

GlobalSCAPE Secure FTP Server lets you quickly and efficiently manage the removal of users, manage temporary accounts, address the revocation and if necessary re-issuance of expired or compromised public-keys or certificates.

Authentication and Authorization

Secure FTP Server supports password, public-key, or one-time-password authentication. User profiles can be managed internally or externally through NTLM, Active Directory (AD) or ODBC data sources.

User and Group Management

Manage system resources including bandwidth, folder access, file types and more using granular or site-wide controls provided for user and group management. Visually manage folder permissions via Explorer-like Virtual File System view. Inherit or override permissions, grant administrative, guest, or anonymous permissions or deny access altogether.

Support

Questions? Navigate to http://www.globalscape.com/support for information on customer service, technical support, software registration, product manuals, and downloads, as well as access to GlobalSCAPE's Knowledge Base and FAQs.

What's New in Secure FTP Server 3

HTTP/S Add On Module

Support transfers over HTTP and HTTPS. Now your users don't need a special client: they can transfers file securely with their browser. Secure FTP Server's HTTP/S Add On Module is downloaded and purchased separately from Secure FTP Server. For more information go to http://www.globalscape.com/gsftps.

Connection Monitoring

Troubleshoot problem connections with connection monitoring. Examine per-connection logs in real-time, search for text, filter results, and configure monitoring options.

Status Monitoring

View information and statistics on the selected connection, including the last three files downloaded, current directory, originating IP, and more.

Verbose Logging

Choose between standard and the new verbose logging format, which captures every client command and server response issued.

Streaming Compression

Accelerate transfers for clients supporting the new MODE Z command through compression of transfers on-the-fly.

Import/Export Configuration

Need to set up a load-balanced architecture or simply back up your existing configuration? Quickly export or import existing configuration files, including custom commands, event rules, and permission groups with a simple wizard.

File Locking

Secure Server now includes provisions for locking files during upload so that other clients won't accidentally download a partially uploaded file.

Max Concurrent Logins

Now you can limit the number of logins in addition to the amount of connections. For example, you can choose to let the user connect and then deny login, or disallow the connection socket altogether.

1

Install and Register

Install

To install Secure FTP Server

1. Start the installation wizard and follow the instructions.
2. Select the components you want to install. You can:
   - Install **Secure FTP Server** and the **Secure FTP Administrator** (Administrator Interface) together.
   - Install only the **Secure FTP Administrator** (Administrator Interface) for remote administration of **Secure FTP Server**.
3. Create a user name and password for the Administrator account for connecting to Secure FTP Server from the Secure FTP Administrator. Remember your user name and password; you need them to connect to Secure FTP Server.

Register

You must register Secure FTP Server with either a serial number or a trial serial number before you can use it.

Register Online (You must be connected to the Internet)

1. Start the Secure FTP Administrator
2. Enter your user name and password to connect to the Secure FTP Server.
3. If you are registering a trial use, select **Enter Trial Serial Number**.
4. If you are registering a purchased license, select **Enter Serial Number**.

Note:
If you are registering a purchased serial number, you can also select **Enter Secure FTP Serial Number...** from the Help menu.

5. Enter your serial number in the Serial Number field.
6. In the name field, enter your name and/or your company name.
7. Select **Next**. If the registration fails, choose from the following:
   - Retry online registration
   - Launch web registration form. This takes you to the GlobalSCAPE website where you can register.

- Email a registration request to GlobalSCAPE Support. A support representative will contact you with your registration information.

If you are behind a proxy, select **HTTP Proxy** and configure the proxy settings accordingly.

Note:
If a firewall or a proxy server is in use, your network administrator should ensure that port 80 is open during the registration process.

8. You should receive a message confirming registration. Click OK.

Note:
Registration must be performed through the Secure FTP Administrator on the server computer. You cannot register through a remote installation of the Secure FTP Administrator.

Note:
You can also email the manual registration information to GlobalSCAPE Technical Support. GlobalSCAPE will confirm your registration and send you the .reg file. You can send the email from any computer with Internet access; just remember to transfer the .reg file back to the computer you are installing the software on.

HTTPS Trial

If you are installing and registering a trial of Secure FTP Server's HTTPS trial, do the following:

To enable HTTP/S

1. From the Secure FTP Server menu, select Help > Enter HTTP/S Serial Number and follow the on-screen instructions.
2. Click the refresh button on the toolbar or press F5.
3. Select the **Server** tab.
4. Select the Site you want to configure from the left-hand navigation tree.
5. In the right pane, select the **Connection Options** tab.
6. Select **Allow HTTPS transfers to allow SSL connections over HTTPS**. Select the port for HTTPS. The default is 443.

Upgrading Secure FTP Server

If you are upgrading or updating **Secure FTP Sever**, use the following procedure. The upgrade or update process does not reset or otherwise affect your server configuration or user settings.

1. Download the most recent release of Secure FTP Server from http://www.globalscape.com/support/reg.asp and save it to your desktop.
2. Document the Administrator user name and password for the existing FTP server.

3. Stop the GlobalSCAPE Secure FTP Server service:
4. Back up the existing Secure FTP Server installation folder. The default installation path is: C:\Program Files\GlobalSCAPE\Secure FTP Server\. At a minimum, the following files should be saved:
    - C:\Program Files\GlobalSCAPE\Secure FTP Server*.aud

      (User database)
    - C:\Program Files\GlobalSCAPE\Secure FTP Server*.cfg

      (Site configuration and user permissions)
    - C:\Program Files\GlobalSCAPE\Secure FTP Server*.bak (Backup of .cfg file from previous session)
    - C:\Program Files\GlobalSCAPE\Secure FTP Server*.pvk (SSH key pair)
    - C:\Program Files\GlobalSCAPE\Secure FTP Server*.crt (Certificate)
    - C:\Program Files\GlobalSCAPE\Secure FTP Server*.key (Private keys)
    - Plus any other third-party certificate or key files you may be using.
5. Launch the Setup file downloaded in Step 1 above and select **Repair**, then click **Next** and follow the onscreen instructions.
6. When the upgrade or update is finished, start the GlobalSCAPE Secure FTP Server service.

If you need additional information or help, visit the Secure FTP Server Support Center.

2

Setting Windows System Services

Starting and stopping Secure FTP Server

Secure FTP Server starts automatically and runs as a Windows system service. If you close Secure FTP Administrator, the Secure FTP Server continues to run in the background as a system service.

To start or stop the Secure FTP Server with the Secure FTP Administrator

1. In Secure FTP Administrator, select **Service Applet Settings...** from the Edit Menu. The **Transfer Engine Service Settings** dialog appears.
2. Select **Start service** (or **Stop service**) and close the **Transfer Engine Service Settings** dialog box.

To start or stop the server using the Services option in the Control Panel

1. In Windows XP or 2000, select **Start**> **Settings**>**Control Panel**>**Administrative Tools**>**Services**.
2. Select **GlobalSCAPE Secure FTP Server** from the Services list.
3. Right-click or double-click and select **Start** (or **Stop**).
4. Close the Services and Administrative Tools windows.

To start or stop the server from the command line

1. Select **Start** > **Run**.
2. Enter **cmd** or **command**.

3. Select **OK**.
4. To start the Secure FTP Server, enter **Net start "globalscape Secure FTP server"** at the prompt. (include the quote marks).
5. To stop the Secure FTP Server, enter **Net stop "globalscape Secure FTP server"** at the prompt. (include the quote marks).
6. After the service is started or stopped enter **Exit**.

Note:
If **Install service** is the only button enabled in the Transfer Engine Service Settings window, select it, then select **Start service**.

WARNING:
Any time you run a server, you expose your computer to outside users. There is the potential for exposing files and programs on your computer and network to malicious outside users, particularly should the server become compromised.
Although you can set folder permissions from within the Server Administrator, you can add an extra level of protection by establishing a user account for the server and then limiting folder access through the server's user account permissions. This establishes a stopgap until server/system integrity can be restored should the server ever become compromised.

To configure the server to run securely

1. Create a user account for the server
2. Assign permissions to this user account
3. Assign the server to the account
4. Log the server on as a service
5. If necessary, configure the server's user account to map a virtual folder to a network drive.

Starting and stopping Secure FTP server remotely

To start or stop the Secure FTP Server remotely

1. In Secure FTP Administrator, select **Service Applet Settings** from the Edit Menu. The **Transfer Engine Service Settings** dialog appears.
2. Under Connection, select **Administer remote machine**. Enter the IP address of the server you want to administer.
3. Select **Connect to Service Manager**.

Note:
The remote Secure FTP Administrator you are logged on to passes your user name and password to the Windows System Services on the computer running the Secure FTP Server. The account you log on with must have

administrative rights on that server to make any changes to the GlobalSCAPE Secure FTP Server service running on it.

4. Select **Start service** (or **Stop service**) and close the Transfer Engine Service Settings dialog box.

Creating a user account for the server

In order to run **Secure FTP Server** securely as a service, you need to create a user account for it in Windows.

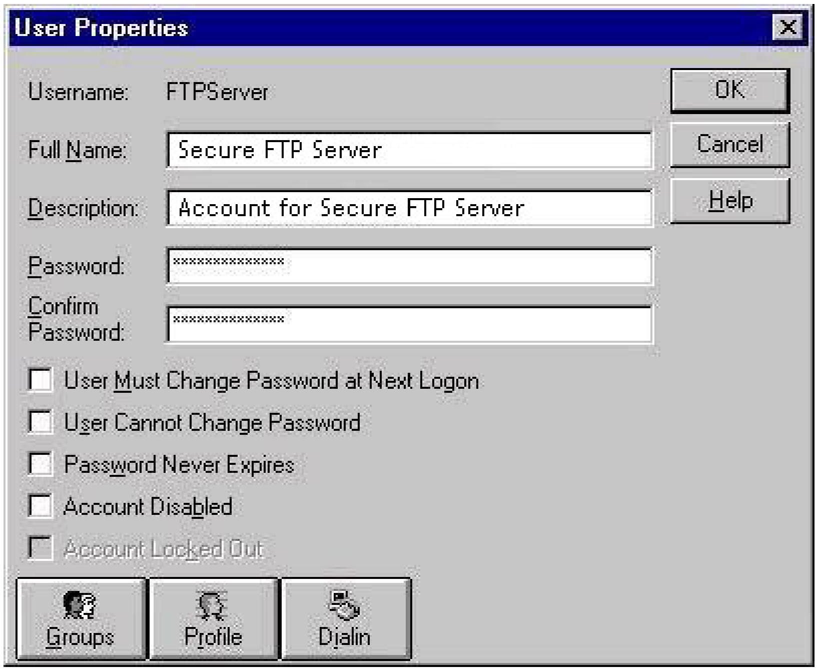
Note:
Setting up a user account increases security, but is not required to run Secure FTP Server.

To create a user account in Windows XP Professional or Windows 2000

1. After you install the server, navigate to **Start > Settings > Control Panel> Administrative Tools > Computer Management**.
2. Select **Local users and groups > Users**.
3. Select **Action> New User** to launch the New User dialog
4. Enter the appropriate information in the New User dialog for an Secure FTPServer user account.
5. Select **Create**.
6. Close the New User dialog box.
7. Close the Computer Management console.
8. From Administrative Tools, open **Local Security Policy**. If the Administrative Tools window is no longer open, navigate to **Start > Settings > Control Panel> Administrative Tools> Local Security Policy**.
9. Navigate to **Security settings > Local Policies > User Rights Assignment**
10. Double-click **Act as part of the operating system** under the **policy** column in the right-hand pane.
11. The **Local Security Policy Setting** dialog appears. Select **Add...**
12. The **Select Users or Groups** dialog appears. Select the new user you just added (**FTPServer**)**,** select **Add...** , then select **OK** twice to apply the change.
13. If necessary, Assign permissions for this user account in Windows.
14. Assign the server to the new user account and log the server on as a service.

To create a user account in Windows NT

1. After you install the server, navigate to **Control Panel > Administrative Tools > User Manager**
2. From the menu, select **File > New User** to create a new user account for "**FTPServer"**.
3. Enter appropriate information in all of the fields in the **User Properties** dialog box, as shown below.



Username: FTPServer

Full Name: Secure FTP Server

Description: Account for Secure FTP Server

Password: ***************

Confirm Password: ***************

☐ User Must Change Password at Next Logon

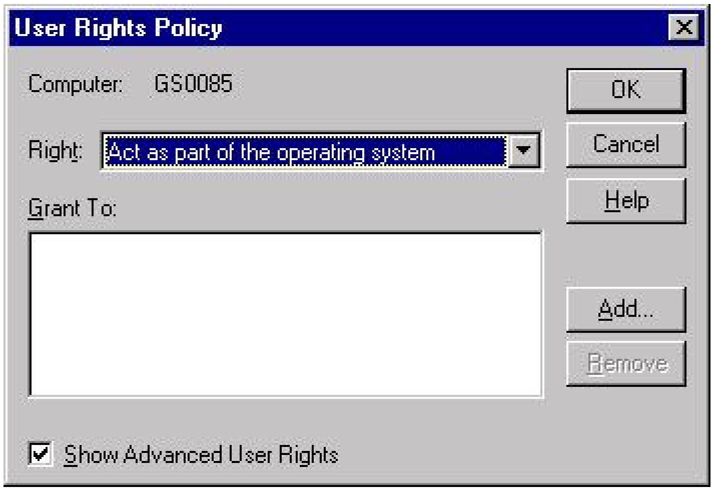
☐ User Cannot Change Password

☐ Password Never Expires

☐ Account Disabled

☐ Account Locked Out

4. Click **OK**.
5. From the menu bar, select **Policies > User Rights**. The **User Rights Policy** dialog will appear.
6. Enable the **Show Advanced User Rights** check box at the bottom of the dialog.
7. Select **Act as part of the operating system** from the drop-down box.



Computer: GS0085

Right: Act as part of the operating system

Grant To:

☑ Show Advanced User Rights

8. Click **Add**. The **Add Users and Groups** dialog will appear.
9. Make sure that the drop-down list at the top of this dialog has your own computer selected. Click the **Show Users** button and select **FTPServer** from the list
10. Click **Add**.
11. Click **OK** in both dialogs.
12. Assign permissions for this user account in Windows.
13. After assigning permissions, you should assign the server to the new user account you have created and then log the server on as a service.

Logging on the server as a service

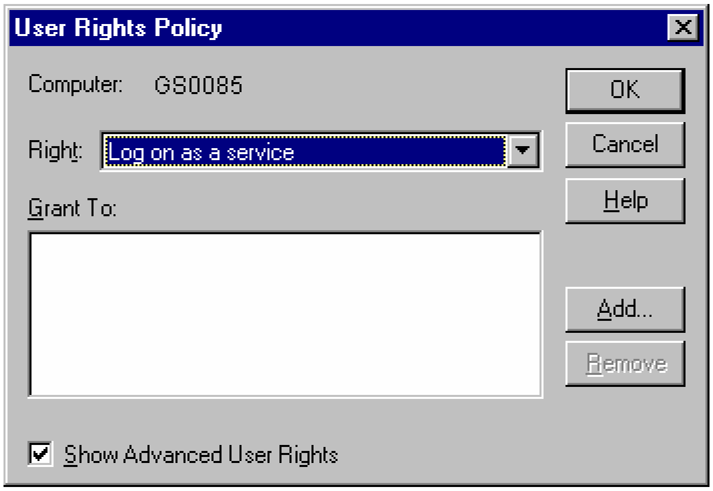
> **Note:**
> The logon as a service right is automatically granted in Windows XP Professional, 2003, and 2000.

Windows XP Professional, Windows 2003, and Windows 2000

1. Go to **Start > Settings > Control Panel> Administrative Tools > Local Security Policy**
2. Navigate to **Security settings > Local Policies > User Rights Assignment**
3. Double-click on **"logon as a service"** under the **policy** column on the right side of the window
4. Select **Add User or Group...** Select the user you want to add (**EFTServer**), select **Add...** , then select **OK** twice to apply the change.

Windows NT

1. Go to **Administrative Tools > User Manager** in your Windows NT operating system.
2. From the menu bar, select **Policies > User Rights**. The **User Rights Policy** dialog will appear.
3. Select the **Show Advanced User Rights** check box at the bottom of the dialog.
4. Select **Log on as a service** from the drop-down box.



User Rights Policy

Computer: GS0085

Right: Log on as a service

Grant To:

☑ Show Advanced User Rights

5. Click **Add**. The **Add Users and Groups** dialog will appear.
6. Make sure that the drop-down list at the top of this dialog has your own computer selected. Click **Show Users** and select **EFTServer** from the list
7. Click **Add**.
8. Click **OK** in both dialogs.

Windows NT permission rules

In order to secure your system, GlobalSCAPE recommends that you create an user account for the server and grant restrictive permissions to that user account. When you are assigning permissions to individual folders or directories in Windows NT, you may want to reference the following three rules. These rules differ somewhat from the VFS rules that govern **Secure FTP Server** permissions.

Three rules determine the permissions that are ultimately granted to a user in Windows NT:

1. **Explicit denial: All users or groups assigned "No Access" have no access**

   If the user, or a group that the user is in, has been assigned "No Access", that user is explicitly prohibited from using the file, folder or drive. No other permissions will change this.

2. **Cumulative permissions: Permissions are combined when a user is not explicitly denied access**

   If the user is not explicitly denied access, the user's permissions will be combined. For example, if user Cal is given read and write permissions for Folder1, and Cal is also in a group that is given execute permissions for that folder, then Cal will be able to read, write and execute files in Folder1.

3. **Implicit denial: A user or group that has never been granted any access at all will not be given access**

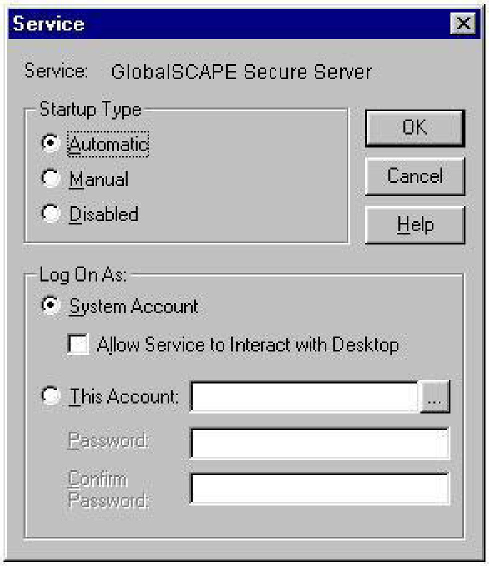
   If the user, or a group containing the user, is not granted any permissions, that user or group will be denied access. Access must be specifically granted.

Assigning the service to a NT user account

After you have installed the server, created an NT account for it and assigned permissions to the account, you need to edit the service itself so that it will not run as a “System Account” (this is the default account choice). Running the service as “System Account” poses the potential hazard of giving users complete access to your system.

To assign the service to an NT account

1. Click on the Windows **Start** button, select **Settings > Control Panel > Services**. *(W2K Control Panel > Administrative Tools > Services.)*
2. Select **EFTServer** from the list of services and double-click or press the **Startup** button. The **Service** dialog box, depicted below for Windows NT, will appear.
3. Below **Log On As**, change the service from a **System Account** to **This Account**.
4. Select the **EFTServer** user account you created previously.
5. Click **OK**.
6. You will need to restart the system in order for the change to take effect.



Service

Service: GlobalSCAPE Secure Server

Startup Type

- ● Automatic
- ○ Manual
- ○ Disabled

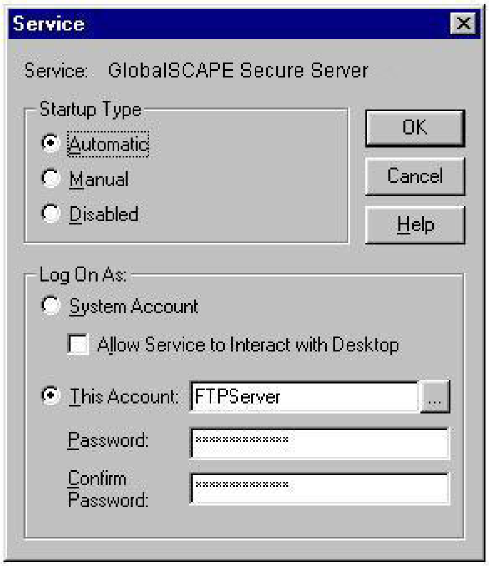
Log On As:

- ● System Account
  - □ Allow Service to Interact with Desktop
- ○ This Account:

Password:

Confirm Password:

Default



Service

Service: GlobalSCAPE Secure Server

Startup Type

- ● Automatic
- ○ Manual
- ○ Disabled

Log On As:

- ○ System Account
  - □ Allow Service to Interact with Desktop
- ● This Account: FTPServer

Password: **************

Confirm Password: **************

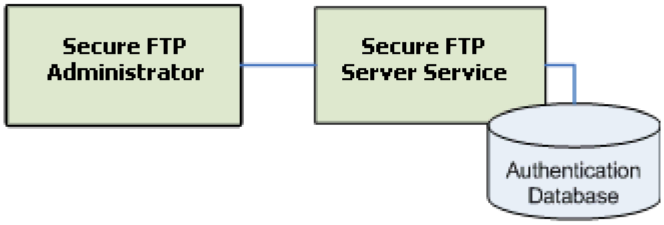
Recommended

3

Using the Secure FTP Administrator

The Secure FTP Administrator

Secure FTP Server is configured and maintained through the Secure FTP Administrator. Creating Server Groups, Servers, and Sites, managing user accounts and permissions, setting security protocols, setting up commands, configuring event rules: everything you do with Secure FTP Server can be managed through the Secure FTP Administrator. The Secure FTP Administrator connects to the server on either a local or remote computer. You can install the Secure FTP Administrator on as many computers as you like, but the Secure FTP Server may only be installed on computers with valid Secure FTP Server software licenses.



The Secure FTP Administrator opens when you select Secure FTP Server from the start menu or desktop. By default, the Secure FTP Server itself runs when the Windows OS boots up on the server. You must connect to an Secure FTP Server to make any changes to it.

Secure FTP Inheritance

Secure FTP Server employs an inheritance hierarchy to manage its server, site and user settings, and group permissions. The Secure FTP Administrator displays this hierarchy as a navigation tree in the left-hand pane (Server tab).

Server Groups are the topmost level. It contains Servers, Sites, and everything else beneath it. This is an organizational function for multiple groups of Servers and you can add an additional Server Group.

Servers represent one or more physical file transfer server (Server Engine) running on your local or remote system.

Multiple **Sites** (or hosts) are allowed within each Server. Sites are like virtual FTP servers bound to one or more IP address. Configuration of site-wide settings can be inherited at lower levels (at the User Setting Level or User levels).

User Setting Levels allow you to apply a setting configuration to an entire group of users. Setting Levels are a powerful way of organizing users into groupings with pre-defined settings. One Setting Level may by quite restrictive, while another may be quite liberal. Power users would be assigned to a level allowing greater flexibility in using server resources while guest users would be assigned to a more restrictive level, limiting access and use of server resources.

Users are individual clients assigned a Setting Level. Each user can be configured to inherit settings from the User Settings Level or have specific settings defined for that particular user.

Permission **Groups** allow the administrator to define access permissions to files and folders. Groups are assigned at the Site level. Users assigned to a Group have their access to folders and files defined by Group permissions. Group permissions are covered in more detail elsewhere in this document.

Commands and Event Rules - see Chapter 10, Automation.

Connecting to a server

To connect to a *local* server

1. Launch Secure FTP Administrator.
2. Select the server you want to administer. (You can manage multiple Secure FTP Servers with a single Secure FTP Administrator.)
3. On the menu bar, choose **File > Connect to Secure FTP Server.** The **Connect to Secure FTP Server** dialog box appears.
4. Enter your administrator **Username**.*
5. Enter your **Password**.*
6. Select **Local Host**.
7. Select **Connect**.

*The administrator username and password is created during installation.

> **Note:**
> If there is an error when trying to connect to Secure FTP Server, make sure that the Windows System Service for it is running.

To connect to a *remote* server

> **Note:**
> Before you can connect to a remote server, make sure you have the server configured for remote administration.

1. Launch Secure FTP Administrator.
2. In the left pane, select the server you want to connect to.
3. On the menu bar, choose **File > Connect to Secure FTP Server.** The **Connect to Secure FTP Server** dialog box appears.
4. Enter your administrator **Username** if it wasn't entered automatically.
5. Enter your **Password**.

6. Select **Remote Host**.
7. In **Host**, enter the IP address for the remote server.
8. In **Port**, enter the Port number for the remote server.
9. Select **Connect**.

Importing and exporting configurations

Import or export configuration files between Secure FTP servers. This is useful for load-balancing or for help with backing up configurations. You can also include user data, custom commands, and event rules you have configured.

To import configuration data

1. In Secure FTP Administrator, select **Configuration > Import/Export** from the menu. You should be connected to the server.
2. The Import/Export wizard appears. Choose **Import** and then select **Next**.
3. Enter the path to the file you want to import or choose browse and select the file from the file dialog.
4. Select **Finish**.

To export configuration data

1. In Secure FTP Administrator, select **Configuration > Import/Export** from the menu. You should be connected to the server.
2. The Import/Export wizard appears. Choose **Export** and then select **Next**.
3. Select from any or all of the following to export:
    - Configuration Data
    - User Data
    - Custom Commands
    - Event Rules
4. Enter the path to the folder you are saving the file to, or choose browse and select the location from the file dialog.
5. Select **Finish**.

Note:
Import/Export does not import or export SSL certificate data, SSH public keys, account passwords or Virtual File System (VFS) data. If you need to replicate a server for disaster recovery, see Copying server configurations.

4

Server Groups and Servers

Server Groups and Servers

Server Groups

Server groups are at the top of Secure FTP's setting hierarchy and allow you to group multiple servers. You can add as many Server Groups as you need.

Servers

Servers control the settings for one or more Secure FTP Servers, either locally or remotely. Servers consist of one or more physical file transfer server (Secure FTP Server) running on your local or remote system.

Create, delete, and rename Server Groups

To create a new Server Group

1. In Secure FTP Administrator, choose **File > Add New Group of Servers**. The **Create New Group** window appears.
2. In the **Group Name** box, type any name you want for the Server Group.
3. Select **OK**.

To rename a Server Group

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. In the left pane select the Server Group you want to rename.
3. On the menu bar, choose **Configuration > Rename**.
4. Next to the Server Group's icon, type any name you want for the Group.
5. Select **Enter** on your keyboard.

To delete a Server Group

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Server Group you want to delete.
3. On the menu bar, choose **File > Remove Group of Servers**.

Create a server

To create a new server

1. In the Secure FTP Administrator, select **File > Add New Secure FTP Server**. The **Add New Server** window appears.
2. In the **Name** box, type any name you want for the Server.
3. If the Server is on the same machine where you have opened the Administrator Interface, choose the **Local host** option and skip to step six. If the Server is on a different machine, choose the **Remote host** option and continue with step five.
4. In the **Host** box, type the IP address of the machine where the new server will be located.
5. In the **Port** box, leave the port at **1100** unless you want to use a different port to administer the server.
6. Select **Save**.

To delete a server

1. In the Secure FTP Administrator, select the Server you want to remove. You should be connected to the server.
2. On the menu bar, choose **File > Remove Server**. A warning window appears, reminding you that your log in information will be lost.
3. Select **Yes**.

WARNING:
When you delete a Server, you also delete all of your login information—you must manually recreate it.

Change global administration password

To change the administrator password and turn on or off server prompts, select **Edit > Server Global Settings**.

Prompt

Prompt for administrator interface login and password

If you want to enter your login information every time you connect the Administrator Interface to the server, leave this option checked. If you are within a secure environment, you might want to bypass the prompt to save time.

Change Administrator Password

Change your administrator password from this box.

User name

Your administrator username.

Password

Enter your new password.

Retype password

Enter your password a second time to confirm it.

Prompt on administrator exit

When exiting the Administrator Interface, you will be prompted and asked what to do with the FTP Server itself. You can choose to either leave it running or close the server, too. Typically, an administrator will choose to leave the server running (so that it can continue to service FTP requests). In this case, the administrator can choose the appropriate option and check the box labeled Don't show this prompt again.

Updating the user information from the authentication database

You can set Secure FTP Server to automatically check the user authentication database at regular intervals to make sure the server's user information is correct and up-to-date. This feature updates Secure FTP Server only. You must manually refresh user information in the Secure FTP Administrator in order to see changes on-screen.

To automatically update Transfer Engine authentication information

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Server you want to configure from the left-hand navigation tree.
3. In the right pane, click the **Server Options** tab.
4. In the **Default User Database Refresh Interval** list, select how often you want the Server Engine service to check for changes to the authentication database. If you do not want the service to check, select **Never refresh user list automatically**.

> **Note:**
> When you select **Refresh** in Secure FTP Administrator it only checks the Secure FTP Server service for updated user information. It does not check the authentication database.

Server log configuration

To monitor server activity, you can reference the server’s log files. Secure FTP Server supports W3C, Microsoft IIS and NCSA log file formats. Server events are logged to a file named [log file format]yymmdd.log. The log file format abbreviations used in the log file name are:

Log File Format	Abbreviation
W3C	Ex
Microsoft IIS	In
NCSA	Nc

To select a log file format

1. In Secure FTP Administrator select the **Server** tab. You should be connected to the server.
2. Select the **Server Options** tab.
3. From the **Log file format** list, choose **W3C**, or **Microsoft IIS** or **NCSA**.

> **Note:**
> Changing the log file format disconnects all active users. It is recommended to stop all Sites or wait until all users are inactive before changing the log file format.

4. Select **Apply**.

> **Note:**
> The W3C format records all times in GMT (Greenwich Mean Time).

To select the log file output folder

1. In Secure FTP Administrator select the **Server** tab. You should be connected to the server.
2. Select the **Server Options** tab.
3. In the **Folder to save log files** box, type the path for your server's log files. To browse for a path, select the yellow folder button.
4. Select **Apply**.

> **Note:**
> By default, log files are saved in the Secure FTP Server program folder.

To set the log type

1. In Secure FTP Administrator select the **Server** tab. You should be connected to the server.
2. Select the **Server Options** tab.
3. Select **Standard** or **Verbose** from the **Log type** pull-down menu.

Standard logging captures the following:

- user
- pass
- quit
- mkd
- stor
- ret
- dele
- abor (for applicable protocols)

Verbose logging captures all client commands and server replies, including the following:

- If login fails USER and PASS are logged.
- If USER command is logged with code 331 (password required).

- If PASS command is logged with the following error codes depending the type of failure:
  - TOO_MANY_CONNECTIONS_PER_SITE
  - TOO_MANY_CONNECTIONS_PER_USER
  - TOO_MANY_CONNECTIONS_PER_IP

  Code - 421, because this is temporary problem and client may retry

  - ACCOUNT_DISABLED
  - PROTOCOL_NOT_SUPPORTED
  - RESTRICTED_IP
  - PASSWORD_NOT_ACCEPTED
  - Code - 530, because this is permanent problem and client can't retry.

Other error codes:

- MAX_LOGIN_EXCEEDED - 421
- MAX_CONNECTTIONS_EXCEEDED - 421
- PROTOCOL_NOT_ALLOWED - 530

Note that connection errors such as banned IP are not logged.

To choose how often the log file is rotated

1. In Secure FTP Administrator select the **Server** tab. You should be connected to the server.
2. Select the **Server Options** tab.
3. Below the **Log file format** list, select one of the following: **Never**, **Daily**, **Weekly**, or **Monthly**.
4. Select **Apply**.

> **Note:**
> Logs are not written to disk in real-time. As events occur, the server buffers those events in real-time and then flushes (writes) them to file once either a) 60 lines are available, or b) 32kb of log data is received in 1 second or less.

Tweaking logging with the registry

You can adjust logging by changing values in the registry.

> **Note:**
> This is for advanced users only.

Registry location

HKLM\Software\GlobalSCAPE\CuteLogger

Values

LogBufferSize - DWORD

This value is the size of the [m_nBufferLen] member of CBaseLog.

The default is **255**.

QueueBufferSize - DWORD

This is the value of the [m_nQueueBufferMaxLen] member of CBaseLog.

The default is **32768**.

LogFlushTimer - DWORD

This is the value, in milliseconds, used by the QueueTimerProc to wait for flushing data to the disk.

The default is **60000** (1 minute).

Note:
Don't set LogFlushTimer to 0. It will max out the server CPU. The lowest setting you should use is 1.

Note:
Be sure to stop and restart the Secure FTP System Service after making any changes to the registry.

Remote administration

To connect to Secure FTP Server from a remote Secure FTP Administrator you must:

- Configure the Secure FTP Server. This must be done locally on the server.
- Configure the remote Secure FTP Administrator.
- Connect from the remote Secure FTP Administrator

Note:
To reconnect, start, or stop the Secure FTP Server service from a remote location, the remote computer must have a user account on the Secure FTP Server computer with the appropriate administrative privileges.

Configure Secure FTP Server for remote administration

1. Launch the Secure FTP Administrator on the Secure FTP Server computer and connect to the server you want to configure for remote administration.
2. Select the **Remote Administration** tab in the right-hand workspace.
3. Select an IP address from the **Administrator home IP** list. You can select a specific IP or all incoming IPs.
4. Select the **Administrator port**. 1100 is the default port.

5. Select the **Allow remote administration** check box. A warning appears advising you to connect over SSL for more secure administration.
6. Select **Yes** to set up secure administration, or **No** to administer over a clear connection.
7. Select **Apply.** If you chose to use SSL you must create or designate an SSL certificate to use for connections.
8. Close the Secure FTP Administrator. Make sure that the server service is still running (from the control panel service applet).

Configure the remote Secure FTP Administrator

1. Launch the Secure FTP Administrator on the remote computer.
2. Select the **Server** tab in the left-hand pane.
3. Select the Server Group you want to add the remote server to.
4. From the **File** menu, select **Add New FTP Server**. The **Add New Server** dialog appears.
5. Enter the name of the server you want to connect to.
6. Choose **Remote host**.
7. Enter the IP address of the Server in the **Host** field.
8. Enter the port number of the Secure FTP Server you are connecting to In the **Port** field.
9. Select **Save**.

Configuring secure remote administration

To configure secure remote administration, first configure the server to allow remote administration. Create or acquire an SSL certificate, and then consider whether you need implicit or explicit SSL.

Once engaged, SSL encrypts all of your remote administration sessions.

To enable SSL during remote administration

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right-hand pane, select the **Remote Administration** tab.
4. Select the **Use SSL for remote administration** check box.
5. Choose the location of the **Certificate file path** and the **Private key file path** with the browse button depicted by a folder.
7. Enter the **Private key passphrase**.
8. Select **Apply**.

Note:
If you do not already have a certificate and you are administering a local server, you can create a certificate using the **Certificate Creation Wizard** located on the menu bar under **Tools**.

You cannot use the **Certificate Creation Wizard** to create a certificate for a remote server. If you need to create a certificate for a remote machine, you must open the Secure FTP Administrator and use the **Certificate Creation Wizard** locally on that machine.

If you set up secure administration over an SSL connection, you will not be able to use the COM interface from remote machines.

Controlling access by IP address

By default, all IP addresses are granted access to the server. Secure FTP gives you two ways to limit which IP addresses can connect to your site:

- Grant access to only one specific IP address or a range of IP addresses.
- Deny access to one specific address or a range of addresses.

To grant access by IP Address

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **IP Access** tab.
4. Select the **Denied Access** radio button.
5. Click the **Add** button. The **IP Mask** dialog box will appear.
6. Enter the IP address or range of IP addresses that WILL have access to your FTP site. Secure FTP Server allows wildcards to select ranges of IP addresses.
7. Select **OK**. The **IP Mask** window disappears.
8. Select **Apply**.

To deny access by IP address

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **IP Access** tab.
4. Select the **Grant Access** radio button.
5. Select the **Add** button. The **IP Mask** window appears.

6. Enter the IP address or range of IP addresses that will *not* have access to your FTP site. **Secure FTP Server** allows wildcards to select IP address ranges.
7. Select **OK**. The **IP Mask** window closes.
8. Select **Apply**.

Configuring SMTP email notification

You can configure the server to send email alerts whenever certain events occur. You must provide **Secure FTP Server** with the address for an outgoing mail server, an address for the administrator, and other details.

To set up the server to send email notifications

1. Select the **Server tab** in Secure FTP Administrator and select a server the server you want to configure.
2. Select the **SMTP Configuration** tab from the right-hand workspace.
3. In **SMTP Server Address**, enter the address of the mail server the Secure FTP Server will use to send outgoing messages.
4. In **SMTP Server Port**, enter the port number where the mail server accepts messages. The standard is **25**.
5. If Secure FTP Server can connect to the mail server without a log in, leave the **Server requires authorization** check box clear and skip to step ten. If the mail server requires a user name and password from the Secure FTP Server computer, select the **Server requires authorization** check box and continue with step eight.
6. In the **Login** box, enter the user name needed to connect to the mail server.
7. In the **Password** box, enter the password needed to connect to the mail server.
8. In the **Name** box of the **Send Messages FROM** group, enter any name you would like for the "From Name" field.
9. In the **Address** box of the **Send Messages FROM** group, enter any address you would like for the "From Address" field.
10. In the **Name** box of the **Send Messages TO** group, enter the name of the server administrator, or any name you wish.
11. In the **Address** box of the **Send Messages TO** group, enter the email address of the person that should be notified of server events.
12. Select **Apply**.

Server statistics

To monitor current statistics at the server, site and user level

1. In Secure FTP Administrator, connect to the server and select the **Status** tab.
2. In the left pane, select the server, site, or connected user to view the related statistics.

Note:
After selecting a user in the left window, you can use the **Kick User** button below the right window in order to disconnect a user from a site. This does

not disable the user, but stops unacceptable activities while you reconfigure the user's access.

Server Statistics

- Server State
- Users Connected
- Active Downloads
- Active Uploads
- Download Speed
- Upload Speed
- Total Speed
- When the server was started
- The local time for the server
- Last Updated

Site statistics

Current statistics shown here include:

- Site Name
- Authentication Method
- Site Root Folder
- Site IP
- Site FTP Port Number
- SSL Enabled
- Site State
- Users Connected
- Active Downloads
- Active Uploads
- Download Speed
- Upload Speed
- Total Speed
- When the server was started
- The local time for the server
- Last Updated

User connection statistics

Current statistics shown here include:

- Login
- ID
- Connection Type
- When the connection was made
- Total time connected
- The IP
- The data type, such as ASCII

- The structure
- The transfer mode, such as stream
- The data connection, if applicable.
- The last three transfers
- The current working directory

Copying a server configuration to several computers

If you are installing Secure FTP Server on several different computers, and want to create a standard configuration for all machines, install the software on one machine to create a prototype configuration.

Installation and deployment considerations

- The prototype site **Administrator Home IP** must be set to **All Incoming**. It must not be bound to a specific IP address.
- Make sure the destination computers' installation paths are the same as the installation path on the prototype computer.

Deploy duplicate configurations

Set up the deployment configuration

1. Install and register Secure FTP Server.
2. Configure as desired. This includes passwords, sites, users, all site options, and all user options.

Note:
The configuration process also creates a specific VFS folder structure. Document this folder structure as needed to recreate it on the destination machine.

3. Stop the Secure FTP Server service.
4. Copy the following files from the **Secure FTP Server** program folder (perhaps in a zip file; the delivery method is up to you):
    - FTP.cfg
    - [YourSite].aud
    - FTP.BAK

Deploy the configuration to other servers/administrators

1. Create the same folder structure on the destination machine(s) as the folder structure created by the configuration of the prototype machine. The easiest way to do this is to simply copy the FTP folder structure from the prototype to the destination machines.
2. Install and register Secure FTP Server.
3. Cancel the automatic site setup wizard that appears the first time you run the Administrator Interface.
4. Stop the **Secure FTP Server** service and close the Administrator Interface.

5. Paste the files gathered from the prototype computer into the **Secure FTP Server** program folder, overwriting existing files as necessary (which should only be the FTP.cfg file at this point).
6. Restart Secure FTP Administrator. This starts the service.
7. Double-check server and site configuration, and make customizations as necessary. At this point, you should be fully set up on the destination computer.

Connection problems

If you are having problems connecting, check to make sure that:

- Your Username and Password are correct. These are case sensitive.
- The Host (the IP address) and Port are correct.
- The service is running.

> **Note:**
> If the service is not running, you may be able to start the service remotely by configuring Transfer Engine Service Settings located at Edit > Service Applet Settings.

- The network connection is functioning.

Server security considerations

Storing your login and password name may be convenient, but it is not secure. The password is available to anyone who uses the server machine.

Changing the administrator password and port is a good option if you do encounter a security breach.

> **Note:**
> It is so easy to configure the server that many administrators do not give careful consideration to administrative password or port changes. If you do not remember your password and port, you will not be able to connect to the server.

You may encounter port conflicts while attempting to run two sites with implicit SSL encryption. Keep this in mind as you configure multiple sites' encryption options.

Carefully consider inheritance as you begin to grant folder access to different VFS groups. Creating virtual folders gives users access to all subfolders of the folder you point towards.

Setting VFS access with patterns of inheritance derived from parent folders in a logical manner ensures that permission groups have predictable access to folders.

Connection monitoring

Secure Sever can monitor user connections in real time, and record activity to a log.

To monitor a user connection:

1. In Secure FTP Administrator, select the **Status** tab. You should be connected to the server.
2. Select the **Server > Site > user connection** you want to monitor.
3. Select the **Monitor User** button on the bottom toolbar. The connection activities display in the bottom right pane. You can set the number of lines the log records (**Log Scrollback**) and toggle automatic scrolling on or off (**Auto Scroll**).

5

Sites

Creating sites

Secure FTP Server allows you to create and run multiple sites through a single Secure FTP server. Each site must connect to a separate IP address or port or both. When you create a new site, the New Site Creation Wizard sets up the new site with FTP access enabled. Once you have finished the Site Creation process you can configure the protocols settings for the Site, such as FTP, FTP over SSL, and SFTP.

To create a new Site

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
1. Select the Server where you want the new site.
2. Choose **Configuration > Create New Site** from the menu. The **Create New Site** window appears.
3. Enter a **Name** for the site.
4. Choose a **Listening IP** address for the site from the drop-down box or select **All incoming**.
5. Choose a **Port**. The default port used for FTP connections is **21**, however, you can enter any value between 1 and 65,535. If you are setting up the site for Secure FTP Connections, you can later turn off plain FTP access in the **Connection Options** tab.

> **Note:**
> Assigning port numbers under 1024 may lead to conflicts with other programs running on your computer.

6. If you want the site to start immediately, select **Start site automatically after creation**.
7. Select the authentication method. The default method is GlobalSCAPE Secure FTP Server Authentication. If you need to use NT Authentication see Creating a site that uses NT authentication. For ODBC authentication, see Creating a site that uses ODBC authentication.
8. Select **Next**.
9. Enter a path to store the user database. Leave the default path unless you want to store the authentication database in a new location.
10. Select a polling option from the **User list refresh interval** pulldown menu. This selects how often the server checks the database for new users.
11. Select **Next**.
12. Enter a path to the root folder for the site.

13. Select **Create standard subfolders...** to automatically create **Bin**, **Pub**, **Usr** and **Incoming** folders with appropriate permissions under the root folder. This is only necessary if you are trying to mimic a typical default *nix Secure FTP server setup.
14. Select **Enable anonymous access to the server...** to create an anonymous account that does not require a password. The account will have limited permissions.
15. Select **Auto assign home folders to site users** to automatically create a user folder under \Site Root\Usr\[username] when a new user is added.
16. Select **Finish**. If the root folder has not already been created, you are prompted to do so: Select **Yes**. The folder is created and the **Create New Site** wizard closes.

Starting and stopping sites with the server running

To start Sites

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select **Go** from the toolbar. A pulldown menu appears.
3. Select the Site you want to start from the pulldown menu. To start all of them, select **All Sites**.

To stop Sites

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select **Stop** from the toolbar. A pulldown menu appears.
3. Select the Site you want to stop from the pulldown menu. To stop all of them, select **All Sites**.

Note:
If you stop a Site while users are connected, the users will be disconnected and file transfers may be interrupted.

Disconnecting problem users

Secure FTP Server employs the following methods to disconnect problem users:

- Blocking anti-timeout schemes
- Disconnecting after a defined number of invalid commands
- Disallowing the NOOP command

- Disabling an account after a defined number of incorrect login attempts
- Setting a maximum idle time limit

To block anti-timeout schemes

Many FTP clients send random commands such as REST 0, PWD, TYPE A, LIST, etc., to an FTP server to keep the session alive while the client is idle. **Secure FTP Server** can attempt to block these schemes.

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Site Options** tab.
4. Select the **Block anti-timeout schemes** check box.
5. Select **Apply**.

To disconnect users after a defined number of invalid commands

The server can automatically disconnect and even ban the IP addresses of users who send an excessive number of invalid commands to the server:

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Advanced** tab.
4. Select **Disconnect user after ___ consecutive invalid commands** and enter the number of invalid commands allowed before you disconnect the user. You may permanently ban the user's IP address from the site by selecting the **Ban IP address after excessive invalid commands** check box. You may later remove the ban on the user by removing their IP address from the list in the site's **IP Access** tab.
5. Select **Apply**.

To allow or disallow the NOOP command

Many FTP clients send a NOOP command to the server during idle times to keep the connection alive. You can choose whether or not to allow the NOOP command. If you disallow the NOOP command it will be considered an invalid command and treated according to your settings under **Disconnect after [Number of] invalid commands**.

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the User Setting Level you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Security** tab.
4. Select the **Allow NOOP command** check box to allow the NOOP command or clear the **Allow NOOP command** check box to treat the NOOP command as an invalid command.

> **Note:**
> If you are banning users who send excessive invalid commands and you are also treating NOOP as an invalid command then you will be banning users for sending the NOOP command. You may later remove the ban on the user

> by removing their IP address from the site's list in the **IP Access** tab. A gray check box in a user account indicates that the account is inheriting parameters from the User Setting Level.

5. Select **Apply**.

To disable an account after a defined number of incorrect login attempts

The server can automatically disable user accounts if users try to connect with the wrong password too many times.

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the User Setting Level you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Security** tab.
4. Select **Disable account after ___ incorrect password retries** and enter the maximum number of password retries you want to allow in the corresponding box. A gray check box in a User account indicates that the account is inheriting parameters from the User Setting Level.
5. Select **Apply**.

Enabling time out

You can automatically disconnect users after a specified time of inactivity.

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the User or User Setting Level you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Quota** tab.
4. In the right pane, click the tab and select the **Enable time out** check box. Enter the maximum allowable seconds of inactivity allowed before the user is disconnected.
5. Select **Apply**.

Flooding and denial of service prevention

You can configure the server to automatically ban IP addresses that may potentially be associated with a DoS (Denial of Service) attack. The server monitors connection patterns, tracks each user's activity density, and then bans IP addresses with unnaturally dense activity.

To activate Auto-ban

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **IP Access** tab.
4. Select a sensitivity level using the slider bar and ban period using the radio buttons based on the following:
    - **Ban IPs for time period proportional to sensitivity (higher = longer)**

If you select this option, IPs are banned temporarily. The server will restrict this IP's access to the server for a minute or two. The amount of time a user is banned from the site depends on the server security setting you selected using the slider bar. Choosing to ban users temporarily means that if the server makes a mistake and identifies an ordinary, but very active user as a threat, the user will soon be able to reconnect to the FTP site.

Banning an IP address temporarily protects the server from attacks. If the server is correct and a temporarily banned IP was the source of an attack, the server will not be harmed by the attempted attack. The server's resources will remain free or minimally burdened, instead of being completely bogged down by the attacking IP.

When you ban IP addresses temporarily, the level of security you set for the slider indicates both the number of seconds the user can attempt to occupy all of the server's resources before being banned and the number of seconds the user will be banned. The higher the security, the shorter the amount of time before the user is banned and the longer the user will remain banned.

- **Ban IPs permanently (Add to TCP/IP Access restrictions list)**

  If you elect to permanently ban the IP addresses of users whose activity fits the pattern of an attack, those users will be immediately banned as soon as they exceed the number of connections allowed for your security level. If the server has banned a user, you will need to modify the TCP/IP Access restrictions list to allow access.

5. Select **Apply**.

Modifying messages

Secure FTP Server can display messages to users in the following situations:

- Successful connection
- Login (under User settings)
- Maximum connections exceeded
- Exit

Connection Message

The connection message appears when a user first connects, but before a user logs on.

To modify the Connection message

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, click the **Site Messages** tab.
4. In the **Connect message** box, enter the text that you want to appear when the user connects.
5. Select **Apply**.

Login Message

Login messages may be applied at the User or User Setting Level. Users automatically inherit the message applied to their User Setting Level. You can optionally display a message unique to a User.

To modify the login message for a User or User Setting Level

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the User Settings you want to configure from the left-hand navigation tree.
3. Select the **Main** tab.
4. In the **Login message list,** choose an option.

- **Use Default**. You cannot add an additional message when a user connects. The default message for a successful login is:

```
230-Login OK. Proceed.
```

- **Add to Default**. This option places the default message on one line, then adds the message you typed into **Login message**. For users, the message set at the User Setting Level is the default.
- **Replace Default**. The server does not display the default message, but displays the message you type in to the Login message box. For users, the message is defaulted at the User Setting Level.
- **None**. No messages appear when a user logs in.

5. Select **Apply**.

Maximum Connections Message

You can configure a site to only allow a specified number of maximum simultaneous connections. If you choose this option, you can specify a message for users when the maximum simultaneous connections number is exceeded.

To Modify the Maximum Connections Message

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, click the **Site Messages** tab.
4. In **User limit message**, enter the message you wish to display if the maximum simultaneous connections number is exceeded.
5. Select **Apply**.

Exit Messages

The server can send an exit message when the client closes the session gracefully by using the FTP QUIT command.

To modify the exit message

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Site Messages** tab.
4. In the **Exit message** box, enter the exit message you wish to display.

5. Select **Apply**.

Site Options

Changing a site's root folder

The site root folder is specified when you create a new site. However you can later change a site’s root folder.

> **WARNING:**
> If you change a Site's root folder, all previously configured user and group folder permissions related to that site are deleted.

To change the Site root folder

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Site Options** tab.
4. In the **Site root folder** box, type the path or click the yellow folder button to choose a new Site root folder. This will be a physical folder.
5. Select **Apply**.

Creating a site that uses NT authentication

Secure FTP Server can create sites using the NT user authentication database so users can connect to the site with their NT user name and password. Permissions are assigned to users from the NT User Database on the domain of the system that is running the server. Secure FTP Server queries the Primary Domain Controller (PDC) for your domain and adds all domain users.

Users are listed as soon as you open the site you created using NT Authentication. You cannot add or change users from Secure FTP Server, but you can change their permissions, settings and status on the server.

> **Warning:**
> NT Authentication transmits passwords over the network without data encryption. To avoid exposing your passwords to possible theft, use SSL connections with NT Authentication.

To create a site

1. Follow steps 1-11 of Creating Sites.
2. In the **Authentication method** list, Choose **Windows NT Authentication**.
3. Click **Next**.
4. Click **Yes**. The **Authentication Options** window opens.
5. Choose **Active Directory (AD) Authentication**, or **NTLM Authentication** to match what is used on the server's domain.

6. In the **Domain Context** section, choose **Use default** if you want to use the authentication database from the machine's current domain, or choose **Custom**, and supply the domain name which has the authentication database you want.
7. In the **Allow access to the following group** section, choose **Everyone** to allow access to every user in the domain's database, or choose **Custom** and supply a group name for users that will have access to the server.
8. In the **User list refresh interval** list, select how often you want GlobalSCAPE Secure FTP Server to check the authentication database for new users.
9. Click **Next**.
10. Enter a path to the root folder for the site.
11. Select **Create standard subfolders...** to automatically create **Bin**, **Pub**, **Usr** and **Incoming** folders with appropriate permissions under the root folder. This is only necessary if you are trying to mimic a typical default *nix Secure FTP server setup.
12. Select **Enable anonymous access to the server...** to create an anonymous account that does not require a password. The account will have limited permissions.
13. Select **Auto assign home folders to site users** to automatically create a user folder under \Site Root\Usr\[username] when a new user is added.
14. Select **Finish**. If the root folder has not already been created, you are prompted to do so: Select **Yes**. The folder is created and the **Create New Site** wizard closes.

Creating a site that uses ODBC authentication

Secure FTP Server can create sites that use an ODBC database for authentication.

To create ODBC database authenticating site

1. Follow steps 1-11 under Creating Sites.
2. In the **Authentication method** list, choose **ODBC Authentication**.
3. Click **Next**. The **Authentication Options** window opens.
4. In the **Please specify user database data source** box, type a connection string for the ODBC database.
5. Select the **Encrypt passwords** check box to encrypt passwords stored in the database.
6. In the **User list refresh interval** list, select how often you want **Secure FTP Server** to check the database for new users.
7. Click **Next**.
8. Enter a path to the root folder for the site.
9. Select **Create standard subfolders...** to automatically create **Bin**, **Pub**, **Usr** and **Incoming** folders with appropriate permissions under the root folder. This is only necessary if you are trying to mimic a typical default *nix Secure FTP server setup.
10. Select **Enable anonymous access to the server...** to create an anonymous account that does not require a password. The account will have limited permissions.
11. Select **Auto assign home folders to site users** to automatically create a user folder under \Site Root\Usr\[username] when a new user is added.
12. Select **Finish**. If the root folder has not already been created, you are prompted to do so: Select **Yes**. The folder is created and the **Create New Site** wizard closes.

Specifying a PASV IP or PASV port range

If the Secure FTP Server is behind a firewall or NAT device, you may need to specify the Server's IP address or range of ports the server chooses from when issuing IP:PORT information to clients.

To specify a PASV connection through a range of ports

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Site Options** tab.
4. Select **Assign PASV mode IP Address**.
5. In the **IP** box, enter the server's IP address as should be seen by those outside your network.

Note:
Usually applies under SSL sessions when the NAT or FW device cannot see and therefore properly map the internal IP address of the server. Also applies if the NAT or FW device is misconfigured. It is recommend you first try connecting to the server with this field left as is.

6. In the **Port Range** boxes, enter the range of ports the server uses for PASV connections.

Note:
User primarily to limit the amount of ports used for the data connection portion of the FTP session, especially when the FW or NAT device was configured to only allow traffic on certain ports.

7. Select **Apply**.

Note:
If you specify a PASV mode port range you must open the same range of ports on your firewall.

Blocking site-to-site transfers

Although site-to-site transfers are great for the user, expediting what otherwise could be a slow transfer, many administrators consider site-to-site transfers a security risk, exposing servers to “port theft” or “FTPing by proxy” attacks. Depending on how your servers are configured, you may want to block these types of transfers.

To allow or not allow site-to-site transfers

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Site Options** tab.
4. Select or clear the **Block site-to-site transfers** check box.
5. Select **Apply**.

Blocking anti-timeout schemes

Enable blocking of anti-timeout schemes to defeat FTP clients that send a series of random commands to maintain an unattended connection to Secure FTP Server.

To automatically disconnect idle users using anti-timeout schemes

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Site Options** tab.
4. Select the **Block anti-timeout schemes** check box.
5. Select **Apply**.

Allow HTTP transfers

To enable HTTP transfers at the site level

1. In the SecureFTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Connection Options** tab.
4. Select **Allow HTTP Transfers on Port**. The default port number is 80.
5. Select **Apply** to save and implement the changes.

Note:
For a user to access SecureFTP Server using HTTP, Allow access using HTTP protocol must be selected at the user or user setting level.

Setting Transfer Protocol Security

SSL

Enabling FTPS, (SSL) at the site level

Secure FTP Server has robust SSL configurations that allow you to configure SSL connections on all sites, at the site level, at the user setting level, or at the user level. You can also configure SSL with a combination of these four levels. In order for SSL support to be available at any level it must first be configured at the site level.

To enable SSL at the site level

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Connection Options** tab.

4. Select **Enable FTP access** to allow both standard FTP connections and SSL connections. Clear **Enable FTP access** to allow only SSL connections to the site.

> **Note:**
> If you clear **Enable FTP access**, you must enable one or more of the other connection options or no one will be able to connect to the site.

5. Select:
   - **Enable explicit SSL connections**,
   - **Enable implicit SSL connections**,
   - Both.
   - Neither.

> **Note:**
> If Enable implicit SSL connections is selected you can change the Implicit SSL port. The default port is 990, which is normally used by FTP clients that support implicit SSL.

7. Select the **Certificate** and **Private Key** file paths. If you used the **Create SSL Certificate Wizard** and selected the **Set up Server to use the generated certificate** check box, then the **Certificate** and **Private Key** file paths will already be completed. Otherwise, choose the files using the browse buttons.
8. Enter the **Private Key Passphrase**. This is the passphrase that was used when the certificate was created. An incorrect passphrase generates errors when you select **Apply**.
9. Select **Require certificates from connecting clients**.

   If **Require certificates from connecting clients** is not selected, then clients that support SSL can connect to the server without supplying a certificate. If this box is selected, then FTP clients requesting an SSL connection must present a certificate before the server will allow them to connect. The client certificate must be in the Trusted Certificates database or signed by a certificate in the Trusted Certificates database. If the client has a certificate that does not meet those conditions, the connection is denied. However, its certificate is placed in the Pending Certificates database, where it can later be added to the Trusted Certificate Database. If the client does not present a certificate, the connection is denied.

Enabling HTTP/S at the site level

HTTP/S is an optional module that is purchased separately from GlobalSCAPE. In order for HTTP/S (HTTP over SSL) support to be available at any level it must first be configured at the site level. SSL can be configured at three levels: (1) at the site level (2) the user setting level, and (3) the user level.

To enable SSL at the site level

1. Start the Administrator Interface and connect to the server.
2. At the bottom of the left pane, click the **Server** tab.
3. In the left pane, expand a server group and server.
4. Select the site where you want to allow enable SSL support.
5. Click the **Connection Options** tab.
6. Enable HTTPS by placing a check in the box labeled: **Allow HTTPS Transfers on port**: (Port 443 is the default port for HTTPS.)

The use of SSL requires a valid certificate. Information about certificates is located on these pages:

- Creating certificates
- Selecting a certificate
- Signing a certificate
- Trusted certificates
- Importing a certificate into the Trusted Certificate Database
- Exporting a certificate from the Trusted Certificate Database
- Importing certificates from Microsoft IIS

7. Enter the file path for the Certificate and Private key.
8. Enter the Private key passphrase
9. Place a check in the **Require certificates from connected client**s box if increased security is needed.
10. Select **Apply** to activate your changes.

Disabling SSL connections

You can disable SSL support for every user on the server by disabling SSL support at the site level, or you can disable SSL for a specific user or User Setting Level.

To disable SSL connections for a site on the server

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.

3. In the right-hand pane select the **Connection Options** tab.
4. Select **Enable FTP access** and enter the port.
5. Clear BOTH **Enable explicit SSL connection**, and **Enable implicit SSL connection**.
6. Select **Apply.**

> **Note:**
> If SSL connections are disabled at the site level, they are also disabled for all User Setting Levels and users on the site.

To disable SSL connections for a user or User Setting Level

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the user or User Setting Level you want to configure from the left-hand navigation tree.
3. In the right-hand pane, select the **Security** tab.
4. Select **Allow access using FTP protocol**.
5. Toggle **Allow access using SSL over FTP protocol** until it is empty and white.
6. Select **Apply.**

Creating certificates

A certificate on the client must be associated with the server in order to initiate an SSL connection. When you are administering the server on the local machine, you may create certificates using the **Certificate Creation Wizard** (**Tools > Certificate Creation Wizard**) or import your own. There are three types of files associated with an SSL certificate key pair:

Private key file (.key) - The private key should never be distributed to anyone. It is used to decrypt the session which is encrypted by the public key.

Certificate request file (.csr) - Each time you create a certificate using GlobalSCAPE Secure FTP Server a Certificate request file is also created. This file can be signed by GlobalSCAPE Secure FTP Server's **Certificate Signing Utility** or sent to intermediate certificate authority such as GeoTrust for signing.

Certificate file (.crt) - This is a signed certificate, whether self-signed or signed by an intermediate certificate authority.

The private key (.key) and certificate request (.csr) files are created at the same time. You are prohibited from creating certificates for the Secure FTP Server while remotely administering the server because this action can create a security breach. Any certificates you create while remotely administering remain on the remote machine unless you take special steps to deliver and associate these files with the local machine.

To create an SSL certificate

1. On **Secure FTP Server**'s menu bar, choose **Tools > Certificate Creation Wizard**. The certificate wizard appears.
2. Enter the **Certificate Name**. Name the certificate that will be generated by the **Certificate Wizard**.
3. Enter the **Output Location** Enter the path or browse to the folder where the certificate is kept.

Note:
If you are purchasing a signed certificate from a certificate authority (CA), you usually need to open the certificate request file (.csr) and copy the contents to forward them to the CA. To do this, locate the .csr and open it with Notepad; then you can copy and paste the contents.

4. Choose the **Expiration Date**. Define how long the certificate remains valid.
5. Enter and confirm the **Passphrase**. Determine the passphrase that is used to encrypt the private key. The passphrase can be any combination of characters or spaces. Do not lose the passphrase. The certificate is useless without it.
6. Choose a **Key Length (in bits)**. Choose 512, 1024, 2048, or 4096 bit keys. Smaller keys are faster, larger keys are more secure.
7. Select **Next**.
8. Enter the **City/Town** where your organization is located.
9. Enter the **State/Province** where your organization is located.
10. Enter the name of your **Organization**.
11. Enter the **Common Name**. This is typically your name or the domain name associated with the site.
12. Enter a valid **E-Mail** address.
13. Enter a **Unit** name. Typically you enter a department or branch name.
14. Enter the two-digit **Country** code that identifies the country where your organization is located.
15. Select **Next**.
16. If **Use this certificate for Server authentication** is cleared, the wizard saves only the certificate files in the folder you previously specified. If selected, the wizard associates the certificate to the administration service or a site(s) you specify.

Note:
Associating a new certificate with a site requires a restart of the site, and any active users will be disconnected. We recommend that you associate certificates when sites are inactive or stopped.

17. If **Add this certificate to the Server Trusted Certificate list** is selected, the wizard adds the certificate to the Trusted Certificates database. Use this feature if you are creating certificates for user distribution. You can limit server access to include just the users that have the certificate. You can verify the addition to the Trusted Certificate Database by selecting **Tools > Certificate Manager**.
18. Use **the Apply certificate to** pull down menu to choose which components of the server are affected.
19. Select **Finish**.

Selecting a certificate

To assign a certificate you have created or obtained to a site

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.

2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Connection Options** tab.
4. Select **Enable explicit SSL connection** check box or the **Enable implicit SSL connection** check box or both.
5. Select the certificate file by clicking the browse button next to the **Certificate file path**.
6. Select the private key by clicking the browse button next to the **Private Key file path**.
7. Enter the **Private Key Passphrase**. The passphrase must match the passphrase that was used when creating the certificate.
8. Select **Apply**.

Signing a certificate

Secure FTP Server can sign certificate requests created by other clients. Typically, the client certificate request is signed with the certificate created for the server. If a certificate from the FTP server's Trusted Certificates database is used to sign client certificates, then all certificates you sign are automatically trusted.

To sign a certificate request

1. Obtain the Certificate Signing Request file (.csr). This can be done through email or any other file delivery method.
2. Choose **Tools > Certificate Signing Utility** from the menu. The **Sign Certificate Request** dialog appears.
3. **Client certificate request -** Click the folder to browse and select the Certificate Signing Request (.csr) file you want to sign.
4. **Output client certificate -** Browse and choose a folder in which to save the signed certificate (.crt) file.
5. **Server certificate -** Browse and choose the certificate you will sign with. This certificate must be in your trusted certificate database in order for clients submitting the signed certificate to connect to the site.
6. **Server private key** - Browse and select the private key file (.key) associated with the server certificate.
7. Enter the **Passphrase** associated with the server certificate.
8. Choose an **Expiration date**.
9. Select **OK**. The new certificate is saved in the folder you selected.
10. Return the certificate file (.crt) to the user.

Trusted certificates

If you require certificates from connecting clients before they can connect, then their certificate must be in the Trusted Certificates Database or signed by a certificate in the Trusted Certificate Database. To manage trusted certificates, select **Tools > Certificate Manager** from the menu.

Importing a certificate into the Trusted Certificate Database

1. On the menu bar, choose **Tools > Certificate Manager**. The **Certificate Manager** appears.
2. Select **Import** on the bottom of the **Trusted Certificates** window.
3. Browse to the folder that contains the client's certificate file and select the file.

Note:
Secure FTP Server can import a digital certificate from the following formats: PEM, Base64 Encoded X509, DER Encoded X509, PKCS#7, PKCS#12.
The Private Key associated with the digital certificate must be in one of the following formats: PEM, DER, PKCS#8, PKCS#12.

4. Select **Open**.
   - Secure FTP Server automatically detects the certificate format.
   - If Secure FTP Server is unable to determine the format, or if the import fails, you can manually convert a digital certificate to one of the above formats and import it. Consult the distributor/vendor of your certificate for details on this process.
5. The certificate is added to the Trusted Certificates database. Clients submitting that certificate are now able to connect to the server.

Exporting a certificate from the Trusted Certificate Database

1. Select **Tools > Certificate Manager** from the menu. The **Certificate Manager** window appears.
2. Select **Export**.
3. Browse to the folder where you want to save the certificate file.
4. Enter a name for the certificate file.
5. Select **Save**.

Importing certificates from Microsoft IIS 5

To use a certificate that you are using in IIS 5 you must:

- Add a Certificate Snap-in to your Microsoft Management Console,
- Export the certificate from IIS 5, then
- Import the certificate into Secure FTP Server.

Add the Certificate Snap-in

1. On the computer containing the certificate you want, select **Start**, then **Run**, and then type mmc to open the **Microsoft Management Console**.
2. On the Console menu, select **Add/Remove Snap-in**... from the console menu.
3. Select **Add**. The **Add Standalone Snap-in** dialog appears.
4. Select Certificates from the list and then select **Add**.

5. Select **Computer account**, then **Next**.
6. Select **Local computer**, then **Finished**.
7. **Close** the **Add Standalone Snap-in** dialog.
8. Select **OK** on the **Add/Remove Snap-in** dialog.

Export the certificate from IIS 5

1. Under the **Tree** tab in the **Microsoft Management Console** expand **Certificates**.
2. Select the **Personal** folder and then the certificate you want to export.
3. On the **Action** menu select **All Tasks>Export**...
4. Select **Next**.
5. Select **Yes**, export the private key, then select **Next**.
6. Select **Personal Information Exchange – PKCS #12 (.PFX)** and then select **Next**.
7. Enter the password you used when you created the certificate and select **Next**. This will create a .pfx file.

Import the certificate into Secure FTP Server

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Connection Options** tab.
4. Select **Enable explicit SSL access**, **Enable implicit SSL access**, or select both.
5. Select the yellow folder next to **Certificate file path** to browse and select the .pfx file you created in Export the certificate from IIS 5.
6. Select the yellow folder next to **Private key file path** to browse and select the .pfx file you created in Export the certificate from IIS 5.
7. Enter the password you used when you created your certificate in **Private key Passphrase**.
8. Select **Apply**. A message appears prompting a site restart.
9. Select **Stop** from the toolbar, stop the site, wait for the site to stop.
10. Select **Go** from the toolbar to restart the site.

SFTP

Enabling SFTP

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **SFTP Settings** tab.
4. Select **Enable SFTP (SSH2) access**. If you have not already done so, a dialog box prompts you to create a server key. Select **Yes**. The SFTP Settings tab appears.
5. Create a Site key pair.

To create a Site key pair

1. Select **Create** next to the site key pair field. A **Create SSH2 Public/Private Keypair** window appears.
2. Enter a name for the key pair and select the location to store it. Select **Finish**. Secure FTP Server generates and stores the key pair.

Select algorithms

1. In the **Use encryption algorithms** list, select any or all algorithms you want to allow for encrypting SFTP sessions. Hold down the **Shift** key on your keyboard to select a series, or hold down the **CTRL** key to select several that are non-contiguous.
2. In the **Use MAC algorithms** list, select any or all the algorithms to allow their use for message authentication. Hold down the **Shift** key on your keyboard to select a series, or hold down the **CTRL** key to select several that are non-contiguous.
3. Select **Apply**. A message appears telling you the site must be restarted for the changes to take effect.
4. Select **Yes**.
5. IF you want to change the SFTP port return to the Connection Options tab and specify the port number next to **Enable SFTP (SSH2) access on port.** 22 is the standard port for the SFTP protocol.

SFTP Transport layer settings

To select Message Authentication Codes (MAC)

Message Authentication Codes are algorithms used to confirm data has not been altered between the client and server.

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server, and SFTP should be enabled.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **SFTP Settings** tab.
4. In the **Use MAC algorithms** list, choose any or all of the four options:
   - hmac-md5
   - hmac-md5-96
   - hmac-sha1
   - hmac-sha1-96
5. Select **OK.** Secure FTP Server tries each selected MAC with the client until an algorithm is agreed upon.

SFTP algorithms

To select encryption algorithms (ciphers)

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server, and SFTP should be enabled.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **SFTP Settings** tab.
4. From the **Use encryption algorithms** list, select any or all encryption methods.

   Encryption algorithms

   - **ARCFOUR**: Arcfour is intended to be compatible with the RC4 cipher trademarked by RSA Data Security, makers of the famous OpenPGP program. It uses a 128-bit key and provides good security.
   - **cbc** - Cipher Block Chaining is an encryption technique used with block ciphers where the previous encrypted block is used as a basis for encrypting the next block, so that every block has to be in the correct order to be decrypted properly.
   - **CAST128**: This cipher is the CAST block cipher using 128 bit keys.
   - **Triple DES (3DES)**: This algorithm uses a 24-bit “triple key” to encrypt data 3 times. The 24-bit key is split into 3 8-bit segments and each is used for encryption. Triple DES is fast, but not as strong as the other algorithms.
   - **Blowfish**: The Blowfish algorithm is a public-domain block cipher method using a 128-bit key. Blowfish was intended to be a replacement for 3DES. It provides good security.
   - **Twofish**: Twofish is an improved version of Blowfish. It provides the strongest security available in GlobalSCAPE Secure FTP Server and should protect your data in most transfers. GlobalSCAPE Secure FTP Server recognizes Twofish encryption using 128 and 256 bit keys.

5. Select **OK**. Secure FTP Server tries each selected algorithm with the client until one is agreed upon.

Assigning a site's IP address and port

A site's IP address is initally specified when it is created.

To change the listening (incoming) IP address

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Site Options** tab.
4. In the **Home IP** list, select the IP address you want for the site.
5. Select **Apply**.

Setting the site port

Secure FTP Server allows you to define a listening port number and IP address for each site. The default for FTP sites is 21. You can enter any value between 1 and 65,535.

> **Warning:**
> Assigning a port number under 1024 may lead to conflicts with other programs running on your computer.

To change a Site's listening port

1. In Secure FTP Administrator, select the Server tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Connection Options** tab.
4. Enter the port number in the **FTP Port** box.
5. Select **Apply**.

Creating client key pairs for SFTP

When clients attempt to create an SFTP connection with your server, the server must send a key to the client verifying its identity. You can create the necessary key with Secure FTP Server. Use the same key for several sites, or create separate keys for individual sites.

To create a client key pair

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, click the **SFTP Settings** tab.
4. Select the **Enable SFTP** check box.
5. Select **Create**. A **Create SSH2 Public/Private Keypair** window appears.
6. In the **Enter path to store key pair**, enter the path where you want to keep the created key pair. Or select the browse button to navigate to the path where you want to store the key pair.
7. Select **Finish**. A note appears telling you the key pair was created successfully.
8. Select **OK**.

> **Note:**
> To use the key for other sites, rather than click Create, in the **Site key pair** box of the **SFTP Settings** tab enter or browse to the path where you stored the key.

Allowing SFTP password authentication

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.

2. Select the user you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Security** tab.
4. In the **SFTP Settings** section, clear the **Require client's public key** check box.
5. Select **Apply**.

> **Note:**
> If the check box appears grayed out, the user is inheriting the permission or requirement from his User Settings Level.

Advanced

Setting maximum transfer speeds

You can control a user's maximum transfer speeds at three levels:

- The site level
- The user setting level
- The user level

> **Note:**
> The Site level sets the limits of the User and User Setting Levels.

To configure maximum transfer speeds at the Site level

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Advanced** tab.
4. Select the **Max transfer speed (KB/s)** check box and enter the maximum transfer speed for the site. The server does not set a maximum transfer speed if the box is cleared.
5. Select **Apply**.

To configure maximum transfer speeds at the User and User Setting levels

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the User or User Setting Level you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Quota** tab.
4. Select the **Max transfer speed** check box and enter the maximum transfer speed (in Kilobytes per second) for the user.
5. Select **Apply**.

Setting maximum concurrent socket connections to a site

You can set the maximum number of connections to the Server at the site level. With multiple sites, this means that some sites can allow more users than other sites.

To restrict the number of socket connections

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Advanced** tab.
4. Select **Max concurrent socket connections** and enter the maximum number of connections you want to allow at any given time. If the box is cleared, the server does not restrict the number of users.
5. Select **Apply**.

> **Note:**
> The Max concurrent connection toggle limits the amount of socket, or low level, connections allowed by the server. When this limit is reached, any subsequent connection attempt generates a socket or network error in the client. It reacts as if the server is not even there. This occurs because the server refuses the connection entirely. For a server set up as an anonymous FTP server, it is recommended to limit connections on a per user basis. In this case, this will at least allow the user to partially connect before being told that the server is full or busy—a more graceful way of denying the connection.

Setting maximum concurrent logins

You can set the maximum number of connections to the Server at the site level. With multiple sites, this means that some sites can allow more users than other sites.

To restrict the number of user logins

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Advanced** tab.
4. Select **Max concurrent logins** and enter the maximum number of logins you want to allow to a user at any given time. If the box is cleared, the server does not restrict the number of users.
5. Select **Apply**.

Setting maximum connections per user

You can set the maximum number of simultaneous connections for a user at three levels;

- The site level
- The user setting level
- The user level

> **Note:**
> The site level provides a limit over all other levels (see Overview). For example, if the site level **Max connections per user** is 5, and a user's User level **Max connections per user** is set to 10, the user can still only connect to the server 5 times simultaneously.

To set maximum connections per user at the Site level

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Advanced** tab.
4. Select the **Max connections per user** check box and enter a number. If the box is clear a user can create an unlimited number of concurrent connections to the site (or according to the limits defined at the User or User Settings level).
5. Select **Apply**.

To set maximum connections per user account at the User and User Setting Level

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the User or User Setting Level you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Quota** tab.
4. Select the **Max connections per user** check box and enter the maximum times you wish that **User** account or users in that **User Setting Level** to be able to simultaneously connect to the site. Keep in mind that a gray check on a **User** account indicates the account is inheriting parameters from the **User Setting or Site level**.
5. Select **Apply**.

Banning unwanted file types

Secure FTP Server can block the upload or download of certain files. You can specify which files to block using wildcards or exact file names.

To ban files

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Advanced** tab.
4. Select **Exclude the following files from the site** and enter the filename or wildcard representation (*.mp3 or *.mp?) for the file(s) you want to exclude from the site. If you have more than one entry, separate each with a comma.
5. Select **Apply**.

6

Users and User Setting Levels

How user setting levels work

Every client account or user must be a member of a User Setting Level. User Setting Levels exist within a Site. User Setting Levels consist of a group of settings used as a template. Each new user is assigned to a User Setting Level where settings determine how server resources may be used. One Setting Level may be quite restrictive, while another may allow more access to resources. Power users would be assigned to a setting level allowing greater flexibility in using server resources while guest users would be assigned to a more restrictive level where use of server resources is very limited. User Setting Levels allow an administrator to make changes at the User Setting Level that affect all users within the level. The basic profile of individual users can also be changed (overriding the template). Users can also be moved between User Setting Levels with a simple drag-and-drop. Users that are moved inherit the properties of the new User Setting Level, but retain any modifications (overrides) made by the administrator.

The server ships with one User Setting Level named **Default Settings**. Additional User Setting Levels can be added to define access to server resources for various types of users.

> **Note**:
> User Setting levels apply to server resources. Permissions assigned for **Groups** control access to folders on your system.

To create new user setting levels

1. At the bottom of the left pane, select the **Server** tab.
2. Expand a Server Group, Server, and Site.
3. Select **User Setting Levels.**
4. Select **New**.

Inheritance

All Users settings initially share those of the User Setting Level where the account was created. When you view user properties, inherited settings are marked by gray check boxes.

You can change a User’s Setting Level by dragging and dropping them into a different level. The User's inherited settings change to reflect the settings of its new User Setting Level.

Overriding a user's inherited settings

You can override a User’s inherited settings. The check boxes toggle through three settings:

- **Inheirited:** A gray check box means no changes have been made by the administrator to the settings inherited from the User Setting level. This is a neutral indicator and simply means User Settings for that parameter are unchanged for that particular user.
- **Overridden, Enabled:** A black check box means the administrator has overridden this inherited setting. This setting is *enabled* for the user even though it was disabled in the User Setting Level for this example.
- **Overridden, Disabled:** A blank check box means the administrator has overridden this inherited option. This setting is *disabled* for the User, even though it is enabled in the User Setting Level.

Note:
If a User account contains modified (overridden) settings and is moved to a new User Setting level, those modifications remain in effect at the new User Setting Level.

Creating user setting levels

You can create one or more user setting levels before or after creating users and subsequently assign users to the desired user setting level. This allows you to control the server’s resources while still giving your users the flexibility they need to transfer essential files.

To create a new User Setting Level

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select **Configuration > Create New User Setting Level** from the menu. The **Create New User Setting Level** dialog appears.

3. Select a Site from the pull down menu.
4. Enter a name for the User Setting Level.
5. Optionally, enter a description for the user setting level.

6. Select **OK**. The new user setting level displays.
7. Enter desired options in the **Main** tab. See Disabling Users and User Setting levels and Specifying a user's home folder for more information.
8. Change the **Login message** by entering or changing text in the text box.
9. Select the appropriate action from the drop-down menu above the text box.
10. Select **Apply** to make the change.
11. Select the **Security** tab. For more information, see:
    - Security Options
        - Disconnecting problem users
        - Allowing users to change their passwords
        - Allowing users to verify file integrity
        - Restricting User to a Single IP Address
    - Protocol permissions:
        - FTP
        - FTPS, SSL and TLS
        - SFTP
12. Select the **Quota** tab. For more information, see:
    - Transfer Limits:
        - Setting maximum transfers per session
        - Setting maximum transfer size
    - Connection:
        - Enable Time Out
        - Set maximum transfer speeds
        - Setting maximum connections per IP
        - Setting maximum connections per User
    - Disks Quota:
        - Configure user disk quotas
13. Select **Apply** to make the changes.

Adding new users to a site

To add a new user to a site

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select **Configuration > Create New User** from the menu. The **New User Account Setup** dialog appears.
3. Select the Site you want to add a user to.
4. Enter the new user's **First Name** and **Last Name**. The server creates a **Username** in the format of [First_Initial_Last_Name]. You can optionally overwrite this.
5. Enter and confirm the **User Password**.
6. Select a **Password Type** from the pull down menu.
7. Optionally add a **Description**.
8. Select **Next**.
9. Make a selection from the **Place user in the following User Setting Level** pull down menu.
10. Select both **Create user home folder** and **Grant FULL permissions...** to create a user folder located in the site root folder and to give the user full permissions to that folder.
11. Select **Next.**
12. In the **Not a member of** pane of the **Setup user groups** section, select one or more Groups where you want the new User to be a member. By default, all new Users are members of the **All Users** group.
13. Select **Finish** to generate the new user.

User and User Setting Level Settings

Disabling users and user setting levels

To disable an user setting level

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the User Setting Level you want to disable from the left-hand navigation tree.
3. In the right pane, select the **Main** tab.
4. Clear **Enable this settings level**.
5. Select **Apply**. A red "X" appears in the right-hand navigation pane over the User Setting Level and any users that have not been enabled independently of the setting level.

To disable a user

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the User you want to disable from the left-hand navigation tree.
3. In the right pane, select the **Main** tab.
4. Clear **Enable this user account**.

5. Select **Apply**. A red "X" appears over the user icon in the right-hand navigation pane.

To disable a user on a specific date (account expiration)

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the User you want to set an expiration date for from the left-hand navigation tree.
3. In the right pane, select the **Main** tab.
4. Select **Expire this account after** and fill in the expiration date. Select the pull down menu to select a date from the pop-up calendar.
5. Select **Apply**. The User account is disabled on the specified date.

Enabling SSL (FTPS, HTTPS) at the user and user access level

Secure FTP Server has robust SSL configurations that allow you to configure SSL connections on all sites, at the site level, at the user setting level, or at the user level. You can also configure SSL with a combination of these four levels.

> **Notes:**
> In order for SSL support to be available at any other level it must first be configured at the Site level.
> HTTP/S is an optional module that is purchased separately from GlobalSCAPE.

To enable SSL at the user setting level

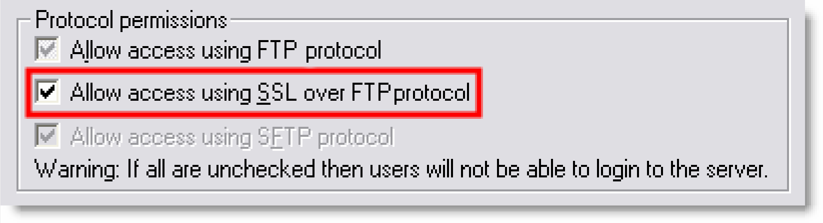
1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the User Setting Level you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Security** tab.
4. Select **Allow access using SSL over FTP protocol** to allow users in the access level to connect using SSL.
5. Either:
   - Clear **Allow access using FTP protocol** to allow only SSL connections, OR
   - Select **Allow access using FTP protocol** to allow both standard FTP and SSL connections.

To enable SSL at the user level

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the User you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Security** tab.
4. Toggle the **Allow access using SSL over FTP/HTTP protocol** check box until it's a black check in a white box to allow users to connect to the server using SSL.
5. Either:
   - Clear **Allow access using FTP protocol** to allow only SSL connections, OR

- Select **Allow access using FTP protocol** to allow both standard FTP and SSL connections.

Note:
Gray check boxes indicate that the user is inheriting that option from the user setting level it belongs to. See Inheritance for more information.



Protocol permissions

☑ Allow access using FTP protocol

☑ Allow access using SSL over FTPprotocol

☑ Allow access using SFTP protocol

Warning: If all are unchecked then users will not be able to login to the server.

Enabling HTTP access

To enable HTTP transfers at the user and user setting levels

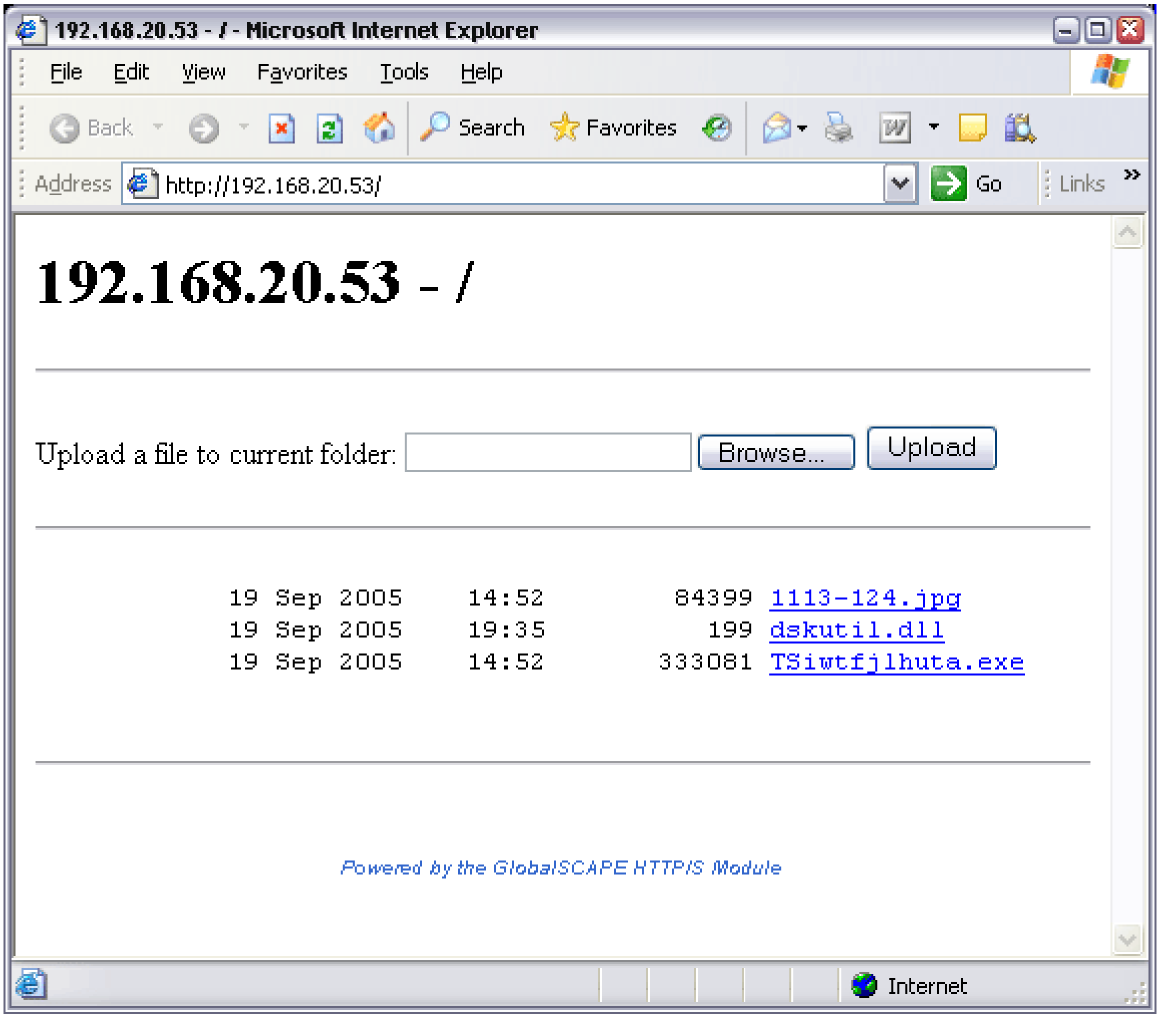
1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the user or user setting level you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Security** tab.
4. Under Protocol permissions, select **Allow access using HTTP Protocol**.
5. Select **Apply** to save and implement the changes.

Note:
HTTP transfers must be enabled at the site level to allow users HTTP access.

HTML Listing and Upload Form

When HTTP transfers are enabled, a user has permissions to upload using HTTP, and that user navigates with a browser to the specified address, the HTML Listing and Upload form appears. This form allows the user to upload and download files from Secure FTP Server. Typically the user is only given access to his or her home directory. Users can enter a direct path (UNC is supported if the OS the user is using also supports it) or they can select **Browse...** and locate the file with the browser's standard file dialog. Note that this upload form limits the user to uploading one file at a time.

Note:
This feature is available if you have the HTTP/S module.



192.168.20.53 - /

Upload a file to current folder:

```
19 Sep 2005    14:52       84399 1113-124.jpg
19 Sep 2005    19:35         199 dskutil.dll
19 Sep 2005    14:52      333081 TSiwtfjlhuta.exe
```

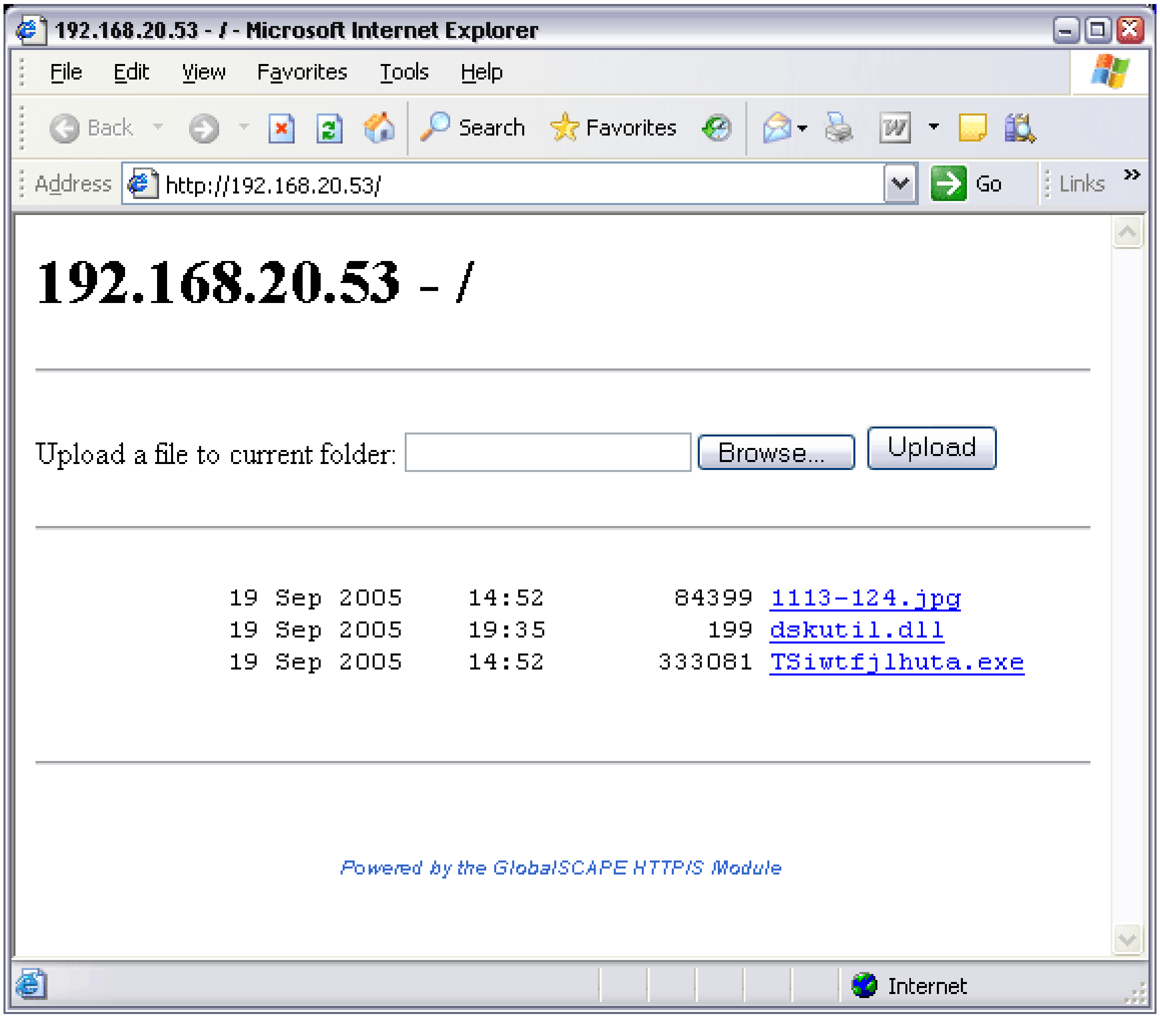
Powered by the GlobalSCAPE HTTP/S Module

HTML Listing and Upload Form

Note:
See also Customizing the HTML Listing and Upload Form upload appearance.

Customizing the appearance of the HTML Listing and Upload Form

The look and feel of the HTML Listing and Upload Form can be customized by modifying the cascading style sheet (CSS) that the form upload HTML calls. You can change options like background color (and/or image), fonts used (including specifying font sizes and styles, etc), link colors and decoration, and more.



192.168.20.53 - /

Upload a file to current folder:

```
19 Sep 2005    14:52        84399 1113-124.jpg
19 Sep 2005    19:35          199 dskutil.dll
19 Sep 2005    14:52       333081 TSiwtfjlhuta.exe
```

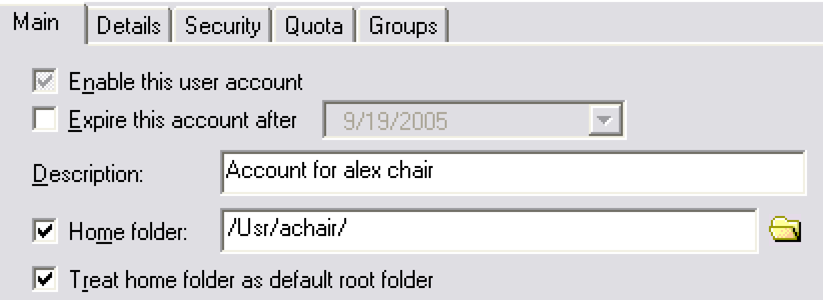
Powered by the GlobalSCAPE HTTP/S Module

HTML Listing and Upload Form, default appearance

To customize the HTML Listing and Upload Form

1. Create a CSS file using your favorite text editor and name it "htmllisting.css".
2. Place the file where the browser can find it. This depends on how the home directory is configured for that user.

a. If **Treat home folder as default** is selected for the user accessing the form upload, the home folder is set as the default root folder, and htmllisting.css should be placed in the folder designated as home folder.



☑ Enable this user account

☐ Expire this account after 9/19/2005

Description: Account for alex chair

☑ Home folder: /Usr/achair/

☑ Treat home folder as default root folder

Secure FTP Server User Account, Main Tab

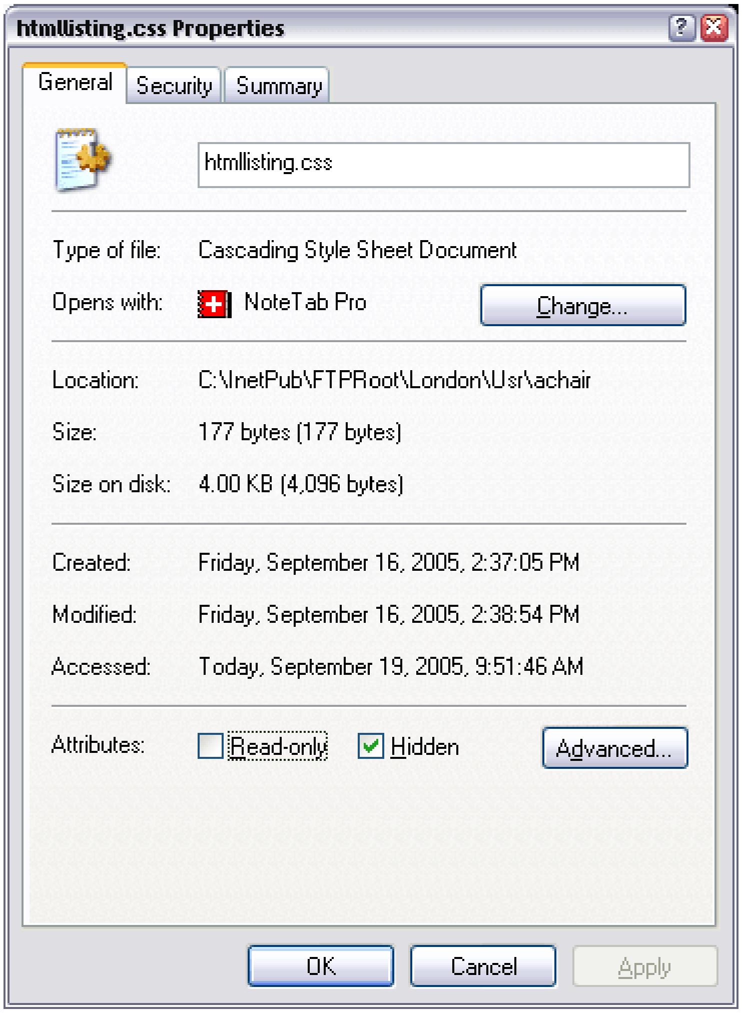
Note:
The user must have file download access in their home directory for the CSS to load in the browser.

Note:
The CSS file will display in the user's directory unless you hide the file. You can hide the file from view by setting the properties for that file in Windows.

b. If **Treat home folder as default** is not selected, htmllisting.css must be placed in the server's root directory, and the user must be granted download access to that directory. Note that if this is done, the user is dropped into the root level and not that user's home folder.

To hide the .css file from the user

1. From Windows, locate the user's home directory.
2. Right-click on htmllisting.css. The properties dialog appears.



htmllisting.css Properties

htmllisting.css

Type of file: Cascading Style Sheet Document

Opens with: NoteTab Pro

Location: C:\InetPub\FTPRoot\London\Usr\achair

Size: 177 bytes (177 bytes)

Size on disk: 4.00 KB (4,096 bytes)

Created: Friday, September 16, 2005, 2:37:05 PM

Modified: Friday, September 16, 2005, 2:38:54 PM

Accessed: Today, September 19, 2005, 9:51:46 AM

Attributes: □ Read-only ☑ Hidden

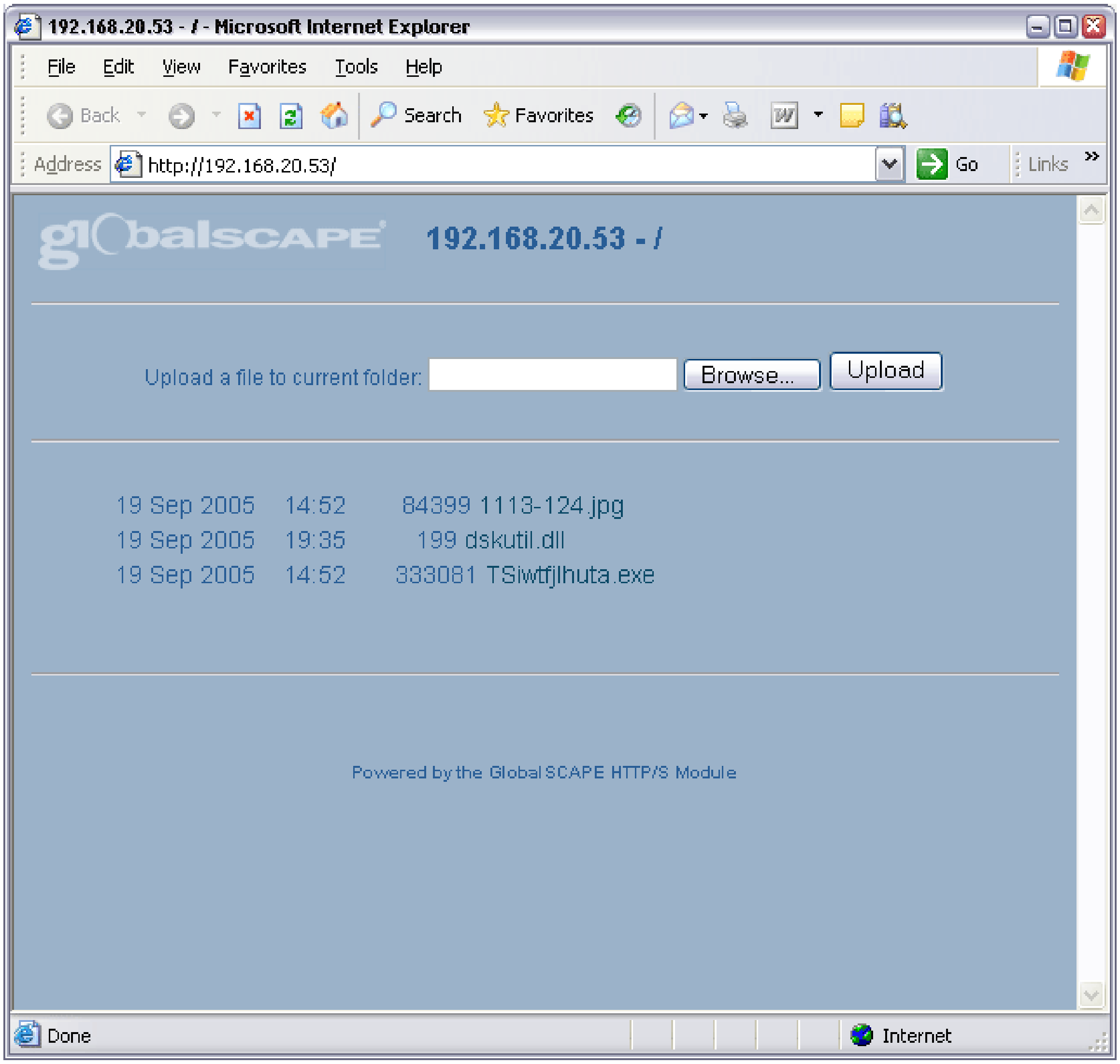
File Properties Dialog

3. Under **Attributes**, select **Hidden**.
4. Select OK.

Example

The following CSS changes the default appearance of the HTML Listing and Upload Form to that shown below:

```
BODY
{ background-color:#9bb2c9;
background-image:url(logo.gif);
background-repeat:no-repeat;
background-position: 14px 10px;}
H1
{ font:18px arial;
font-weight:bold;
line-height:20px;
color:#295d97;
text-align: center;}
PRE
{ font: 14px arial;
font-weight:normal;
line-height:20px;
color: #295d97;}
FORM
{ font:12px arial;
font-weight:normal;
line-height:20px;
color:#295d97;
text-align: center;}
EM
{ font:10px arial;
font-weight:bold;
line-height:20px;
color:#295d97;
text-align: center;}
A {color: #0a4966; text-decoration: none; }
A:HOVER {color : #ffffff; text-decoration: none;}
A:ACTIVE {color : #0066cc;}
```



http://192.168.20.53/

192.168.20.53 - /

Upload a file to current folder:

```
19 Sep 2005    14:52     84399 1113-124.jpg
19 Sep 2005    19:35       199 dskutil.dll
19 Sep 2005    14:52    333081 TSiwtfjlhuta.exe
```

Powered by the Global SCAPE HTTP/S Module

HTML Listing and Upload Form Customized with CSS

Restricting a user to a single IP address

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the user setting level you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Security** tab.
4. Select the **Restrict to this IP** check box and enter the IP address. Wildcards or ranges are not accepted.
5. Select **Apply.**

Specifying a user's home folder

You can determine the user's login folder at the user and User Setting Level. This is typically set at the user level.

To set a user's home folder

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the user you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Main** tab.
4. Select **Home folder**.
5. Select the browse button next to the **Home folder** box.
6. Select the folder you want the user to be placed in when they log on.
7. **Select Apply**.

> **Note:**
> Selecting **Treat home folder as default root folder** makes the home folder *the users* root folder.

To verify that the User Setting Level is not controlling the user's home folder

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the User Setting Level of the user you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Main** tab.
4. Clear **Home Dir**. Note that selecting this box forces the default root folder of all users in the setting level to a specified folder.
5. Select **Apply**.

Changing a user's password

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the user you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Main** tab.
4. Select **Change Password**. The Change User Password dialog appears.
5. Enter and confirm the password.
6. Choose the **Password Type** from the drop down list. You may choose:
   - **Standard** – A plain text password is required.
   - **Anonymous** – Any password, including nothing, allows an anonymous connection.
   - **Anonymous (Force Email)** – Any well formed email address is the password
7. Select **OK**. The Change User Password dialog closes.
8. Select **Apply**.

Accelerating transfers with Mode Z

Mode Z compression compresses files on the fly for file transfers, saving bandwidth and improving transfer times. The client must support MODE Z also to take advantage of this feature.

If MODE Z is enabled, Secure FTP Server will listen for MODE Z requests, then enable it for subsequent transfers from the client that requested it.

To allow a client to use Mode Z compression

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the user or user setting level you want to configure from the left-hand navigation tree.
3. Select the **Security** tab.
4. Select Allow **MODE Z Compression**.

Configuring user details

User details are the account specific details associated with the particular User, such as phone number, pager, and email address. Some of these fields (such as the email address) can be used in other parts of the program (such as the Event Rules) to notify the user of a completed transaction.

To configure User details

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the user you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Details** tab.
4. Fill out all necessary fields.
5. Select **Apply**.

Allowing users to change their passwords

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the user or User Setting Level you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Security** tab.
4. Select the **Allow PSWD command for client change password request** check box.
5. Select **Apply**.

To prohibit users from changing their passwords

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the user or Using Setting Level you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Security** tab.
4. Clear **User can change password (SITE PSWD command)**.
5. Select **Apply**.

Allowing users to verify file integrity

Although TCP/IP checks that all packets are received, malformed packets or other mishaps can occur, leading the FTP client to believe that a transfer was successful when it was not.

Secure FTP Server's file integrity command is defined as XCRC. Once an XCRC enabled client performs a transfer, it can request the server to do a checksum calculation on the file. If it matches the checksum on the client, then the transfer is deemed successful. Performing XCRC checksum calculations are processor intensive so enable or disable the feature accordingly.

To enable file integrity (XCRC) checking

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the user or User Setting Level you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Security** tab.
4. Select **Allow XCRC command** to enable XCRC file integrity checking.
5. Select **Apply**.

XCRC

XCRC is a proprietary command and is not defined nor endorsed by any FTP related RFC. Competing servers who wish to implement this command may do so using the following syntax: described below.

```
XCRC <File Name>
XCRC <File Name>, <EP>
XCRC <File Name>, <SP>, <EP>
```

SP = Starting Point in bytes (from where to start CRC calculating)

EP = Ending Point in bytes (where to stop CRC calculating)

FTP client log example

Client command:

```
COMMAND:>      XCRC "/Program Files/MSN Gaming
Zone/Windows/chkrzm.exe" 0 42575
```

- SP and EP are optional parameters. If not specified then it calculates the CRC for the whole file. If only EP is specified, then the CRC calculation starts from the beginning of the file to the EP.
- This command can be used for a single file at a time. It does not allow file lists as parameters.
- The standard CRC32 algorithm is used (for speed and efficiency).
- A client can invoke this command for uploads, downloads, single and Multi-Part transfers.

Server replies:

```
250 <XCRC>.
```

This returns the calculated CRC value.

```
450 Requested file action not taken.
```

This indicates that the file is busy.

```
550 Requested action not taken.
```

This indicates that the file is not found, has no read permission, or the SP or EP are not correct.

Setting maximum transfers per session

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the user or User Setting Level you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Quota** tab.
4. Select the **Uploads or Downloads per session** check box and enter a number. This number will be the maximum allowed during the user's session.
5. Select **Apply**.

Setting maximum transfer size

The maximum transfer size limits the user to a specified number of uploaded or download kilobytes per session. File Transfer Protocol does not send information to the server regarding the number of bytes that a user sends.

A user can start a transfer of virtually any size; however, once the limit is reached, the server will not transfer the rest of the file.

To set the maximum upload size

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the user or User Setting Level you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Quota** tab.
4. Select **Max Upload/Download Size** and enter the maximum amount of data (in kilobytes) the user may transfer during a session.
5. Select **Apply**.

Setting maximum connections per IP

You can set the maximum number of simultaneous connections emanating from a same IP address at three levels:

- The Site level
- The User Setting Level
- The User level

Note:
The Site level sets the limits of the User and User Setting Levels.

To set maximum connections from same IP at the site level

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Advanced** tab.
4. Select **Max connections from same IP** and enter a number. If the box is clear a user can create as many concurrent connections to the site from the same IP address that are allowed by the limits defined at the User or User Settings level.
5. Select **Apply**.

To set maximum connections per user account at the user and User Setting level

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the user or User Setting Level you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Quota** tab.
4. Select **Max connections from same IP** and enter the maximum times you wish that **User** account or users in that **User Setting level** to be able to simultaneously connect to the site from the same IP address. Note that a gray check on a **User** account indicates the account is inheriting parameters from the **User Setting or Site level**.
5. Select **Apply**.

Setting maximum connections per user

The maximum number of simultaneous connections for a User are set at three levels:

- The Site level
- The User Setting Level
- The User level

Note:
The Site level sets the limits of the User and User Setting Levels. For example, if the Site level **Max connections per user** is set to five, and a user's User level **Max connections per user** is set to ten, the user can have a maximum of five simultaneous connections.

To set maximum connections per user at the site level

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Advanced** tab.

4. Select the **Max connections per user** check box and enter a number. If the box is clear a user can create an unlimited number of concurrent connections to the site (or according to the limits defined at the User or User Settings level).
5. Select **Apply**.

To set maximum connections per user account at the user and User Setting Level

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the user or User Setting Level you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Quota** tab.
4. Select the **Max connections per user** check box and enter the maximum times you wish that **User** account or users in that **User Setting Level** to be able to simultaneously connect to the site. Keep in mind that a gray check on a **User** account indicates the account is inheriting parameters from the **User Setting or Site level**.
5. Select **Apply**.

Setting time-out

Setting a timeout value causes the server to disconnect a user after inactivity for the specified number of seconds.

To set Time Out function at the User and User Setting Level

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the user or User Setting Level you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Quota** tab.
4. Select **Enable Time Out (sec)** and enter the number of seconds you wish to limit that **User** account or users in that **User Setting Level** to be inactive. Keep in mind that a gray check on a **User** account indicates the account is inheriting parameters from the **User Setting or Site level**.
5. Select **Apply**.

Note:
Many popular FTP clients have keep-alive functionality that will attempt to issue do-nothing commands such as NOOP in order to simulate user activity and prevent a time-out. If **Block anti-timeout schemes** is enabled for the server, such do-nothing commands are ignored and will not reset the counter for the time-out limit.

Setting maximum transfer speeds

You can control a user's maximum transfer speeds at three levels:

- The site level

- The user setting level
- The user level

> **Note:**
> The Site level sets the limits of the User and User Setting Levels.

To configure maximum transfer speeds at the Site level

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Advanced** tab.
4. Select the **Max transfer speed (KB/s)** check box and enter the maximum transfer speed for the site. The server does not set a maximum transfer speed if the box is cleared.
5. Select **Apply**.

To configure maximum transfer speeds at the User and User Setting levels

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the User or User Setting Level you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Quota** tab.
4. Select the **Max transfer speed** check box and enter the maximum transfer speed (in Kilobytes per second) for the user.
5. Select **Apply**.

Configuring user disk quotas

Disk space management is an important aspect of server administration. Setting quotas allows you to specify the maximum amount of disk space available to each user in their home folder.

Max disk space sets the maximum disk space that users can consume in their home folders. The server administrator can assign each user a **Max disk space** for that user's home folder. As the user uploads and downloads, the server measures the user's **Used disk space**. Uploading files increases the **Used disk space** and deleting files decreases this number. If a user uploads too many files and the **Used disk space** equals the **Max disk space** that the server administrator assigned, the user has to delete files before uploading again.

When a server administrator uses Windows Explorer to add or delete files, the server will update file quotas appropriately. This means that a user's **Used disk space** will change when the administrator adds or deletes files in the user's home folder. Additions and deletions to a user's folder that are performed outside of server will only be monitored by the server and not prohibited, even when the number of files added exceeds the **Max disk space** allowed. In this situation, the user's file quota would be updated and the user would be prohibited from uploading until the **Used disk space** was less than the **Max disk space**.

To set a user's disk quota

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the User Setting level or user you want to configure from the left-hand navigation tree.
3. In the right pane, select the **Quota** tab.
4. Select **Enable disk quota for user's home folder** and enter the maximum number of kilobytes the user may use in their home folder.
5. Select **Apply**.

Multi-part transfers

Secure FTP Server supports multi-part transfers. The Secure FTP Server can accept multi-part uploads from advanced FTP clients such as CuteFTP Professional. The user must have appropriate privileges and be authorized to connect multiple times concurrently. The connecting client takes care of most details, including splitting the file apart, sending the multiple parts, and then requesting that the server to join them again upon receipt. The COMB command joins the parts back together. The benefits of segmented (multi-part) and concurrent delivery for accelerated transfers include:

- Accelerate throughput and maximize available bandwidth available to the client by allowing uploaded files to be split apart and transferred in multiple segments simultaneously.
- Command can be toggled on or off.

The COMB command is a proprietary command and is not defined nor endorsed by any FTP related RFC. However, he command can be integrated with other servers with the following syntax:

```
COMB <TF> <SF 1> … <SF n>
```

<TF> - path to target file, which will contain the combined data from the source parts.

<SF #> - source files (parts).

- Combine ***n*** source files (SF 1...***n***) into one file (TF).

- If the Target File already exists, then server appends source files to it.

- The server will delete all the source files once combined successfully.

- All File Names should be in quotes.

7

Permission Groups

Permission groups

Permission Groups set user access permissions to folders. Permission **Groups** are different from **user setting levels and users**. User setting levels control access to server resources such as bandwidth allowances and connectivity privileges.

Secure FTP Server creates three default permission groups for every site: Administrative, All Users, and Guests.

You can configure permission groups of your own or modify the settings for the default groups. Consider your security and access needs, then configure permission groups according to those needs. Add users to groups accordingly.

To view permission groups

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Expand a Server Group, Server, Site and **Groups**.

To add users to a permission group

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-
3. Expand a Server Group, Server, Site and **Groups**.
4. Select a permission Group.
5. In the right panel, click a User in the **Not a member of** window and click the left arrow. Users can be members of more than one Group at a time.

Note:
If a User is a member of more than one Group and you have not modified that User’s permissions, that User will have the highest level of access allowed to any folder or folder action. As the administrator, you can individually modify User permissions. The modified permissions outweigh all

Group permissions. For instance, a User is a member of three Groups that all have upload permissions to a particular folder, but you have denied that specific User permission to upload to the folder, then the User will be unable to upload to the folder.

Creating groups

You can create a permission group and add any users from the site to a group. You can then grant permission to folders by groups rather than granting permissions to each individual user.

To create a permission group

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Group you want to configure from the left-hand navigation tree.
3. Select **New** on the right-hand pane.
4. Select a site from the **Site** drop down list.
5. Enter a name for the Group in the **Group Name** box.
6. Select **OK**. The new group appears under the site selected in **Groups**.

Deleting groups

Deleting permission groups does not delete individual users.

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Group you want to configure from the left-hand navigation tree.
3. Right-click and choose **Delete** or select **Delete** from the toolbar.
4. Select **Yes** when asked **Remove group "[groupname]"?**

Adding or removing users

You can add any user on a site to any group on the same site. You cannot add users from one site to another site.

To move users into a group

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Group you want to configure from the left-hand navigation tree.
3. Select a User and use the arrows in the right panel to move the User to the desired membership.
4. Select **Apply**.

To move users out of a group

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Group you want to configure from the left-hand navigation tree.

3. Select a User and use the arrows in the right panel to remove the User to the desired membership.
4. Select **Apply**.

8

Virtual File System Permissions

Virtual File System (VFS)

The VFS lets you grant access to files and folders on your system. After selecting the left window's **VFS** tab, you can choose which files and folders will be available to users and then configure Group and User permissions for these folders.

Types of folders

The Virtual File System (VFS) allows you to create physical folders and virtual folders.

- **Physical folders** are folders you create on your hard drive from the server. They are simply called folders within the program.
- **Virtual folders** reference, or point to, currently existing folders on your computer or another system. Because a virtual folder name is only an alias for the real folder, when you create a virtual folder you do not have to give it the same name as the actual folder it references.

Permissions

You make the files, physical folders, and virtual folders available to users by granting permissions. To view the folders currently on the server, create new folders, or delete existing folders, select the **VFS** tab at the bottom of the left window.

VFS Permissions are constructed to allow users the least restrictive access to folders. For example, a user is a member of one group that has read, upload, download and delete permissions to a folder. Even if the user is a member of another group that has only download permissions to the same folder, the user will be able to read, upload, download and delete files from that folder.

VFS rules for folder access

The VFS access system is regulated by three rules. The rules differ from the rules that govern Windows NT permissions.

1. **User permissions are given priority.**

   If the server finds user specific permissions that are not those from groups in the folder that the user wants to access the server does not look for any group permissions. The server gives priority to individually configured permissions. For example, there is an individual user with the user name Bob. Bob is a member of two permission groups that have download and list permissions only for Folder1. However, you have decided that you want to give Bob full permissions for Folder1 without creating a new permission group, so you add Bob to the file and give him full permissions. Since the server looks for these individual user permissions first, then Bob will have full permissions for Folder1 no matter how his group membership is configured. This same rule also implies that if Bob has individual permissions that only allow him to download files from that particular folder, it does not matter if he is a member of two groups that have full permissions for the folder. Bob will only have permission to download files.

2. **If a user does not have individual permissions for a folder and is a member of more than one group, the server gives the user the least restrictive access for the folder.**

   From their groups, users receive all the permissions available for the folder. For example, suppose a user with the user name Jan is a member of two groups, Group1 and Group2, that both have permissions for a particular folder, Folder2. If Group1 has download permission and Group2 has upload permission then Jan will have both upload and download permissions for Folder2.

3. **The All Users group is the same as any other group except that it can't be removed from the root folder permissions list.**

   You can use the All Users group to determine inherited permissions from the parent folder in the Secure FTP Administrator. If you change any inherited permissions for the All Users group, the server display a screen to make sure you want to change the inherited permissions.

VFS permission inheritance

Any time a new folder is created it inherits permissions from its parent folder. Using permission inheritance, administrators can make global access changes by simply changing group access in a parent folder*.*

You can modify a folder's permissions even while it is inheriting permissions from a parent folder.

To modify a permission

1. Select a folder in the VFS structure.
2. Highlight an existing group or user or click **Add...** to add a User or Group to the selected folder.
3. Select the user or group you wish to modify permissions for.
4. Leave **Inherit permissions from parent folder** selected and then select any other additional permissions.

> Note:
> This will affect all sub-folders containing this User or Group who have the option **Inherit permissions from parent folder** turned on.

Disabling inheritance

In the course of server administration, you may want to reconfigure a folder’s settings. You can override a user’s inherited settings by clearing the **Inherit permissions from parent folder** check box.

If you manually clear **Inherit permissions from parent folder,** you can configure the folder's permissions the way you want them. If you later decide you want the folder to inherit permissions again, simply select **Inherit permissions from parent folder**.

The following instructions show you how to prevent a folder from inheriting its parent folder's permissions, force a single modified folder to begin inheriting permissions to sub-folders or reset all subfolders of a particular parent folder to inherit permissions from that parent.

To stop a folder from inheriting permissions

1. In Secure FTP Administrator, select the **VFS** tab. You should be connected to the server.
2. Select the folder you want to configure from the left-hand navigation tree.
3. In the right pane, clear **Inherit permissions from parent folder**.
4. Select from the following options:
    - **Copy** copies the permissions from the parent. You may later edit the permissions.
    - **Remove** removes all inherited permissions.
    - **Cancel** cancels the change.

To force a folder to inherit permissions from a parent folder

1. In Secure FTP Administrator, select the **VFS** tab. You should be connected to the server.
2. Select the folder you want to configure from the left-hand navigation tree.
3. In the right pane, select **Inherit permissions from parent folder**.
4. Select from the following options:
    - **Copy** copies the permissions from the parent. You may later edit the permissions.
    - **Remove** removes all inherited permissions.
    - **Cancel** cancels the change.

To reset folder permissions for all subfolders of a parent folder

1. In Secure FTP Administrator, select the **VFS** tab. You should be connected to the server.
2. Select the top level folder from the left-hand navigation tree.
3. Right-click the folder and select **Reset Subfolders.**

Note:
This deletes all existing permissions from the subfolders and forces them to inherit permissions from the selected folder.

Creating a new physical folder

1. In Secure FTP Administrator, select the **VFS** tab. You should be connected to the server.
2. From the left-hand navigation tree, right-click the folder you want ro create a subfolder in.
3. Choose **New Physical Folder...** from the menu.
4. Type a name for the new folder, and select **OK**.

Changing the name of a physical folder

You can change the name of a physical folder on the server but you cannot change the name of a virtual folder.

To rename a physical folder

1. In Secure FTP Administrator, select the **VFS** tab. You should be connected to the server.
2. Select the folder you want to configure from the left-hand navigation tree.
3. Choose **Rename Folder**.
4. Type the new name and press ENTER.

Deleting a physical folder

When you delete a physical folder from within the server, the folder is deleted from the Secure FTP server and your computer's hard drive.

1. In Secure FTP Administrator, select the **VFS** tab. You should be connected to the server.
2. Select the folder you want to delete from the left-hand navigation tree.
3. Choose **Delete Folder** from the menu.
4. Select **Yes** when asked **Are you sure you want to remove the folder "[foldername]"?**

Creating a new virtual folder

Virtual folders reference currently existing folders on your computer's hard drive. A virtual folder name is only an alias for the real folder. When you create a virtual folder you do not have to give it the same name as the actual folder it references.

1. In Secure FTP Administrator, select the **VFS** tab. You should be connected to the server.
2. Select where you want to add a virtual subfolder from the left-hand navigation tree.
3. Choose **New Virtual Folder...** from the menu.
4. Type a name for the folder in the **Alias** box.
5. Choose the target folder by typing the path in the **Target** box, or click the little yellow folder and browse to the target folder.
6. Select **OK**.

Deleting a virtual folder

When you delete a virtual folder, you merely delete a pointer, not the actual folder it references.

1. In Secure FTP Administrator, select the **VFS** tab. You should be connected to the server.
2. Select the folder you want to delete from the left-hand navigation tree.
3. Choose **Delete Folder** from the menu.
4. Select **Yes** when asked **Are you sure you want to remove the folder "[foldername]"?**

Setting folder permissions

You set permissions for physical and virtual folders the same way.

1. In Secure FTP Administrator, select the **VFS** tab. You should be connected to the server.
2. Select the folder you want to configure from the left-hand navigation tree.
3. Select the user or group you want to modify or click **Add...** to add a User or Group.
4. Select or clear the appropriate permission check boxes in the **File**, **Folder**, and **Contents** groups.
5. Select **Apply**.

Resetting folder permissions

Resetting folder permissions from a **parent folder** forces subfolders to exactly mirror those permissions. This simplifies the permissions status of these folders, making them more predictable

To reset folder permissions from a parent folder

1. In Secure FTP Administrator, select the **VFS** tab. You should be connected to the server.
2. Select the parent folder you want to configure from the left-hand navigation tree.
3. Right-click the folder and choose **Reset Subfolders**.
4. Select **OK**.

> **Note:**
> Resetting folder permissions from a parent folder differs from manually changing the inheritance values of subfolders because in a subfolder you have the option to either mirror the parent folder's permissions or to keep permissions for any new Users and Groups you have added while also mirroring the permissions for all Groups in the parent folder.

Mapping a virtual folder to a network drive

Factors to consider before mapping a folder to a network drive

If you want to map a virtual folder to a network drive:

- Are you the administrator of the computer where you are running the service? If not, you need to become the administrator of this computer, or have your system administrator make the changes.
- Are you in a domain or a workgroup?
- Do you understand how to use UNC path names? When you are remotely administering the server, you will need to enter UNC path names.

Necessary user accounts

In order to map to a network drive, establish the following server user accounts:

1. A separate account for the GlobalSCAPE Secure FTP Server service. It must have full access to any folder you want to make available on the server.
2. A **Secure FTP Server** administrator account. Your account on the computer where the server is running must have full access to any folder you want to make available on the server.

To map to a network drive in a domain

1. Through the Windows **Services** control panel, create and assign an NT account on the computer where the service is installed.

   Note:
   This should not be the default (system) account.

2. Assign restrictive file and folder permissions for this account.
3. In Secure FTP Administrator, create a virtual folder for a folder on your networked drive. If you are remotely administering, or the drive is not mapped to your computer, make sure that you use a UNC path name
4. Assign permissions for users by selecting the **VFS** tab within server, selecting the folder in question and then selecting or clearing the appropriate permission boxes.

   Note:
   You need to have administrative rights on the system the service is running on in order to create accounts.

To map to a network drive in a workgroup

1. Through the Windows **Services** control panel, create and assign an NT account on the computer where the GlobalSCAPE Secure FTP Server service is installed.

   Note:
   This should not be the default (system) account.

2. Assign restrictive file and folder permissions for this account.
3. Create a matching account on the target remote machine. Make certain it uses the SAME user name and password. Restrict permissions to this account to allow users access to only those folders they need.
4. In Secure FTP Administrator, create a virtual folder for a folder on your networked drive. If you are remotely administering, or the drive is not mapped to your computer, make sure that you use a UNC path name
5. Assign permissions for users by selecting the **VFS** tab, selecting the folder in question and then selecting or clearing the appropriate permission boxes.

   Note:
   You must have administrative rights on the system the service is running on in order to create accounts.

9

Authentication

Authentication types

Secure FTP Server supports three database types for authenticating users: Secure FTP Server, NT Authentication, and ODBC Authentication. Once a site has been configured through the **Create Site Wizard**, you cannot change the authentication method.

- **GlobalSCAPE Secure FTP Server Authentication** does not rely on outside sources for user information (accounts protected from the OS). All information is contained within the .aud file located in the server engine (cftpste.exe) folder. All information is encrypted and can only be modified through the Secure FTP Administrator.
- **ODBC Authentication** allows all users in an external ODBC database to have access to the server. See the topics under ODBC book for more information on configuring ODBC authentication.
- **NT (NTLM/AD) Authentication** Using this method, Secure FTP Server assigns permissions to users from the NT User Database on the system that is running the server. Secure FTP Server queries the Primary Domain Controller (PDC) for your domain and adds all domain users.

ODBC

Using an ODBC data source for user authentication

Secure FTP Server allows you to use any ODBC compatible database as a source for user authentication. You may add and remove users and set certain permissions using your existing database utility or through the Secure FTP Administrator.

In order to use an external ODBC data source you must:

- Create tables in an ODBC data source.
- Establish a System Data Source Name (DSN) in the ODBC Source administration tool.
- Set GlobalSCAPE Secure FTP Server to use the System DSN.
- Have Microsoft Data Access Components (MDAC) 2.6 or higher installed.

If you are using the server on Windows XP, you do not need to install MDAC 2.6 or higher on your computer. For any other Windows operating system you can download **MDAC** 2.6 or 2.7 from http://www.microsoft.com/data/download.htm

If you are using an Access database, you may also need to download a Jet driver. For more information about Jet drivers, see http://support.microsoft.com/default.aspx?scid=kb;en-us;Q282010.

For more information, see http://support.microsoft.com/default.aspx?scid=KB;EN-US;Q271908.

Creating tables for your ODBC data source

You must create two tables in the database for your data source:

The **ftpserver_users** table lists the user accounts and permissions groups in the site. A user account uses the information from all fields. A permissions group only uses the ID, Name, and Description fields and is used only for organizational purposes, not as a user login.

ftpserver_users: Table			
Field Name	Data Type	Field Size	Description
ID (Primary Key)	AutoNumber	Long Integer	User ID
Name	Text	50	Login name for this user
Password	Text	200	Password for this user
Description	Text	200	Description for this user
Type	Number	Integer	0=Group, 1=User
Password_Type	Number	Integer	Standard, OTP_MD4, OTP_MD5: Differentiates Regular vs. SKEY (OTP) password type. 0 = standard FTP password, 1=MD4 OTP, 2=MD5 OTP.
MD_Iter	Number	Long Integer	Current MDX iteration – used by OTP accounts only
OTP_Seed	Text	16	OTP Seed to be used for MDX Passwords – used by OTP accounts only.
Anonymous	Number	Long Integer	0=Normal Password, 1=Any password
Anonymous_Email	Number	Long Integer	0=Any anonymous password, 1=Email password required
Fullname	Text	200	User's full name
Email	Text	200	User's email address
Phone	Text	200	User's phone number
Pager	Text	200	User's pager number
Fax	Text	200	User's fax number
Comments	Text	200	User comments
Enabled	Number	Integer	0=Account disabled, 1=Account enabled
HomeDirectory	Text	512	**Secure FTP Server use only**

Note:
HomeDirectory must be created for ODBC authentication to work properly with Secure FTP Server, but you cannot use it for user account directories.

The **ftpserver_ids** organizes users into "groups" of permission levels. For each permissions group to which a user belongs there should be one entry in the table below.

ftpserver_ids: Table			
Field Name	Data Type	Field Size	Description
ID	AutoNumber	Long Integer	Unique ID for the record (key field).
User_ID	Number	Long Integer	This value refers to a user record in the ftpserver_users table. A corresponding (where ftpserver_ids.User_ID = ftpserver_users.ID) ftpserver_users record must exist with Type = 1.
Group_ID	Number	Long Integer	This value refers to the user setting level that the User_ID user record belongs to. A corresponding (where ftpserver_ids.Group_ID == ftpserver_users.ID) ftpserver_users record must exist with Type = 0.

Establishing a system data source name (DSN)

After you have created your database, you must associate it to your system.

To establish a system DSN

1. In Windows, open your **Control Panel**.
2. Select the **Data Sources (ODBC)** administrative tool.
3. Select the **System DSN** tab.
4. Select **Add**.
5. Select the appropriate driver from the list. There is a sample Microsoft Access (**GSFTPS.mdb**) database included within the installation folder.
6. Select **Finish**.
7. Enter the **Data Source Name** and **Description**. The default **DSN** is **GSFTP Server**.
8. Select **Select** and choose the database file you created when following the steps described in Create Tables for your ODBC data source or the supplied database described in Step five.
9. Select **OK**.

Using a DSN-less connection with ODBC authentication

You can create a Site with a DSN-less connection to your authentication database. If you have several simultaneous database connections, a DSN-less connection may be slightly faster than a DSN connection.

To create a site with a DSN-less connection

1. Open **Secure FTP Serve**r.
2. In Secure FTP Administrator, select **Configuration > Create New Site** from the menu.
3. Give the site a name, choose the IP address and Port.
4. In the **Authentication method** list, choose **ODBC Authentication**.
5. Click **Advanced**. The **Authentication Provider Options** window appears.
6. Enter the connection string in the box and click **OK**.
7. Click **Next** to continue with your site creation.

To create the string for a DSN-less connection

You must know the correct driver to use with your database. Create a connection string and enter it into Secure FTP Server. The connection string includes the name of the driver you need for your database, the location of your database, the name of your database, and, if necessary, a user name and password to access the database.

For local databases the connection string must include

- Provider [Provider=]
- Driver [DRIVER=]
- Database path and name, including the file extension [Dbq=]
- Username [Uid] and Password [Pwd] are required only if the database is password protected

For remote databases your connection string must include

- Driver [DRIVER=]
- Server [SERVER]
- Database [DATABASE]
- Username [UID]
- Password [PWD]

Examples

If you are pointing to Access 2000 database on the local machine named **Example** that was in the **xyz** sub-folder of your **c** drive the connection string is:

> **Provider=MSDASQL;Driver={Microsoft Access Driver(*.mdb)}Dbq=c:/xyz/Example.mdb;Uid=;Pwd=;**

If you have a remote MYSQL database named **Example** your connection string is:

> **Provider=MSDASQL;DRIVER={MySQL ODBC 3.51 Driver};SERVER=10.10.10.1;DATABASE=Example;UID=myusername;PWD=my password;**

> **Note:**
> Do not put any line breaks in your connection strings.
> You must have MDAC version 2.7 or higher to use a DSN-less connection.

NT Permissions

NT permissions

After it is installed, Secure FTP Server has access to local folders and files. To run it as a service with permissions to the network and mapped drives; however, you must create an NT account for the server, assign the server service to the account, and log the server on as a service.

Setting permissions for Secure FTP Server user accounts in Windows NT

Using Windows NT's permissions, set the permissions for files or drives of this user to be as restrictive as possible, while still allowing the server to run. After carefully determining what files and network folders your users will need to access, gradually increase the permissions.

> **Note:**
> Using NT Authentication, users permissions override the server's permissions. For example, if the server has read-only access to folder1, but user John Doe has read and write permissions to folder1, John Doe has those same permissions when he accesses folder1 through Secure FTP Server.

Windows NT permissions can be edited through the **Security** tab in the **Properties** of an object. On the **Security** tab, select **Permissions** to display and edit the permissions for the object. The appearance of this window is slightly different for files and directories, but in both cases the following permissions can be granted to users or groups:

- R (Read)
- W (Write)
- D (Delete)
- P (Edit permissions)
- O (Take ownership)

Keep in mind that you have the option to grant or withhold read and write permissions. Read-only permissions are the most secure. They allow users to access a file, but not to change it. For example, most users will need limited read access to the Windows folders (C, WinNT). However, most FTP Servers will not need ***any*** access to these directories at all.

In addition to the individual permissions, Windows NT also provides access levels that are simply pre-built sets of the existing permissions. Typically, you assign an access level to a user rather than granting individual permissions. One such access level is called "No Access." It does not contain any permissions.

10

Protocols and Security

Protocols and security

Secure FTP Server supports the following protocols:

- FTP
- SSL/TLS
- FTPS
- SSH
- SFTP
- HTTP
- HTTPS

FTP

The FTP protocol is an interactive file transfer mechanism that enables file transfers between Internet sites, or, more specifically, between two systems. It was created for transferring files independently of the operating system used, for example between a Macintosh and Windows PC. FTP's more notable features include handling for specific error situations and ensuring that a file sent from point A to point B will get there reliably.

FTP and Security

The FTP protocol specification (RFC 959) was published years ago when security was not a priority issue. As security became a concern, secure mechanisms such as SSL and TLS were adapted to help protect the FTP session from being intercepted or exploited. GlobalSCAPE's Secure FTP Server provides security through the use of FTPS (using SSL/TLS).

About SSL and TLS

Secure Socket Layer (SSL) is a protocol for encrypting and decrypting data across a secure connection from a client to a server with SSL capabilities. The server is responsible for sending the client a certificate and a public key for encryption. If the client trusts the server's certificate, an SSL connection can be established. All data passing from one side to the other will be encrypted. Only the client and the server will be able to decrypt the data. You can get a clearer idea of how SSL works by examining the representation of an explicit SSL transfer below.

GlobalSCAPE **Secure FTP Server** supports SSL for client and server authentication, message integrity, and confidentiality. You can configure GlobalSCAPE **Secure FTP Server**'s security features to verify users' identities, allows users to verify your identity and to encrypt file transfers. The key to understanding how SSL works is to understand the elements that take part in the process.

Elements that work together to establish a secure SSL connection

Client: The client needs to be an FTP client with SSL capabilities.

Certificate: Certificates are digital identification documents that allow both servers and clients to authenticate each other. A certificate file has a .crt extension. Server certificates contain information about your company and the organization that issued the certificate (such as Verisign or Thawte) while client certificates contain information about the user and the organization that signed the certificate. You can choose to either trust or distrust a certificate. In some cases, the client's certificate must be signed by the server's certificate in order to open an SSL connection.

Session Key: The client and the server use the session key to encrypt data. It is created by the client via the server’s public key.

Public Key: The client encrypts a session key with the server’s public key. It does not exist as a file, but is produced when a certificate and private key are created.

Private Key: The server's private key decrypts the client's session. The private key has a .key extension and is part of the public-private key pair.

Certificate Signing Request: A certificate signing request is generated each time a certificate is created. A certificate signing request has a .csr extension. This file is used when you need to have your certificate signed. Once the Certificate Signing Request file is signed, a new certificate is made and can be used to replace the unsigned certificate.

Authentication

Secure FTP Server supports two levels of authentication with SSL:

- High - The server is configured so that it contains a certificate, but does not require a certificate from the FTP client.
- Highest - The server is configured so that it provides a certificate and also requests a certificate from the client. The server compares the client certificate to a list contained in its Trusted Certificates database. The server either accepts or rejects the connection based upon a match.

Explicit versus implicit SSL

Netscape originally developed Secure Socket Layer (SSL) for secure Web browsing. When both a client and server support the AUTH SSL command security is accomplished through a sequence of commands passed between the **two** machines. The FTP protocol definition provides at least two distinct mechanisms by which this sequence is initiated: explicit (active) and implicit (passive) security.

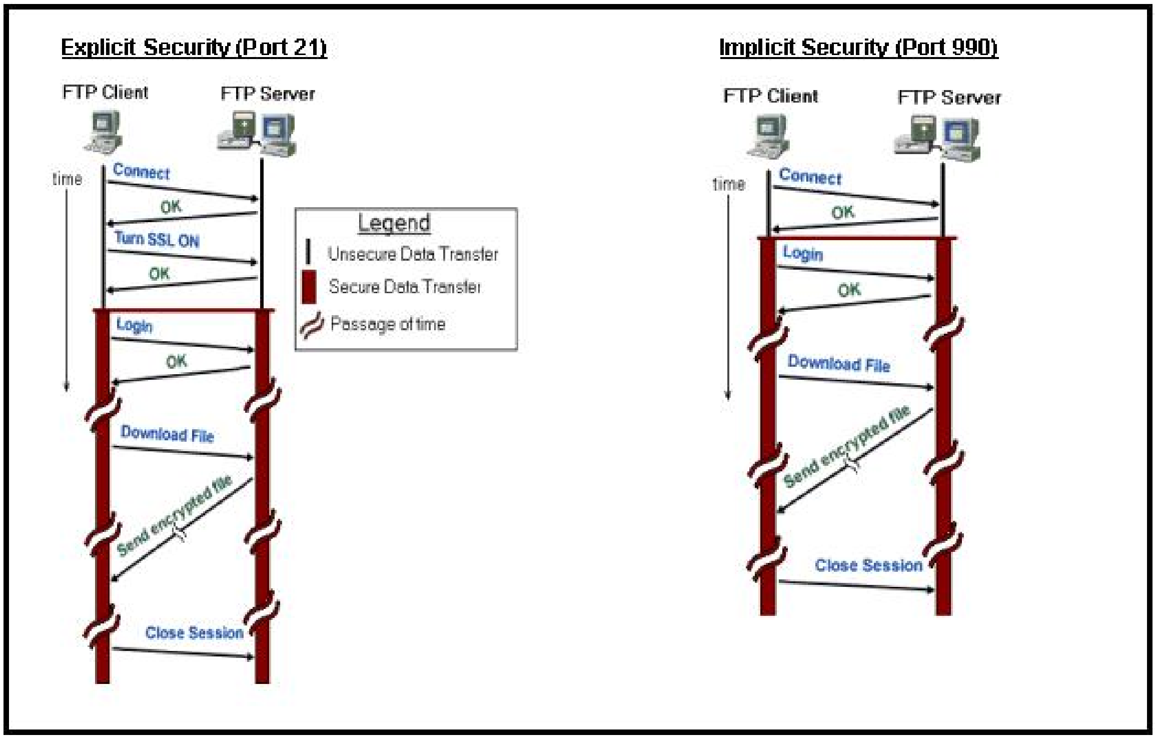
Explicit Security: In order to establish the SSL link, explicit security requires that the FTP client issue a specific command to the FTP server **after** establishing a connection. The default FTP server port is used. This formal method is documented in RFC 2228.

Implicit Security: Implicit security automatically begins with an SSL connection **as soon as** the FTP client connects to an FTP server. In implicit security, the FTP server defines a specific port for the client (990) to be used for secure connections.

Note:
Implicit SSL is discussed in various SSL drafts but is not formally adopted in an RFC. For strict compliance to standards, use the explicit method.

Because implicit SSL has a dedicated port strictly used for secure connections, implicit SSL connections require less overhead when you establish the session. There are various FTP servers that support this mode, including GlobalSCAPE Secure FTP Server, GlobalSCAPE Secure FTP Server, RaidenFTPD, IBackup's FTP server, and others.

You can think of implicit security as "always on" and explicit security as "turn on." The following diagram contrasts implicit and explicit SSL connections.



FTPS

FTPS is an enhancement to standard FTP that uses standard FTP commands (and protocol) over secure sockets. FTPS adds SSL security in both the protocol and data channels. FTPS is also known as FTP-SSL and FTP-over-SSL. You may also see the term SSL used in conjunction with TSL. SSL has been merged with other protocols and authentication methods into a new protocol known as Transport Layer Security (TLS). Secure FTP Server employs SSL/TSL to perform FTPS and keep your data secure.

SFTP

SFTP is an FTP-like protocol that uses SSH1 and SSH2 protocols to provide security. When clients make an SFTP (SSH2) connection with Secure FTP Server there are two components or layers involved: the Transport and Authentication layers.

Transport Layer

When users first attempt to connect to your SFTP site, the user's client software and the server determine whether the transmission should be encrypted or clear, compressed or uncompressed, what Method Authentication Code (MAC) to use, and what kind of encryption (cipher) to use.

Once the encryption method is chosen:

1. Secure FTP Server sends a public key to the client.
2. The client generates a session key, and encrypts it with the server's public key.
3. The client then sends the encrypted session key back to the server.
4. The server then decrypts the session key with its private key and from that time all transmitted data is encrypted with the session key.

Authentication Layer

After the Transport Layer is established, the server attempts to authenticate the client.

There are two methods Secure FTP Server can use for authentication.

- **Public Key Authentication Method: publickey**

  To use this method, the client will need a private key and public key. The public key is passed to the server. The server encrypts a random number with the public key and sends it to the client.

  1. The client asks the user for a passphrase to activate the private key.
  2. The private key decrypts the number and sends it back to the server.
  3. The server recognizes the number as correct and allows the connection.

- **Password Authentication Method: password**

  Using this method, the client sends its password to server. The client doesn't need to explicitly encrypt the password, because it will be automatically encrypted by the Transport Layer mentioned above. With this type of authentication, the connection will fail if the Transport Layer cannot encrypt the data.

After the encryption method is established, and authentication is complete, the two systems are ready to exchange secure data. The client sends a secured FTP connection along the encrypted data tunnel, the server responds and the user can then transfer files securely.

HTTP

What is Hyper Text Transport Protocol (HTTP)?

HTTP is the communications protocol for establishing a connection with a Web server and transmitting HTML pages to the client browser or any other files required by an HTTP client application.

HTTP is a stateless request/response system. The connection is maintained between client and server only for the immediate request after which the connection is subsequently closed.

How does HTTP support in Secure FTP Server differ from a typical Web Server?

Secure FTP Server is primarily a file transfer server, not a Web server. This means it is not meant to "serve up" Web pages such as a typical Web server does for connecting HTTP clients (such as you Web browser). However there are provisions for transferring files in the HTTP protocol, which is a convenience when a connecting partner, customer or employee doesn't have an FTP client installed but does have an HTTP client or access to a Web page with HTTP PUT capabilities (usually an ActiveX control or Java applet).

When Secure FTP Server is setup to allow HTTP file transfers, any HTTP client will be able to PUT (upload) or GET (download) files to the Secure FTP server provided the client supports both of these HTTP commands. Most Web browsers only support the GET command or if they support the PUT command, they provide no interface for browsing to the user's local file system in order to select and upload (PUT) files onto the Secure FTP Server. A few dedicated clients (such as CuteFTP Professional) and various thin clients (based on ActiveX controls or Java applets) support both PUT and GET capabilities, allowing these clients to transfer files to the Secure FTP server in both directions.

HTTP Limitations in Secure FTP Server

- Secure FTP allows you to customize messages sent by the server upon connection, login, maximum connections reached, and disconnect (for FTP sessions). Due to the nature of the HTTP protocol, custom login messages will not be displayed for connecting HTTP clients.
- Another limitation of HTTP is after a connection is established the browser will see the server's root folder instead of the user's home holder. A workaround is to setup a distinct Site for HTTP sessions.
- Certain syntax cannot be used when using some clients (such as a Web browser). When attempting to use an address in the format http://test:test@localhost the attempt fails and an invalid syntax error message appears. The protocol refuses to acknowledge this type of address and will not connect to the server. To connect, use the standard syntax: http://192.168.20.62, which prompts for the username and password.
- If you create an event rule that sends a notification email for each successful login event, an email is sent every time a user connected through HTTP changes directories. This is a result of HTTP being a stateless protocol. This can result in a large volume of notification email even when performing typical directory browsing.

HTTPS

HTTPS is the protocol for accessing a secure Web server where authentication and encrypted communication is possible. Using HTTPS in the URL instead of HTTP directs the message to a secure port number rather than the default Web port number of 80. The default TCP/IP port of HTTPS is 443. The session is then managed by a security protocol. HTTPS encrypts the session data using the SSL (Secure Socket Layer) protocol ensuring reasonable protection from eavesdroppers and man-in-the-middle attacks.

Secure Socket Layer (SSL) is a protocol for encrypting and decrypting data across a secure connection from a client to a server with SSL capabilities. The server is responsible for sending the client a certificate and a public key for encryption. If the client trusts the server's certificate, an SSL connection can be established. All data passing from one side to the other will be encrypted. Only the client and the server will be able to decrypt the data. The SSL protocol is the same protocol used in FTPS. Additional information on how SSL works is available under the FTPS section FTPS, SSL, and TLS.

Elements that work together to establish a secure HTTPS connection:

Client: The client must have SSL capabilities.

Certificate: Certificates are digital identification documents that allow both servers and clients to authenticate each other. A certificate file has a .crt extension. Server certificates contain information about your company and the organization that issued the certificate (such as Verisign or Thawte) while client certificates contain information about the user and the organization that signed the certificate. You can choose to either trust or distrust a certificate. In some cases, the client's certificate must be signed by the server's certificate in order to establish an SSL connection.

Session Key: The client and the server use the session key to encrypt data. It is created by the client via the server’s public key.

Public Key: The client encrypts a session key with the server’s public key. It does not exist as a file, but is produced when a certificate and private key are created.

Private Key: The server's private key decrypts the client's session. The private key has a .key extension and is part of the public-private key pair.

Certificate Signing Request: A certificate signing request is generated each time a certificate is created. A certificate signing request has a .csr extension. This file is used when you need to have your certificate signed. Once the Certificate Signing Request file is signed, a new certificate is made and can be used to replace the unsigned certificate.

Note:

In web pages that use HTTPS, the URL begins with 'https://' rather than 'http://'. HTTP clients should connect using standard requests (i.e. https://domain_name). Secure FTP Server can be set up to provide connecting clients with a certificate and even require that the client provide a certificate upon connect (to further validate the client's identity).

11

Automation

Automation

Secure FTP Server provides extensive automation functionality through commands, event rules, and a programmatic interface using COM APIs.

Custom Site Commands

Command-line executables can be configured to execute any program that the server has access from its filesystem. Open a program and provide a specific script or program to execute. You can give users permissions to execute the command, or you can configure an event rule to trigger a command.

Event Rules

Event rules enable task management automation. Event rules allow Secure FTP Server to carry out actions based on predetermined criteria. You can schedule routine tasks after a transfer. Event rules consist of an event trigger, optional conditions, and actions.

COM

Secure FTP Server's COM APIs allow you to program a unique or solution-specific interface and hook it right into the Secure FTP Server's functionality.

Custom Site Commands

Creating a custom command

Custom commands allow connecting users to execute programs with command line arguments on the Server. The connecting user would issue the command directly from their FTP client.

To add a custom site command

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. Select the **Commands** icon in the left window.
4. In the right pane, click the **New** button and click the **Command** tab.

Note:
You may also launch a Commands Wizard from the toolbar icon represented by the command prompt (black box)

5. Enter the name of the command in the **Command** box. This is the command name that the connect user will issue in his/her FTP client.
6. Type in a **Description** that will help you identify the function of this command.
7. Choose the path to the executable by browsing, or typing the path in the **Executable** box.
8. Select the **Redirect output to client** or **Redirect output to system log** check box, depending on where you want the output of the command sent. You can select both or neither.
9. In the **Advanced** tab type the parameters (if any) that will be passed to the command line. The variable format used is %N%. You may specify multiple variables or hard coded values. (For Example: -c %1% %2%).
10. Select the **Require Parameter** check box if you want to force the FTP client to send a minimum number of parameters. If the box is checked specify the minimum number of parameters required in the text box. You can also write a message in the **Invalid parameter count message** text box that users will receive when the parameter number is not met.
11. Select the **Enable process timeout** check box if you want the Secure FTP Server to return an error should the launched process fail to respond.
12. Enter the number of seconds you wish the server to wait before terminating the command.
13. Select the **Permissions** tab and verify that the appropriate users have permissions to run the newly created command.
14. Select **Apply**.

Custom command example

The following example command shows the configuration of a custom command from the perspective of both the **Secure FTP Server** and client. CuteFTP Professional will be the client used in this example, although any client that supports custom commands or raw FTP commands will work.

This command will compress an archive from the command line using CuteZIP's command line functions. Before attempting the following example, you will need to download and install CuteFTP Professional and CuteZIP. Both are available as a free evaluation trial and can be downloaded from www.globalscape.com.

Creating a custom command in Server

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. Select **Commands**.
4. In the right pane, click **New**. A new set of tabs appear at the top of the pane named **Commands**, **Advanced** and **Permissions**.

Commands tab

1. On the **Commands** tab, select the **Enable this command** check box.
2. In the **Command** field, enter the name **ZIP**.
3. In the **Description** field, enter **Compress selected files**.
4. In the **Executable** field, browse until you locate **C:\program files\globalscape\cutezip\cutezip.exe**.

> **Note:**
> You must have already installed CuteZIP on your computer.

5. Beneath **Output**, select the **Redirect output to client** check box.
6. Select the **Redirect output to system log** check box.

Advanced tab

1. Select the **Advanced** tab.
2. In the **Pass the following to the command** box, enter **-c %1% %2%**.
3. Select the **Require parameters** check box.
4. Select 2 in the **Command must have at least __ parameters** drop down box.
5. In the **Invalid Parameter Count Message** field, enter **Invalid Command! Usage Site ZIP [destination] [source]**.
6. Select the **Enable process timeout** check box.
7. Select **60** in the **Terminate Process if still running in __ seconds** drop down box.

Permissions tab

1. Select the **Permissions** tab.
2. Add the users or groups with permission to execute the command.

FTP Client Configuration

You must have CuteFTP Professional installed on your computer before you complete this section.

Creating a custom command for the FTP client

1. Start CuteFTP Professional and connect to Secure FTP Server.
2. Select **Tools** on the menu bar, find and expand **Custom Commands** then select **Edit Custom Commands**.

> **Note:**
> You must be connected to an FTP server in order for the **Commands** option to be available on CuteFTP Pro's menu bar.

3. Select the **New Command** icon and give your command a name.
4. Right-click the new command and select **Properties**.
5. For the **Label**, enter **ZIP Files on the Server**.
6. For the **Command**, enter **SITE ZIP %at[archive name] %ff** .

> **Note:**
> Commands must start with SITE and then the command name you used in Server.

7. Choose any key or key combination for the **Shortcut Key**.

8. Choose any icon for the **Toolbar Icon**.
9. Select the **Place on the Custom Commands toolbar** check box.
10. Select **OK** in the **Custom Commands Properties** dialog box.
11. Select **OK** to exit the **Commands** dialog box. Your custom command should now be enabled.

Testing the custom command

1. Start CuteFTP Pro and connect to Secure FTP Server.
2. Select **Tools > Custom Commands > New Command Name** from the menu.
3. Enter the item name to be zipped in the **Archive Name** window text box.
4. Select **OK**.
5. Monitor the output to the client log. You should receive various response messages indicating the progress of the archive.

Possible Error Situations

- If you repeat the hard coded parameters in both the client and server, such as SITE ZIP -c %at[archive name] %ff is used in the client, and -c %1% %2% in the sever, then the first parameter (-c) that the client sends will be used as %1%. So the resulting string would be: -c -c filename.ext. Therefore it is important to educate the user on the proper syntax and supply most of the hard coded parameters on the server side.
- If you do not add the user to the **Permissions** table in the **Permissions** tab on the server they will receive a "Permission Denied" error.
- Certain command line utilities that may show a Windows prompt or other dialog may not execute properly when called from the FTP Engine while it is running as a service. This is especially true when the service is being logged in from a Local System account.
- The server may return an error if the client provides the wrong number of parameters or invalid parameters. In order to limit security vulnerabilities to the server, the server administrator should only allow limited access to commands that launch processes.

Event Rules

Event Rules

Event Rules automate management tasks. Define an event rule to trigger an action, or several actions, when specified criteria are met. For example, event triggers can be used to initiate additional activities after a file has been uploaded/downloaded. You can synchronize content across systems or provide an automatic response with an event rule, or you can trigger a custom command to run a custom application or script. You can specify

Event rules in Secure FTP Server consist of triggering **Events**, any optional **Conditions** affecting the event rule, and the resulting **Actions** that are carried out.

When multiple actions are defined for a single event rule, Secure FTP Server carries out the actions in the following order:

1. Execute Custom Command
2. Email Notification
3. Stop Processing Rules

> **Warning:**
> It is possible to configure event rules that create infinitely recursive cycles. Since all event rules operate synchronously, a file upload event cannot be completed until all corresponding event actions are finished. This could lead to unpredictable server behavior due to conflicts with shared access to the same files or deleting open files. Be careful not to create circumstances where such recursive cycles might occur. For file upload events, recursive cycles are not typical. It is recommended that you move files on the same server with the filesystem—not FTP.

Events

The following events can trigger actions:

Server Events

- **Rotate Log**—When the current activity log closes and opens a new one.
- **Service Stop**—When the **Secure FTP Server** service stops.
- **Service Start**—When the **Secure FTP Server** service starts.

Site events

- **Site Start**—When the site starts.
- **Site Stop**—When the site stops.

Connection Events

- **User connect**—When a user connects to the site (this occurs before log in).
- **User connect failed**—When a user attempts to connect and fails (this can occur before log in).
- **User disconnect**—When a user disconnects from the site (this can occur before log in).

User Events

- **User account disable**—If the user account is disabled by the administrator or by the server.
- **User quota exceeded**—If the user has taken too much disk space on the server.
- **User logout**—If the user closes a session gracefully.
- **User login**—If the user logs in to the server.
- **User login failed**—If the user attempts an incorrect username or password.
- **User password change**—If the user or administrator changes a user's password.

File System Events

- **File delete**—If a file is deleted from the site.
- **File upload**—If a file is uploaded to the site.
- **File download**—If a file is downloaded from the site.
- **File rename**—If a file on the site is renamed.
- **Folder create**—If a folder is created on the site.
- **Folder delete**—If a folder is deleted from the site.
- **Upload Fail**—If an upload does not occur.
- **Download Fail**—If a download does not occur.
- **Folder change**—If a user navigates to a new folder on the site.
- **File move**—If a file is transferred to another location.

Conditions

Conditions allow you narrow the trigger definition an event rule. Conditions are optional: you do not have to define a condition on an event rule to make it trigger an action, but they do allow fine control over when an action may take place.

Server Conditions

You can only apply these conditions to **Server events**.

- **If service is running** – The Secure FTP Server service is currently running.
- **If log type** - The log type is a specific type.
- **If log location** - The log location matches a specific path.
- **If old log file path -** The log file path matches a specific path.
- I**f new log file path -** The log file path matches a specific path.
- **If old log file name -** The log file path matches a specific path.
- **If new log file name -** The log file path matches a specific path.

Site Conditions

You can only apply this condition to **Site events**.

- **If site is running** – The site has already started and is currently running.

Connection Conditions

You can apply these conditions to **Connection events**, **User events**, and **File system events**.

If Remote IP

- a connection is made from a remote IP address that matches a predefined IP address or IP mask.
- a connection is made from a remote IP address that does NOT match a predefined IP address or IP mask.

If Local IP

- a connection is made to a local IP address that matches a predefined IP address or IP mask.
- a connection is made to a local IP address that does not match a predefined IP address or IP mask.

If Local Port

- a connection is made on a predefined port.
- a connection is made NOT on the predefined port.
- a connection is made on one of a predefined range of ports.
- a connection is made NOT on one of a predefined range of ports.

If Protocol

- an FTP/SSL/SFTP connection has been made or is being used.
- a connection has been made or is being used that is NOT an FTP/SSL/SFTP connection.

User Conditions

You can apply user conditions to **User events** and **File system events**.

If User

- the user account belongs to a specific group or set of groups.
- the user account does not belong to a specific group or set of groups.

If Login

- a user name matches a specific word.
- a user name does not match a specific word.
- a user name contains a specific string of characters.
- a user name does not contain a specific string of characters.

If Account Enabled

- a user account is enabled.
- a user account is disabled.

Back to top

If Settings Level

- the user belongs to a predefined Setting Level.
- the user does NOT belong to the predefined Settings Level.

if Full Name

- a user's name matches a predefined name.
- a user's full name does not match a predefined name.
- a user's full name contains a predefined string of characters.
- a user's full name does not contain a predefined string of characters.

if Description

- the user's description matches a predefined description.
- the user's description does NOT match a predefined description.
- the user's description contains a predefined string of characters.
- the user's description does NOT contain a predefined string of characters.

if Comment

- the user's comment matches a predefined comment.
- the user's comment does NOT match a predefined comment.
- the user's comment contains a predefined string of characters.
- the user's comment does NOT contain a predefined string of characters.

if Email Address

- the user's email address matches a predefined address.
- the user's email address does NOT match a predefined address.
- the user's email address contains a predefined string of characters.
- the user's email address does NOT contain a predefined string of characters.

if Phone Number

- the user's phone number matches a predefined phone number.
- the user's phone number does NOT match a predefined phone number.
- the user's phone number contains a predefined string of characters.
- the user's phone number does NOT contain a predefined string of characters.

if Pager Number

- the user's pager number matches a predefined number.
- the user's pager number does NOT match a predefined number.
- the user's pager number contains a predefined string of characters.
- the user's pager number does NOT contain a predefined string of characters.

if Fax Number

- the user's fax number matches a predefined number.
- the user's fax number does NOT match a predefined number.
- the user's fax number contains a predefined string of characters.
- the user's fax number does NOT contain a predefined string of characters.

if Home Folder

- the location of a user's home folder matches a predefined physical location.
- the location of a user's home folder does NOT match a predefined physical location.

if Home Folder is root

- the user's home folder is their root directory.
- the user's home folder is NOT their root directory.

if Quota Max

- the user's account has a size limit equal to a predefined size in Kilobytes.
- the user's account has a size limit less than or equal to a predefined size in Kilobytes.
- the user's account has a size limit less than a predefined size in Kilobytes.
- the user's account has a size limit NOT equal to a predefined size in Kilobytes.
- the user's account has a size limit NOT less than or equal to a predefined size in Kilobytes.
- the user's account has a size limit NOT less than a predefined size in Kilobytes.

if Quota Used

- the user has used a predefined amount (in kb) of allowed disk space.
- the user's filled disk space is less than or equal to a predefined amount (in kb) of allowed disk space.
- the user has used less than a predefined amount (in kb) of allowed disk space.
- the user has NOT used a predefined amount (in kb) of allowed disk space.
- the user's filled disk space is NOT less than or equal to a predefined amount (in kb) of allowed disk space.
- the user has NOT used less than a predefined amount (in kb) of allowed disk space.

if Invalid login attempts

- the user has attempted and failed to login a predefined number of times.
- the user's failed login attempts are less than or equal to a predefined number.
- the user's failed login attempts are less than a predefined number.
- the user has NOT attempted and failed to login a predefined number of times.
- the user's failed login attempts are NOT less than or equal to a predefined number.
- the user's failed login attempts are NOT less than a predefined number.

if User can change password

- the user has permission to change their own password.
- the user does not have permission to change their own password.

if Home IP

- the user's allowed IP address matches a predefined IP address or set of IP addresses.
- the user's allowed IP address does not match a predefined IP address or set of IP addresses.

if User can connect using SSL

- the user has SSL capability enabled.
- the user does not have SSL enabled.

if User can connect using FTP

- the user has configured a site and has an FTP account.
- the user does not an FTP site with an account configured.

if User can connect using SFTP

- the user has SFTP capability enabled.
- the user does not have SFTP enabled.

File System Conditions

You can apply file system conditions only to **File system events**.

if Virtual Path

- the file or folder exists at a predefined virtual location.
- the file or folder does NOT exist at a predefined virtual location.

if Physical Path

- the file or folder exists at a predefined physical location (the full folder path including the file name).
- the file or folder does NOT exist at a predefined physical location (the full folder path including the file name).

if Physical Folder Name

- the file or folder exists in a predefined physical folder (the folder path without a file name).
- the file or folder does NOT exist in a predefined physical folder (the folder path without a file name).

if File Name

- the file name matches a predefined string of characters.
- the file name does not match a predefined string of characters.

Event Properties

You can apply particular properties to specific conditions for **Upload Fail** and **Download Fail** only in **File system events**, for **User Login Failure** and **User Logout** in **User events**, and for **User Connect Failure** in **Connection events**.

These are special conditions are defined by using the **specific reason** parameters found in the drop down menu in the **specify rule condition and action parameters** section.

File System Events

if Upload Fail (or Download Fail)

- the upload/download was aborted by User.
- access was denied.
- connection was closed.
- file was banned type.
- bandwidth quota was exceeded.

User Events

if User Login Failure

- the user account was disabled.
- an invalid password was used.
- the protocol used was not supported.
- the IP was restricted.
- there were too many connections per IP
- there were too many connections per site.
- there were too many connections per user.

if User Logout

- the FTP session was closed due to error.
- the FTP session was closed by a timeout.
- the FTP session was closed by the user.
- the IP address was banned.
- the maximum number of incorrect logins was reached.
- the TCP/IP connection was closed by a peer.
- the User was kicked by the administrator.

Connection Event

if User Connect Failure

- the IP address was rejected.
- the IP address was rejected and banned.
- there were too many connections per IP.
- there were too many connections per site.

Actions

Actions are the results of event triggers. You can specify multiple actions to occur from a single trigger.

Rule actions

- **Execute command**—The custom command in a specific location is triggered.
- **Send Notification Email**—An email message is sent to the address specified.
- **Stop processing more rules—**No further rules are processed.

RULE ACTION POSSIBILITIES

ACTIONS EVENTS	EXECUTE COMMAND IN FOLDER	SEND NOTICE EMAIL	STOP PROCESSING MORE FILES
SERVER EVENTS			
Service Start	X	X	X
Service Stop	X	X	X
Timer	X	X	X
Rotate Log	X	X	X
SITE EVENTS			
Site Start	X	X	X
Site Stop	X	X	X
CONNECTION EVENTS			
User Connect	X	X	X
User Disconnect	X	X	X
User Connect Fail	X	X	X
USER EVENTS			
Account Disabled	X	X	X
Quota Exceeded	X	X	X
Password Changed	X	X	X
User Login	X	X	X

User Logout	X	X	X
User Login Failure	X	X	X
FILE SYSTEM EVENTS			
File Delete	X	X	X
File Upload	X	X	X
Before Download	X	X	X
File Download	X	X	X
File Rename	X	X	X
Folder Create	X	X	X
Folder Delete	X	X	X
Folder Change	X	X	X
File Move	X	X	X
Upload Fail	X	X	X
Download Fail	X	X	X

Stop Processing

Stop Processing ends any further rule processing. The Stop Processing action is part of the process that is used in the development of how event rules operate. The example below shows three rules that are triggered with an On Upload event. The stop processing action causes the other two processes in this example to stop.

Creating, Editing, and Disabling event rules

To Create an Event Rule

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. Select **Configuration > Create New Event Rule** from the menu. The **Create New Rule** window appears.
4. Enter a name for the rule.
5. In **Should be applied when**, select the event you want as a trigger.
6. Select **OK**. The **Create New Rule** window closes and the conditions and actions available for your rule are displayed in the right-hand pane.
7. Optionally select any **conditions** for the event rule.
8. Specify the action(s) the event rule triggers.

- Choose **Execute command in folder** to run any custom command you have created for the site.
- Choose **Send email notification** to send an email message to the address you entered in the server **SMTP Configuration** tab, and optionally send a message to a user.

- If you want other rules for the event to be ignored if this rule is met, select **Stop processing more rules**.

9. In **Specify rule condition and action parameters**, select the blue and red text links to toggle behavior and select executables, email addresses or define file paths used in definition of the event rule. Secure FTP Server does not save the rule unless it is adequately defined.
10. Select **Apply** to enable the rule.

> **Note:**
> Red links in **Specify rule condition and action parameters** indicate parameters that have not yet been defined. They must be defined to save the rule.

To edit an event rule

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. From the left-hand navigation tree, select Event Rules under the Site you want to configure.
3. The Event Rules for that Site display in the right-hand pane.
4. Select the event rule you want to change.
5. Select **Edit**. The event rule's details appear in the right-hand pane.
6. Make any desired changes to the event rule.
7. Select **Apply**.

To disable an event rule

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. From the left-hand navigation tree, select Event Rules under the Site you want to configure.
3. The Event Rules for that Site display in the right-hand pane.
4. Clear the check box next to the event rule you want to disable.
5. Select **Apply**.

To re-enable an event rule

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. From the left-hand navigation tree, select Event Rules under the Site you want to configure.
3. The Event Rules for that Site display in the right-hand pane.
4. Select the check box next to the event rule you want to re-enable.
5. Select **Apply**.

To delete an event rule

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.

2. From the left-hand navigation tree, select Event Rules under the Site you want to configure.
3. Select the event rule you want to delete.
4. At the far right of the Administrator interface, click **Delete**. A **Delete Rule** dialog appears.
5. Select **Yes**. The rule is deleted from the site.

Using an event rule to trigger a custom command

You can configure the server to automatically run custom commands when specific events occur. You can find a list of the events you can use as triggers in Server events and conditions.

To automatically start a custom command

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. Select **Event Rules**.
4. In the right pane, select **New**. The **Create New Rule** dialog appears.
5. Enter a name for the rule.
6. Select the event trigger from **Should be applied when:**
7. Select **OK**. The **Create New Rule** window closes and the new rule is displayed in the right-hand pane.
8. If you need to apply any conditional behavior select it from **Specify rule conditions**.
9. From **Specify rule actions**, select **Execute command in folder**.
10. From **Specify rule condition and action parameters**, choose the red link **select**. The **Custom Command** window appears.

> **Note:**
> Red links in **Specify rule condition and action parameters** indicate parameters that have not yet been defined. They must be defined to save the rule.

11. Select the desired command from **Select command**.
12. Optionally include any parameters for the command in **Specify command parameters**. You can also select the items in the **Available Tags** list to add them as parameters.
13. In **Specify command working folder** type the path or click the yellow Browse button to choose the folder where the custom command executable resides.
14. Select **OK**.
15. At the top of the right pane, make sure **Rule Enabled** is selected.

Customizing event rule email notifications

You can create a custom email message for every event rule you define.

To customize an event rule email message

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.

2. Select the Site you want to configure from the left-hand navigation tree.
3. Expand **Event Rules**.
4. Select the event rule with the message you want to customize.
5. In the middle of the right-hand pane, make sure **Send notification email** is selected.
6. At the bottom of the right-hand pane, select the blue **notification email** text. The **Edit Mail Template** window appears.
7. In the **Subject** box, type any text you would like as a subject.
8. In the **Body** box, type any text you want.
9. In the **Available Tags** box, click any property you want to insert in the email message. The text surrounded by per cent signs signifies text that will be replaced by the server with specific information about the event, the user, or the connection.
   - If you just want the specific text in your email message, click the text surrounded by the per cent sign in the right column of the **Available Tags** box.
   - If you want the specific text, and the explanatory text before it, click the text in the left column of the **Available Tags** box.
10. If you want to send a copy of the message to the involved user select the **CC Mail Notification to user** check box. In order for the **CC Mail Notification to user** check box to be available your rule must be based on a **User Event**. To base a rule on a **User Even**t, create a new rule and select an option from the **User Event** list.
11. Select **OK**.

Note:
Red links in **Specify rule condition and action parameters** indicate parameters that have not yet been defined. They must be defined to save the rule.

Note:
You can edit the email messages even if you are not familiar with HTML. If you delete all the HTML tags the message is sent as a plain text message.

To send email as plain text

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the Site you want to configure from the left-hand navigation tree.
3. Expand **Event Rules**.
4. Select the event rule with the message you want to customize.
5. In the middle of the right pane, make sure the **Send notification** email check box is selected.
6. In the bottom of the right pane, click the blue **notification** email text. The **Edit Mail Template** window appears.
7. In the **Body** box, delete all the HTML tags.
8. Select **OK**.

Configuring SMTP email notification

You can configure the server to send email alerts whenever certain events occur. You must provide **Secure FTP Server** with the address for an outgoing mail server, an address for the administrator, and other details.

To set up the server to send email notifications

1. Select the **Server tab** in Secure FTP Administrator and select a server the server you want to configure.
2. Select the **SMTP Configuration** tab from the right-hand workspace.
3. In **SMTP Server Address**, enter the address of the mail server the Secure FTP Server will use to send outgoing messages.
4. In **SMTP Server Port**, enter the port number where the mail server accepts messages. The standard is **25**.
5. If Secure FTP Server can connect to the mail server without a log in, leave the **Server requires authorization** check box clear and skip to step ten. If the mail server requires a user name and password from the Secure FTP Server computer, select the **Server requires authorization** check box and continue with step eight.
6. In the **Login** box, enter the user name needed to connect to the mail server.
7. In the **Password** box, enter the password needed to connect to the mail server.
8. In the **Name** box of the **Send Messages FROM** group, enter any name you would like for the "From Name" field.
9. In the **Address** box of the **Send Messages FROM** group, enter any address you would like for the "From Address" field.
10. In the **Name** box of the **Send Messages TO** group, enter the name of the server administrator, or any name you wish.
11. In the **Address** box of the **Send Messages TO** group, enter the email address of the person that should be notified of server events.
12. Select **Apply**.

COM

COM APIs

You can interact directly with Secure FTP Server from your own custom applications using any COM enabled programming language such as Visual Basic (VB), Java, or C++. You can create a script with the development IDE of your choice. To create a new script file, you must be familiar with programming concepts and should have experience with COM enabled programming languages.

For more information see GlobalSCAPE's *Secure FTP COM API Reference Manual*.

12

Troubleshooting

Accepting and signing SSL certificates

If you require certificates for SSL connections users will have to provide signed certificates, or send certificate signing requests. You will have to add the signed certificates to a trusted list, and sign the certificate signing requests before the users can connect.

To add a certificate to the trusted list

1. Choose **Tools > Certificate Manager** from the menu bar. The **Certificate Manager** will appear.
2. If the certificate is already displayed in the **Pending Certificates** window Click **Make Trusted** and skip to step 6. If the certificate is not in the window continue with step three.
3. Click **Import** under the **Trusted Certificates** window.
4. Browse to the folder that contains the client's certificate file and select the file. GlobalSCAPE Secure FTP Server can import a digital certificate that is in any of the following formats: PEM, DER, PKCS#7, PKCS#12. The Private Key associated with the digital certificate can be in one of the following formats: PEM, DER, PKCS#8, PKCS#12.
5. Click the **Open** button. GlobalSCAPE Secure FTP Server will attempt to automatically detect the certificate format. If it is unable to determine the format, or if the import fails, an informative error message will appear. You might have to manually convert a digital certificate to one of the above formats in order to import it into GlobalSCAPE Secure FTP Server. Please consult the distributor/vendor of your certificate for details on this process.
6. The certificate is added to the **Trusted Certificate**s database. Clients submitting that certificate will now be able to connect to the server.

To sign a certificate

1. Obtain the Certificate Signing Request file (.csr). This can be done through email or any other file delivery method.
2. Choose **Tools > Certificate Signing Utility** from the menu bar. The **Sign Certificate Reques**t dialog will appear.
3. **Client certificate reques**t - Click the folder to browse and select the Certificate Signing Request (.csr) file you want to sign.
4. **Output client certificate** - Browse and choose a folder in which to save the signed certificate (.crt) file.
5. **Server certificate** - Browse and choose the certificate you will sign with. This certificate must be in your trusted certificate database in order for clients submitting the signed certificate to connect to the site.

6. **Server private key** - Browse and select the private key file (.key) associated with the server certificate.
7. Enter the **Passphrase** associated with the server certificate.
8. Choose an **Expiration date**.
9. Click **OK**. The new certificate will be saved in the folder you selected.
10. Return the certificate file (.crt) to the user.

IP conflicts

If you already have an FTP server running and attempt to configure Secure FTP Server to listen on the same IP and port, you will receive an error message stating that you cannot start that site. This is by design: you are not allowed to have two services listening on the same IP and port.

Either stop the other FTP server or configure the servers to listen on different IP addresses or ports.

If you have Microsoft Internet Information Services on your computer, see Unable to create socket on port 21, for instructions resolving IP conflicts with Secure FTP Server.

FTP client hangs on the list command

Internet Security and Acceleration (ISA) Server maintains secondary connections for secure network address translation (NAT) clients in Kernel mode, which can improve data throughput for protocols that use secondary connections. Secondary connections for secure NAT clients are only supported if an application filter that can process the protocol is installed on ISA Server.

For example, File Transfer Protocol (FTP) uses the secondary connection over port 20 to transfer data. Because the primary connection is enabled only if the requirements for all applicable rules are met, and the secondary connection is established only after the primary connection is established, there is no need to inspect the traffic for this connection.

For more information on correcting this problem, view the following articles in Microsoft's knowledge base:

http://support.microsoft.com/default.aspx?scid=kb;en-us;Q279347

http://support.microsoft.com/default.aspx?scid=kb;en-us;Q294679

I cannot connect to Secure FTP Server using SSL

The most common cause for this is because the server is behind a firewall and the firewall is unable to decrypt the information being transferred between the client and server. To correct this problem you need to specify a range of ports for the server to use and open the same range of ports on your firewall.

To specify a range of ports

1. Start the Administrator Interface and connect to the server.
2. At the bottom of the left pane, click the **Server** tab.
3. In the left pane, expand a server group, and server.
4. Select a site.
5. In the right window, click the **Site Options** tab.
6. Select the **Assign PASV mode IP address** check box.

7. Enter the **IP address** for your firewall in the IP box.
8. Enter the range of ports you want to limit the server to in the **Port Range** boxes.
9. Click the **Apply** button.

> **Note:**
> You must also open the same range of ports on your firewall.
> You must configure your SSL FTP client to use PASV mode.

Files/Folders do not show the date and time modified, only the year

When a file or folder was last modified in the same year as the server current time, the HOURS:MINUTES display in the directory listing. However, if it was last modified during a previous year, it only displays the YEAR modified.

This is not a bug, it is standard directory listing behavior for Unix systems. For example, if you look at the sites pre-built into CuteFTP Professional, you will notice that this type of folder/file listing is typical for many FTP servers. Additionally, this listing behavior is standard for Redhat Linux, Macintosh operating system support, Microsoft and Palm.

Port conflicts

When you create more than one site, each site needs its own port, the place where it listens for user connections and commands. The port for a site is often described as the data channel.

When you are running multiple sites, check your site settings to insure that each site has its own port. When two sites are both assigned the same port, you may have a port conflict. Only one site will be able to run at a time.

FTP sites

The default port for FTP sites is 21, but you can select any number between 1 and 65,535.

> **Note:**
> Assigning a port number under 1024 (other than 21) may lead to conflicts with other programs running on your computer.

Users have to wait a long time before they can resume an upload

If users frequently lose their connection to Secure FTP Server, resuming an upload may take several minutes.

If you want to allow users with problematic connections to quickly resume broken uploads, you will need to set their accounts to time out quickly.

If users lose their connection to Secure FTP Server while uploading a file, the portion of the file on the server will remain locked to changes (like a resumed upload) until the server tries to disconnect. Generally, Secure FTP Server will not try to disconnect until nothing has happened for the amount of time set in **Enable time out**.

The default connection time out value in **Enable time out** is 600 seconds (ten minutes). It can be set as low and 1 second, and as high as 9999 seconds (almost 3 hours), but a connection time out at 30 seconds or 60 seconds will be less likely to interfere with transfer tasks.

When you set the time out value, you can tell the users the value, and if they have an FTP client that automatically attempts to reconnect and resume the transfer, they can set their client to wait the same amount of time before reconnecting.

To set accounts to time out quickly

1. In Secure FTP Administrator, select the **Server** tab. You should be connected to the server.
2. Select the User or User Setting Level you want to configure from the left-hand navigation tree.
3. Select the **Quota** tab from the right-hand pane.
4. Select **Enable time out** and enter the maximum allowable seconds of inactivity allowed before the user is disconnected. If the check box is gray or blank, click until you see a black check in a white box. See Inheritance for more information.
5. Select **Apply**.

Cannot create a new user or group in an ODBC data source

MDAC not installed

If you can't create a new user or group while using an ODBC data source for user authentication, you must install Microsoft Data Access Components (MDAC) 2.6 or higher. If you are running GlobalSCAPE Secure FTP Server on Windows XP you will not need to make any changes.

You can download Microsoft Data Access Components (MDAC) 2.6 or 2.7 from http://www.microsoft.com/data/download.htm

If you are using an Access database, you may also need to download a Jet driver. Go here http://support.microsoft.com/default.aspx?scid=KB;EN-US;Q271908 for more information about the Jet driver.

Using v1.x or v2.x database with v3.x

If you have upgraded your server from v1.x or v2.x to v3.x you must also upgrade your database tables. See Creating tables for your ODBC data source for v3.x database table format.

Forgotten password

If you cannot remember your password to log in to the administration interface, you will need to reset the password. Resetting the administrator username and password using this procedure will result in the loss of all user and group specific settings. The user accounts and their folder structures will remain, but permissions and settings will be lost.

To reset the administrator username and password

1. In the Windows **Control Panel**, choose **Administrative Tools > Services** and stop the GlobalSCAPE Secure FTP Server (GSFTP Server) service.
2. Navigate to the folder where GSFTP Server is installed. By default this will be **C:\Program Files\GlobalSCAPE\Secure FTP Server *X.x***.
3. The user accounts and folder structures are stored in one or more of the files containing a **.aud** extension. Copy these configuration files to a safe place for backup.
4. Using the Windows **Add/Remove Programs** utility, uninstall GSFTP Server.
5. Reinstall GSFTP Server entering a new administrator username and password. If you need to download the software you can do so from here ftp://ftp.globalscape.com/pub/gsftps/gsftps.exe.
6. Start the software and login through the administrator interface with a new username and password.
7. Recreate your FTP site(s) with the EXACT same site name as what was previously used. The site name must match character for character.
8. In the Windows **Control Panel**, go to **Administrative Tools > Services** and stop the GSFTP Server service.
9. Navigate to the folder containing the **.aud** files from Step three above and copy the files back into the folder where you have installed GSFTP Server. Choose **Yes** when asked if you want to overwrite the existing **.aud** files.
10. Restart GSFTP Server and log in. The individual groups and user accounts should be preserved. You will now need to assign permissions and settings.

Users in ODBC database cannot login

If you are using an ODBC data source for authentication on your server, and users always get an error saying they are not logged in, make sure the "anonymous" row in the "ftpserver_users" table is set to "0" or "1". It cannot be set to "Null".

Resetting the login

To reset the administration login you will need to recreate the ftp.cfg file that is located in the Server program directory. You can keep your existing sites but you will need to complete the following steps to make a new ftp.cfg without losing your site information.

1. Browse to the program directory (c:\program files\globalscape\SecureFTP Server).
2. Copy any .aud files (note the name of the aud files, you will use them in step Five) and save the files in a new location.
3. Stop the **GlobalSCAPE Secure FTP Server** service.
4. Uninstall GlobalSCAPE Secure FTP Server from the control panel.
5. Reboot the computer (this completes the uninstall of the service).
6. Download the server from the following link: http://www.globalscape.com/files/gsftps.exe.
7. Install the server. You will be prompted to create an administrator login during setup.
8. Make a new site (or sites) using the filenames from the aud files. For example if the aud file is called "MyServer.aud" you need to make a site called "MyServer")

9. Stop the service and copy the .aud files you saved in step two back to the directory, overwriting the new aud file that was created by the install.

   When you start the service again it will read the ftp.cfg (which points to the .aud files), and ask for your username and password. The program will recognize the username and password you created in the new install, but will show all your old site settings. The administration settings and password are kept in the ftp.cfg file, the site settings are in the .aud file (so you have created a new ftp.cfg file but you are using your old .aud files).

Automatically encrypting and compressing transfers

You can compress and encrypt files after transferring them by using GlobalSCAPE Secure FTP Server's Custom Site Commands. The benefits of encrypting files prior to or after transfer depends on the circumstances and level of trust for the particular host. You can also encrypt and compress files before transfer using GlobalSCAPE's CuteZIP and CuteFTP Pro.

In the example script below, an entire folder (including sub-folders) is compressed, encrypted (using Twofish 128 bit encryption) and then transferred via regular FTP to an FTP server. Since the archived file is encrypted, there is no need to connect using SSL, OTP, or SSH2 unless you wished to also protect the login process.

Example

```
Dim WshShell, MySite, Return
Set WshShell = CreateObject("WScript.Shell") 'Window's Scripting
Host shell object
'next line calls the run method of the WSH shell object. It returns
true once CuteZIP does its thing.
'The complete command line instructions for CuteZIP are located
here.
If Return = WshShell.run ("c:\progra~1\global~1\CuteZIP\cutezip.exe
-c -p12345 c:\archive c:\temp", 0, true) Then
Set MySite = CreateObject("CuteFTPPro.TEConnection")
MySite.Option ("ThrowError") = True
MySite.Host = "ftp://user:pass@myftpsite.com" 'one of the ways to
connect using the Pro TE
MySite.Connect
MySite.Upload "c:\archive.zip" 'upload the new archive, then check
to see if it made it up to the server.
if not CBool(MySite.RemoteExists("\archive.zip")) then
MsgBox "Failed to Upload, Exiting!"
Else
MsgBox "Success!"
End If
MySite.Disconnect
MySite.Close
Else
MsgBox "Compression and Encryption Failed, Exiting!"
End If
```

> **Note:**
> You can optionally protect the FTP login by connecting with SSL, SSH2 or OTP using CuteFTP Pro's Transfer Engine (GlobalSCAPE's Secure FTP Server supports SSL, OTP, and SSH2 logins). Use the Protocol property to set the connection type prior to calling MySite.Host and MySite.Connect.

Unable to create socket on port 21

This error is generally the result of running the Microsoft IIS FTP server and the GlobalSCAPE Secure FTP server on the same computer.

By default the FTP server in Microsoft IIS binds to port 21 on any and all IP addresses. If you want to run both the IIS FTP server and GlobalSCAPE Secure FTP Server, you will need to disable socket pooling for the IIS FTP server.

To disable socket pooling in IIS FTP server

1. In Microsoft Internet Information Services, stop the FTP site (see below).
2. Open a command prompt window (see below).
3. Change directories to **C:\InetPub\Adminscripts.**
4. Type **CSCRIPT ADSUTIL.VBS SET MSFTPSVC/DisableSocketPooling TRUE** and press the **Enter** key. You should get a response like this;

   disablesocketpooling : (BOOLEAN) True

4. Exit the command prompt and restart the FTP site. This should prevent IIS from binding to all IP addresses on port 21, freeing up an IP address for GSFTPS sites on the default FTP port of 21.

More information on Microsoft IIS socket pooling is available at;

http://support.microsoft.com/default.aspx?scid=kb;en-us;259349

http://support.microsoft.com/default.aspx?scid=kb;EN-US;238131

These links discuss the IIS Web server, but the same information applies to the IIS FTP server.

To open a command prompt window and change directories

1. On the Windows task bar, choose **Start > Run**.
2. Type **cmd** and click **OK**. The command prompt window opens.
3. Type **cd InetPub\AdminScripts** and press the **Enter** key on your keyboard.
4. You should now see **C:\InetPub\Adminscripts>**.

To stop and start an IIS FTP site

1. On the windows desktop, right click **My Computer** and choose **Manage**. The **Computer Management** window opens.
2. In the left pane, select **Services and Applications**.
3. In the right pane, double-click the **Internet Information Services** folder. New items appear in the right pane.

4. In the right pane, select **Default FTP site**.
5. On the menu bar, choose **Action > Stop** to stop the FTP site, or choose **Action > Start** to start the FTP site.

Man in the middle attack

If users contact you reporting that their SSH FTP client is reporting a possible "Man-in-the-middle" attack, either they have an old SSH keypair for your site, or they are actually victims of a possible attack.

FTP clients will report a possible Man-in-the-middle attack whenever you change your SSH keypair. The client software is correctly reporting that your "fingerprint" has changed.

To avoid confusion, notify your users every time you change your keypair.

Site settings are lost when the service is stopped

If you lose settings and user accounts whenever you restart the Secure FTP Server service, you need to reset permissions on the computer where the Server service is running.

The service runs under a user account. That account must have full administrative rights to the folder where you installed Secure FTP Server. With administrative rights the service can save all your settings.

Server service will not start

If you get an error similar to "Could not start the Secure FTP Server service on Local Computer..." there may be a minor problem in your registry.

To start the Secure FTP Server service

1. On the computer where Secure FTP Server is installed, select **Start>Run**.
2. The **Run** dialog appears. In **Open**, enter **regedit**. The **Registry Editor** window appears.
3. Navigate to **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Secure FTP Server**
4. Select **ImagePath**.
5. On the **Registry Editor** menu bar, choose **Edit > Modify**. An **Edit String** window appears.
6. Add quote marks before and after the text in the **Value Data** box. When you have placed the quote marks it should look similar to this: **"C:\Program Files\GlobalSCAPE\Secure FTP Server\cftpstes.exe"**.
7. Select **OK**. The **Edit String** window closes.
8. Close the **Registry Editor**. Secure FTP Server should restart.

Connecting with Microsoft Internet Explorer

A connection to Secure FTP Server using Microsoft's Internet Explorer (MSIE) can normally be accomplished using the default settings for both products. You may need to allow two or more concurrent connections from the same user and the same IP address to facilitate connection from a Web browser.

For your convenience, some basic connection and troubleshooting information is shown here. If you continue to have trouble establishing a connection using MSIE, you should consult the MSIE documentation or help file.

Depending on the firewall configuration on either the FTP client or server side, you may need to change the mode that is used by the FTP client. MSIE 5 and later support both Standard (PORT) and Passive (PASV) modes.

To Change the MSIE FTP Client Mode

1. Start Internet Explorer.
2. On the **Tools** menu, select **Internet Options**.
3. Select the **Advanced** tab.
4. Under **Browsing**, clear **Enable folder view for FTP sites**.
5. Select **Use Passive FTP (for firewall and DSL modem compatibility)**.
6. Select **OK**.

If you select the **Enable folder view for FTP sites**, MSIE behaves as a Standard (PORT) mode FTP client even if you also select **Use Passive FTP**. If you clear the **Enable folder view for FTP sites** check box and then select the **Use Passive FTP** check box, MSIE behaves as a Passive (PASV) mode FTP client. By default, both MSIE and Secure FTP Server use Standard or Port mode.

Standard (PORT) mode FTP clients first establish a connection to TCP port 21 on the FTP server. This connection establishes the FTP command channel. The client sends a PORT command over the FTP command channel when the FTP client needs to receive data, such as a folder list or file. The PORT command contains information about which port the FTP client receives the data on. In Standard (PORT) mode, the FTP server always sends data from TCP port 20. The FTP server must open a new connection to the client when it sends data.

Passive (PASV) mode FTP clients also start by establishing a connection to TCP port 21 on the FTP server to create the control channel. When the client sends a PASV command over the command channel, the FTP server opens an ephemeral port (between 1024 and 5000) and informs the FTP client to request data transfer from that port. The FTP server responds to the request by using the ephemeral port as the source port for data transfer. If this occurs, the FTP server does not have to establish a new inbound connection to the FTP client.

Firewall configuration

Many firewalls do not accept new connections through an external interface. The firewall may detect these connections as unsolicited connection attempts and, therefore, drop them. Standard mode FTP clients do not work in this environment because the FTP server must make a new connection request to the FTP client.

Firewall administrators may sometimes not want to use Passive (PASV) mode FTP servers because the FTP server can open any ephemeral port number. Although Secure FTP Server by default uses the default ephemeral port range of 1024 through 5000, many FTP servers are configured with an ephemeral port range of 1024 through 65535. Firewall configurations that allow full access to all ephemeral ports for unsolicited connections may sometimes be considered unsecured.

Internet Explorer warning on Windows 2003

When you first install and start Secure FTP Server, you may see an Internet Explorer Warning about security. Select **Add** to allow Internet Explorer to recognize content from www.globalscape.com as safe content.

The message appears because your Internet Explorer settings are set at **High Security**. The trial version of Secure FTP Server attempts to open a page from the Internet to update you on the status of the trial.

After you register the full version of Secure FTP Server, you can change your Internet Explorer security settings and remove globalscape.com from your list of trusted sites.

This issue occurs most often in new installations of Windows 2003 Server.

Unable to import SSH keys

If SSH keys fail to import then with no error message then make sure you are importing the public opposed to the private key. Normally public are named *.pub.

WS_FTP 7.x. will not connect using SSL

WS_FTP 7.x (tested on 7.5, 7.6) does not adhere to RFC 2228 when establishing SSL connections. It authenticates properly, however it does not apply the correct PBSZ and PROT sequence necessary to protect the data channel and subsequently breaks the connection when the server replies with a clear-text directory listing.

RFC-2228 (http://www.ietf.org/rfc/rfc2228.txt) explicitly states that the data connection will be in the clear if no PROT argument is provided. >From RFC 2228, section 3. "The default protection level if no other level is specified is Clear. The Clear protection level indicates that the data channel will carry the raw data of the file transfer, with no security applied."

Another point taken from http://www.ford-hutchinson.com/~fh-1-pfh/ftps-ext.html#bad: "Once the control connection is secured, the state of the data connection is implicitly secure. This approach is in direct disagreement with [RFC-2228] which requires the PROT command to be issued and so should not be used in new implementations."

Solution:

Use a WS_FTP version 8 or a client such as CuteFTP 6 Home or Professional which properly authenticate using the AUTH/PBSZ/PROT sequence and also support newer TLS (SSL v3.1) mechanisms.

Workaround:
There is a workaround where you can cause WS_FTP to issue the proper sequence upon connect. In the Site Options for the particular site, go to the Startup tab and enter "PBSZ 0;PROT P" (sans quotes) in the Initialize

Command box. You should now be able to connect and establish a secure data channel using WS_FTP 7.x.

Calling methods in ICIClientSettings

Q. I’m trying to use the SetEnableAccount or similar methods in the ICIClientSettings interface. How do you obtain the user or settings level so that the various methods under ICIClientSettings interface can be applied?

A. You need to obtain a handle to the specific user or settings level from ICISite's GetUserSettings method or ICISite's GetSettingsLevelSettings method before you can perform an action upon that user or settings level. Here is a sample code snippet using PHP that demonstrates this technique:

```
<?php
// first create server object
$Server = new COM("SFTPCOMInterface.CIServer") or
die("Unable to
instantiate Server");
// connect to server
$Server->Connect("localhost",1000,"admin","admin");
// get handle to list of sites
$Sites = $Server->Sites();
// chose your site. On most one-site systems this will be
"0"
$MySite = $Sites->Item(0);
// Pull the settings for the user that you want.
$Settings = $MySite->GetUserSettings("juser");
// enable or disable or inherit.
// Enable is 1. Disable is 0. Inherit is -2
// Notice that you apply this to the settings for the user
that you just pulled
$Settings->SetEnableAccount(1);
// Be sure to apply the settings or else nothing will
really change
$Server->ApplyChanges();
// close the connection.
$Server->Close();
?>
```

Troubleshooting NAT firewalls

Most NAT setups allow outbound connection initializations without any problems but if you need to configure outbound connections manually, refer to the chart below to see which ports should be allowed for *outbound* communications on the client side. The server side configuration is much more important in most cases.

For FTPS (SSL), the connection port and the PASV ports need to be forwarded. For SFTP (SSH2), there's no such thing as PASV mode, so only the connection port needs to be forwarded. The following chart shows what ports should be forwarded to the server, assuming the default connection ports are being used. Please keep in mind that these ports need only be opened for *inbound* connections on the server side. You do not have to allow connections to be initialized outbound on the server side, as the connections will always be initiated from the client to the server.

Explicit SSL: 21, PASV range

Implicit SSL: 990, PASV range

SFTPS (SSH2): 22

The PASV range, which consists of ports 28000 to 30000 by default, can be defined in the Site Options tab, which is made accessible by clicking on the site in the tree view on the left. If you feel you need to restrict that range, the general rule of thumb is to take the maximum number of concurrent connections you think you're going to experience, and then add a third of that number to itself to get the total number of ports that should be open for smooth operation. NOTE: since the server will be behind NAT, you MUST specify the valid, external IP address (behind which the Server resides) in the PASV Mode Options.

Importing Secure FTP Server key pairs in CuteFTP Professional

CuteFTP Professional does not support the key pair format generated by GlobalSCAPE Secure FTP Server.

GlobalSCAPE SecureFTP Server creates SSH key pairs in the OpenSSH format. It can also accept (import) keys in the other popular format, typically referred to as the ssh.com format. On the other hand, CuteFTP Professional can only create and import keys pairs in the ssh.com format. If you create an OpenSSH style key pair in Secure Server and attempt to import the key pair into the client, it will fail.

From a business processes standpoint, you shouldn't user the server's key pair in the client anyway, as doing so would involve sending the client the public and private key, creating a potential security vulnerability. It is also an atypical way of setting up public key authentication for one or more clients. The common practice is to create the key pair in each client and subsequently make the client's public key available to the server administrator, who in turn should import the client's public key into the server's trusted list.

If you are using CuteFTP but not our server, and the server is running OpenSSH, you will need to convert the ssh.com style public key generated by CuteFTP prior to importing it into your server's key directory.

Conversion

To convert an ssh.com key to an OpenSSH key, perform the following steps using the ssh-keygen utility:

1. Create a key pair in the client.
2. Send the public key (Identity.pub) to the server administrator (via FTP, e-mail, etc. No need to secure the transmission of the public key)
3. The server administrator must then convert the public key to OpenSSH, check it with wordcount, and add it to authorized_keys.

Command sequence:

```
ssh-keygen -i -f Identity.pub > sshpub
wc sshpub
cat sshpub > ~\.ssh\authorized_keys
```

Note:
Wordcount should return a "1" as the first number. OpenSSH is going to ask for the identity file's password the first time you log in. If CuteFTP fails to connect, please contact our support team and provide the kernel version, OpenSSH build, and the CuteFTP Professional build number (located under **Help > About**).

The system could not find the environment option that was entered

If your Secure FTP Server service will not start and it generates this error message: "The system could not find the environment option that was entered", then make sure:

- The system account the Server service runs under has full access to the Server executable (cftpses.exe).
- You are launching the service from the NT Services applet, not from the command prompt.

Server will not run on Windows XP

Some versions of Windows XP have an built-in Internet Firewall that blocks FTP traffic. This firewall is active by default.

To turn off the Windows XP firewall

Follow the instructions Microsoft has posted on their Web site at http://www.microsoft.com/WINDOWSXP/home/using/howto/homenet/icf.asp.

Changing the log file format kicks users off the server

Any active users are kicked off the server whenever the log file format is changed. It is recommended that the log file format is selected as part of the initial configuration before you start any sites.

System cannot find the environment option that was entered

If Secure FTP Server will not run and generates the error message "The system could not find the environment option that was entered," verify that the system has full access to the Secure

FTP Server executable (cftpses.exe). Also, make sure you are launching the service from the NT Services applet, not from the command prompt.

13

Configuration Tips

How to Configure Automatic Email Notification After an Upload

An FTP server connected to the Internet risks attack by malicious users who might attempt to compromise the server. This means file transfers might be subject to corruption. These instructions extend the capability of GlobalSCAPE Secure FTP Server to permit a user to validate the integrity of files uploaded to the server. These instructions describe how to use a PHP scripting technique coupled with a system event to provide a customized email to the upload user to validate the integrity of the files transferred.

- **Technology Used in These Directions**
  These instructions use PHP scripting techniques to extend Secure FTP Server's functionality. For this capability to operate correctly, PHP Version 4.3.3 (or above) must be installed on the same machine as Secure FTP Server.
- **Reminder**
  To send an email response to the upload user, the SMTP server specified in the PHP.ini file must work correctly and be accessible by the Secure FTP server.

Secure FTP Server has several features that can be customized to provide unique services. Validation of a complete and correct file transfer is one unique service that can be easily added to the program. A confirmation email of a successful upload can be returned to the user with a few simple steps to modify the program. The process is implemented using a custom command from Secure FTP Server in conjunction with a PHP script. The script uses the input parameters of the command to have an email sent to the initiator of the upload.

The script is launched from an **event rule** that uses the contextual information of the completed transfer to determine the email address where the confirmation email will be sent. The confirmation email will include details of the transfer such as time transferred, size of the file, and MD5 value. (MD5 (**M**essage **D**igest **5**) is a popular one-way hash function used to create a message digest for digital signatures.)

Server Configuration and Insertion of the PHP Script

1. Configure the Intranet FTP server to allow incoming client connections from the publicly accessible FTP server. Create a user account and configure user's email address.

2. Copy the **uploaded.php** file to a location on the publicly accessible server outside the VFS. Provide appropriate permissions so the service account running Secure FTP Server can read and execute the script.

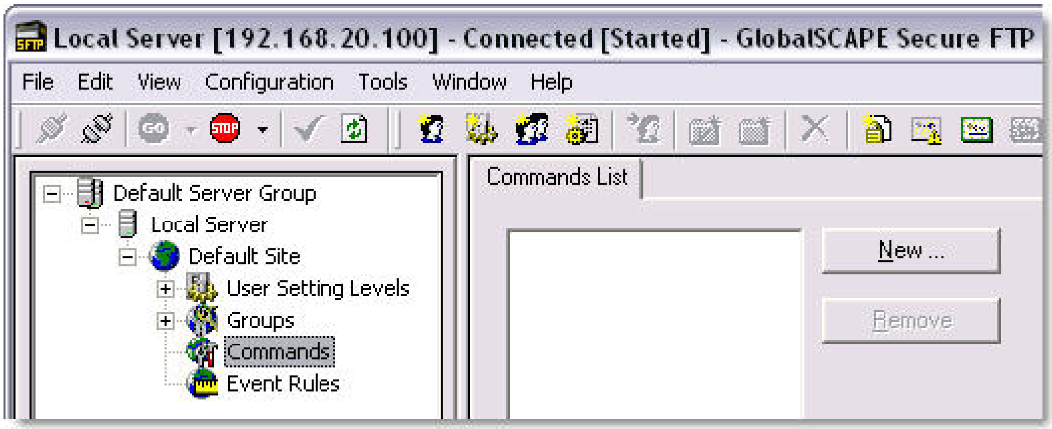
Script Source: uploaded.php

```
<?php
/*==================================================================
=========
*= File: uploaded.php
*= Created: 07/20/2004 WHT
*= Purpose:
*= Send email to the user who upload a file to the Secure FTP
Server.
*= The email tell user uploading time, file size and MD5 value of
the file.
*= NOTES:
*= Event Rule: On File Upload
*= Command Paramaters:
*= -f uploaded.php "%EVENT.TIME%" "%FS.VIRTUAL_PATH%"
"%FS.FILE_NAME%" "%FS.PATH%" "%USER.LOGIN%" "%USER.EMAIL%"
*=
*= USAGE:
*= php -f uploaded.php <time> <virtual path> <file name> <physical
path> <user name> <user email>
*= EXAMPLE:
*= php -f uploaded.php "20 Jul 04 14:44:08" "../Pub/log.cpp"
"log.cpp" "D:/Pub/log.cpp" "test" "user@mail.com"
*= NOTE
*= REQUIRES PHP 4.3.3 or above version
*=
*====================================================================
========
*/

//Check if the email address is set
if (($argc != 7) || empty($argv[6]))
exit;

//Get the virtual path
$vpath = str_replace($argv[3], "" , $argv[2]);

$title = "The user \"" . $argv[5] . "\" has uploaded " . $argv[3] .
" into " . $vpath;

//Get file size and md5 value
$file_size = @filesize($argv[4]);
$md5_value = @md5_file($argv[4]);
//Make the email body
$body = "This message was sent to you automatically by GlobalSCAPE
Secure FTP Server 2.0.\n";
$body .= $title . " on ". $argv[1]. ".\nThe file size is ".
$file_size . " bytes, MD5 value is " . $md5_value . ".\n";

//Send email to the person who upload this file.
@mail($argv[6], $title, $body);
exit;
?>
```

Part Two: Create a Custom Command to Launch the PHP Script

Create a custom command by following the regular process shown below.

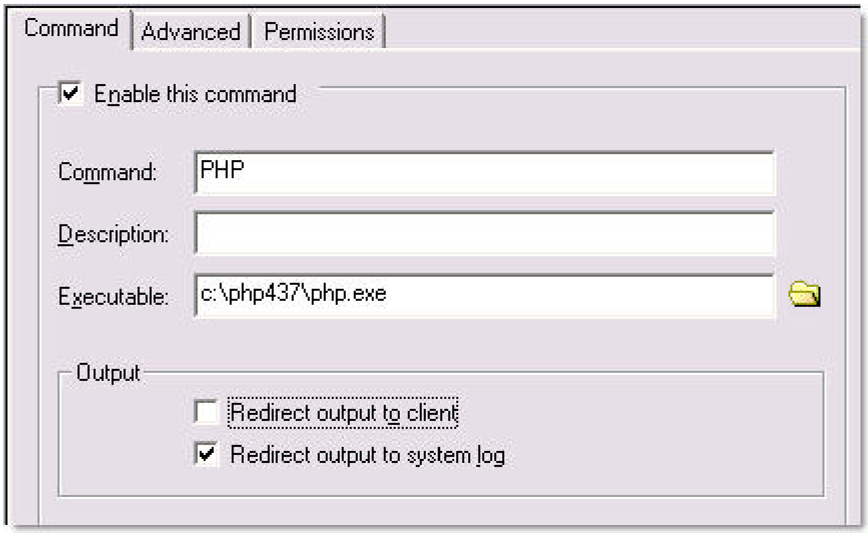
1. Launch the Secure FTP Administrator on the Server machine and connect to the local Secure FTP Server.
2. Expand the **Local Server** tree to view the **Commands** icon.
3. Select the **Commands** icon to display the **Commands List** window.
4. Press the **New** button to add a new command.


Local Server [192.168.20.100] - Connected [Started] - GlobalSCAPE Secure FTP
File Edit View Configuration Tools Window Help
Default Server Group
Local Server
Default Site
User Setting Levels
Groups
Commands
Event Rules
Commands List
New ...
Remove

The **Command** window displays.

5. Enter **PHP** in the **Command** text box.
6. Enter the location of the PHP script command line in the **Executable** text box.
7. Uncheck the **Redirect output to client** box.

In this example the executable line reads: **C:\php437\php.exe**



☑ Enable this command

Command: PHP

Description:

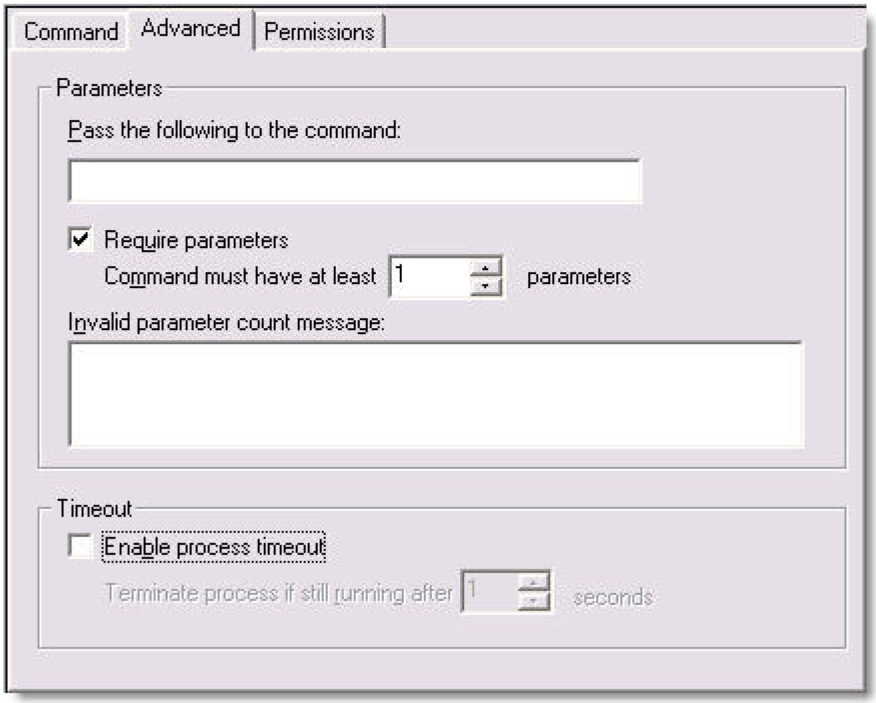
Executable: c:\php437\php.exe

Output

☐ Redirect output to client

☑ Redirect output to system log

8. Select the **Advanced** tab on the right side of the window to configure the parameters.
9. Place a check in the box **Require parameters** check box.
10. Make sure there is a "**1**" in the **Command must have at least ____ parameters** box.



Parameters

Pass the following to the command:

☑ Require parameters

Command must have at least 1 parameters

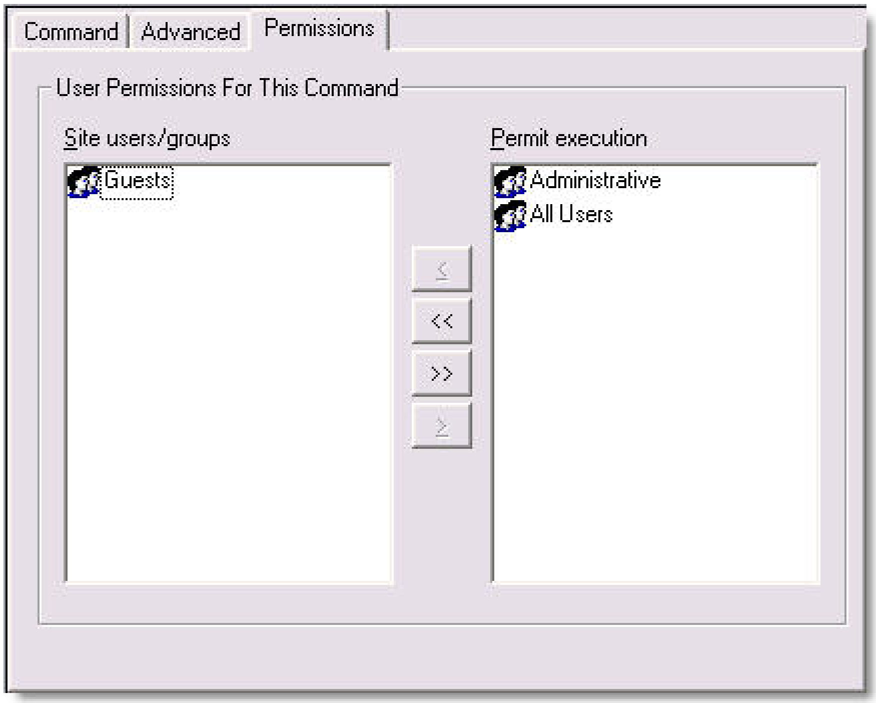
Invalid parameter count message:

Timeout

□ Enable process timeout

Terminate process if still running after 1 seconds

11. Select the **Permissions** tab to set necessary permissions for proper users to execute the encryption command.
12. Click **Apply** to set the rule and **Permissions**.

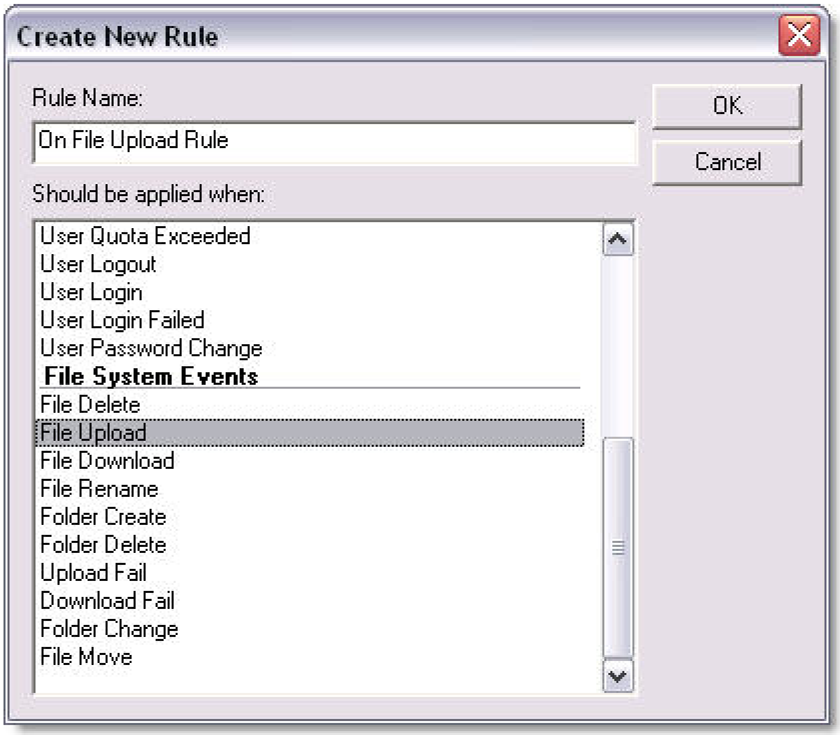


User Permissions For This Command

Site users/groups: Guests

Permit execution: Administrative, All Users

The Custom Command is complete.



Create New Rule

Rule Name:

On File Upload Rule

Should be applied when:

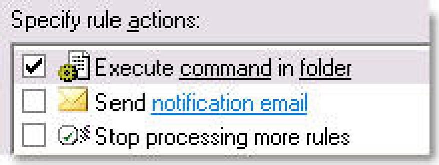
User Quota Exceeded
User Logout
User Login
User Login Failed
User Password Change
File System Events
File Delete
File Upload
File Download
File Rename
Folder Create
Folder Delete
Upload Fail
Download Fail
Folder Change
File Move

Part Three: Specify an Event Rule to Trigger the Command

Event Configuration Steps:

1. Start the Secure FTP Administrator on the Server machine and connect to the local Secure FTP Server.
2. Select the site where the event will take place.
3. **Choose Create New Event Rule** from the **Configuration** drop down menu (or right-click on the Site node in the tree view and choose **New Event Rule**).
4. Enter the name of the new rule in the **Rule Name** text box.
5. In the **Should be applied when:** section, double click the **File Upload** entry under the **File System Events** category.

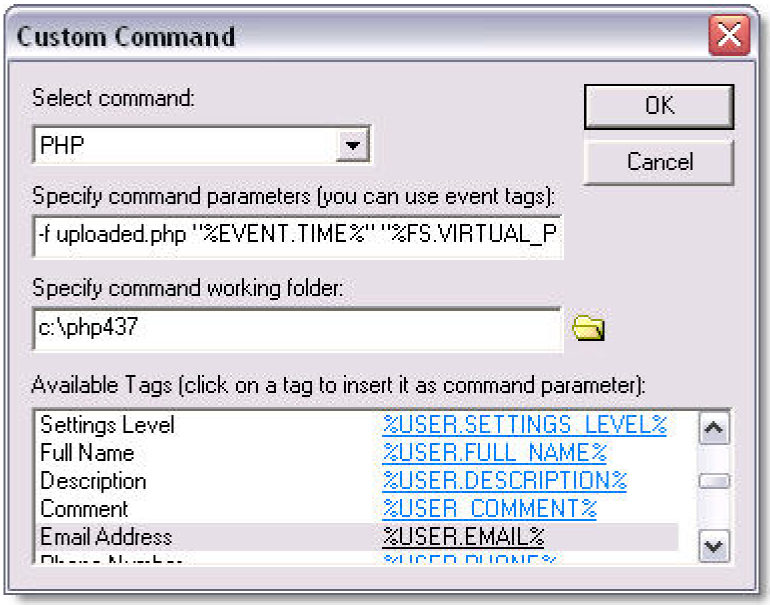
   This displays the **Event Rule** screen.



Specify rule actions:

☑ Execute command in folder
□ Send notification email
□ Stop processing more rules

6. On the Event Rule screen that appears, place a checkmark next to the **Execute command in folder** checkbox in the **Specify rule actions** area.



Select command: PHP

Specify command parameters (you can use event tags): -f uploaded.php "%EVENT.TIME%" "%FS.VIRTUAL_P

Specify command working folder: c:\php437

Available Tags (click on a tag to insert it as command parameter):

Settings Level	%USER.SETTINGS_LEVEL%
Full Name	%USER.FULL_NAME%
Description	%USER.DESCRIPTION%
Comment	%USER_COMMENT%
Email Address	%USER.EMAIL%

7. In the **Specify rule condition and action parameters** section on the Event Rule screen, click on the **'PHP -f uploaded.php "%EVENT.TIME%" "%FS.VIRTUAL_PATH%" "%FS.FILE_NAME%" "%FS.PATH%" "%USER.LOGIN%" "%USER.EMAIL%"'** hyperlink. The Custom Command window will appear.
   a. In the **Custom Command** window, select the **PHP** command from the choices in the **Select command** drop down menu.
   b. In the **Specify command parameters** text box there should be seven parameters.

```
 -f uploaded.php
"%EVENT.TIME%"
"%FS.VIRTUAL_PATH%"
"%FS.FILE_NAME%"
"%FS.PATH%"
"%USER.LOGIN%"
"%USER.EMAIL%"
```

Note:
The quote marks " " surrounding the parameter must be included.

   c. For the working folder, specify the location of the custom command's executable. In this case it is: **C:\php437**
   d. Click **OK** to save the entries and close the window.

EFT Server is now ready to provide a verification email upon every upload event. The email will inform the user of when upload occurred, file size, and MD5 value of the file.

How to sign and encrypt log files

These instructions show how to encrypt log files to safeguard information in compliance with established and emerging information technology security standards. Secure FTP Server creates a log entry for all user authentication and transfer events. To ensure the log files are safe from tampering, and comply with the need for additional security, these log (audit) files need to be encrypted. Encryption is performed using a secret key that permits only authorized users to access or change the log file.

The following instructions show how to encrypt log files using **event rules** and external utilities. The process to encrypt a log file uses an **On Log Rotation** event to trigger a Windows Scripting Host script that activates a .NET Web Service to perform the encryption.

Technologies Used in These Directions

These directions use several different techniques to extend the functionality of Secure FTP Server with built-in capabilities of the Microsoft Windows operating system. Specifically, this example relies upon the following utility applications, all of which must be installed and configured in order to operate the encryption service:

- Microsoft .NET Framework on a server running IIS (this could be the same machine as the Secure FTP Server, or any other machine to which the Secure FTP Server computer has HTTP access)
- Windows Scripting Host on the Secure FTP Server machine
- Microsoft Soap Toolkit 3.0 on the Secure FTP Server machine

Note:
Web Services provide functionality by issuing requests over the HTTP protocol. This means that the Web Service used may reside on the same computer as the Secure FTP Server or on a different machine entirely. The only requirement is the Secure FTP Server must access the Web Service machine using port 80.

Log File Encryption Using a Web Service

Secure FTP Server can rotate logs on a daily, weekly, or monthly basis. Selection of the rotation period is based on administrator preference and the amount of server traffic. Since an access log file typically grows 1 MB or more per 10,000 requests, even a moderately busy server will generate large log files.

When the log rotation event occurs the encryption process begins for logs created during the period. An **On Log Rotate** event triggers a **custom site command** that opens a Windows Scripting Host (WSH) file. The file contains a script (Appendix A) that uses the COM interface of the Microsoft Soap Toolkit to specify a parameter that encrypts the log file using a password that is also passed as a parameter. The event rule contains context information that specifies the file to be encrypted. Decryption of the file can be accomplished with the appropriate password using a separate script (Appendix B) and Web Service.

The custom site command will also hard-code the password for encryption. Hard coding is the option used as the example in these instructions. The script could be changed to substitute a parameter indicating a different password. Using a parameter instead of a static password allows greater flexibility in selecting the basis of different types of passwords and enhances security.

Configuration of the Crypto Web Service

The presence of a Web Service that exposes encrypt and decrypt functions is essential for the encryption process. To install this Web Service, the machine must be running IIS with the .NET Framework installed. When these requirements are met, follow these steps to install the service:

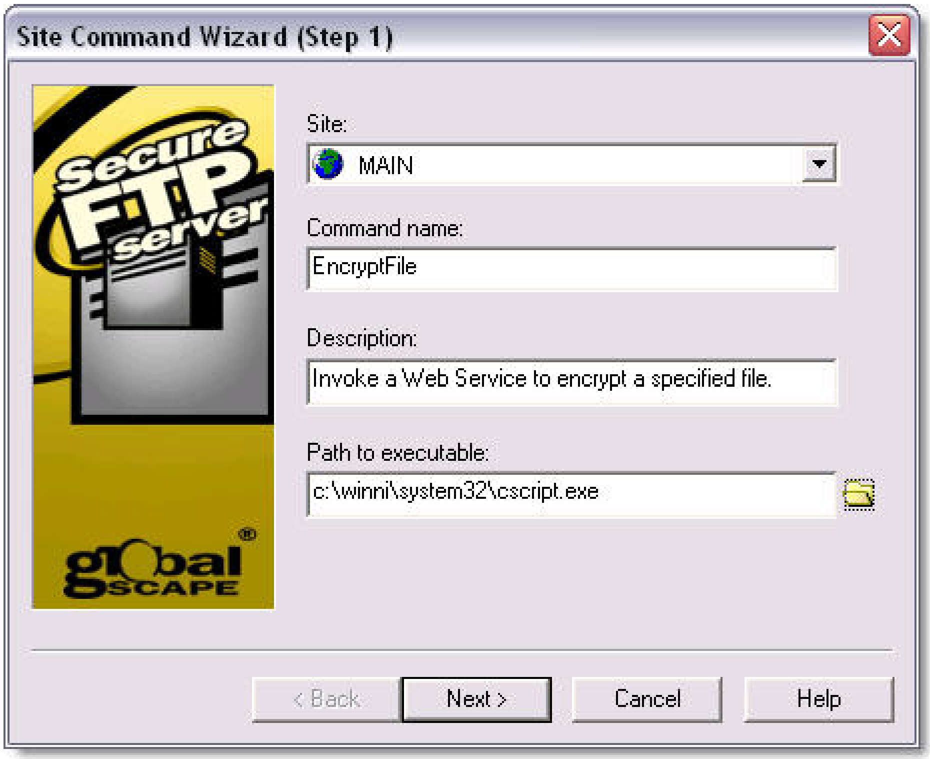
1. Extract the contents of **crypto.zip** to a new folder on that machine, e.g. **c:\inetpub\wwwroot\crypto**
2. Open the IIS administrator and add a new Virtual Site to your Default Web Site.
    a. Right-click on the **Default Site** and choose **New > Virtual Directory**.
    b. When prompted by the wizard, specify an alias of **crypto** and set the Directory to the folder created in step 2a.
    c. Put checkmarks next to **Read, Run Scripts,** and **Execute**.
3. Test the Web Service by opening up a browser and navigating to the URL for your IIS machine followed by **/crypto/crypto.asmx**, for example: **http://192.168.20.12/crypto/crypto.asmx**. This should produce a .NET Web Service HTML page that describes the "CryptoService" Web Service and its two operations, **Encrypt** and **Decrypt**.

Custom Command Configuration for Secure FTP Server Log Encryption

The GlobalSCAPE Secure FTP Server needs to be configured to start the **encryptfile.js** script (Appendix A) when log file rotation occurs. These configuration directions create a **custom site command** on the Secure FTP Server machine. These are the steps to get it working:

1. Unzip the contents of **"Encryptfile.zip"** and put them in a folder on the Secure FTP Server, e.g., c:\program files\globalscape\Secure FTP Server\.
2. Open these files in a text editor (e.g., right-click on the file and choose Edit With . Notepad)
3. Modify the URL's for the Web Service so the address matches the server configured with the **crypto** Web Service. (This configuration was done in the previous sequence, Configuration of the Crypto Web Service.) The URL to modify is found on the line which reads

    SoapClient.mssoapinit http://localhost/crypto/crypto.asmx?wsdl
4. Launch the Secure FTP Server GUI Administrator on the Server machine and connect to the local Secure FTP Server.
5. Click on the site for which you want log files encrypted and choose "Create New Command" from the configuration menu (or press CTRL+M on the keyboard). The Site Command Wizard will appear.
6. Enter **EncryptFile** in the **Command name:** text box.
7. Specify **escript.exe** in the **Path to executable:** text box as the MicroSoft Windows® command-line Scripting Host.



Site: MAIN

Command name: EncryptFile

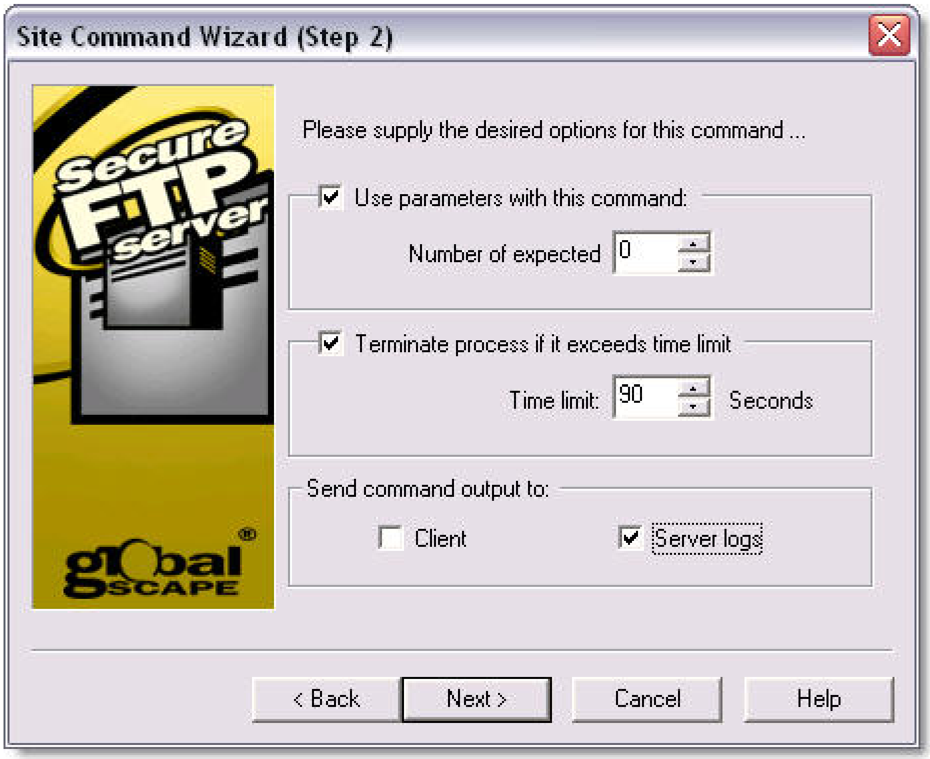
Description: Invoke a Web Service to encrypt a specified file.

Path to executable: c:\winnt\system32\cscript.exe

When this is entered, click **Next** to continue.

8. On the Site Command Wizard (Step 2) page:
    a. Place a check the **Use parameters** box and specify that one parameter is needed. (This will be the name of the file to encrypt. The password for the encryption will be hardcoded in the next step.)

       Alternatively, the command could specify two parameters, the second of which is the password. This would be a more extensible solution where the Event Rule could be customized to determine the password based upon the month of rotation, or the IP address of the server, or any number of other possibilities. In this example, the password will simply be hardcoded.
    b. Place a check in **Terminate process** box to prevent a 'runaway process.' The example shows that Secure FTP Server will terminate this command if it takes over 90 seconds to complete.
    c. Check the **Server logs** box so the output will be sent to the log file so the success/failure of the command can be reviewed at a later date. When this is done, click **Next** to continue.



Site Command Wizard (Step 2)

Please supply the desired options for this command ...

☑ Use parameters with this command:

Number of expected 0

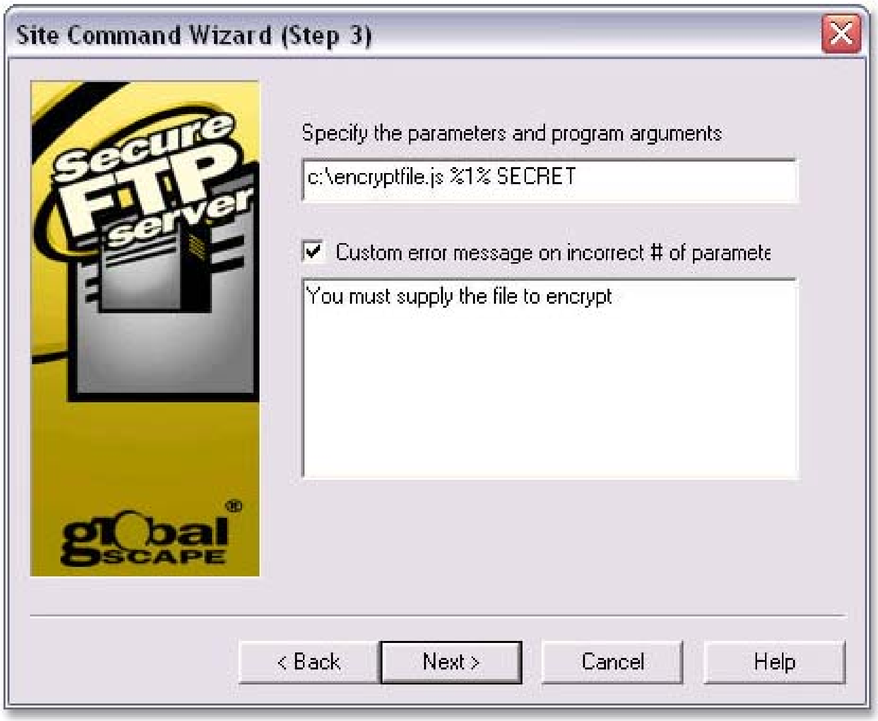
☑ Terminate process if it exceeds time limit

Time limit: 90 Seconds

Send command output to:

□ Client ☑ Server logs

9. In Step 3 of the Wizard, the parameters to the CSCRIPT executable are specified.
    a. In the **Specify the parameters and program arguments** text box enter the path to the **encryptfile.js** file. (This file was extracted in step #1.)
    b. The next parameter is the name of the file to encrypt. Leave this as a PARAMETER value.
    c. The last parameter is the password, which is hardcoded as "SECRET" in the example.
    d. The example also shows that a custom error message will be displayed in case someone invokes this command without supplying a file name parameter.



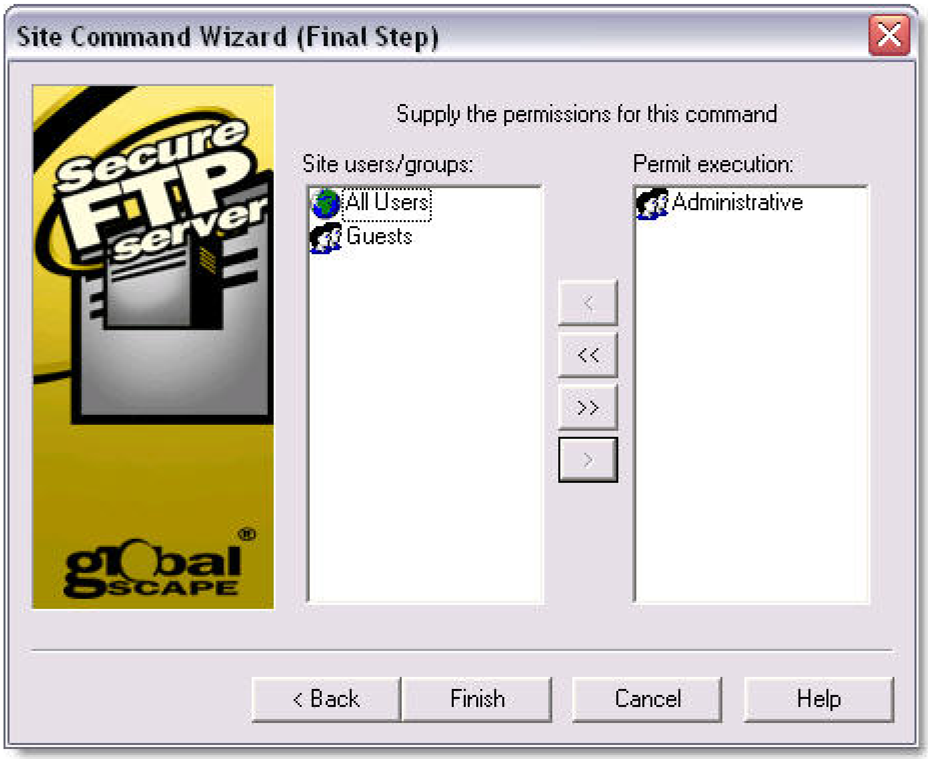
Specify the parameters and program arguments

c:\encryptfile.js %1% SECRET

☑ Custom error message on incorrect # of paramete

You must supply the file to encrypt

10. The last screen of the command setup wizard sets permissions. The example indicates that only the Administrative group can access this custom command. This restriction will prevent a typical FTP user from invoking this function.



11. If desired, this custom command can be verified by logging into the server as an **Administrative** user and issuing the raw command "SITE EncryptFile <filename>".

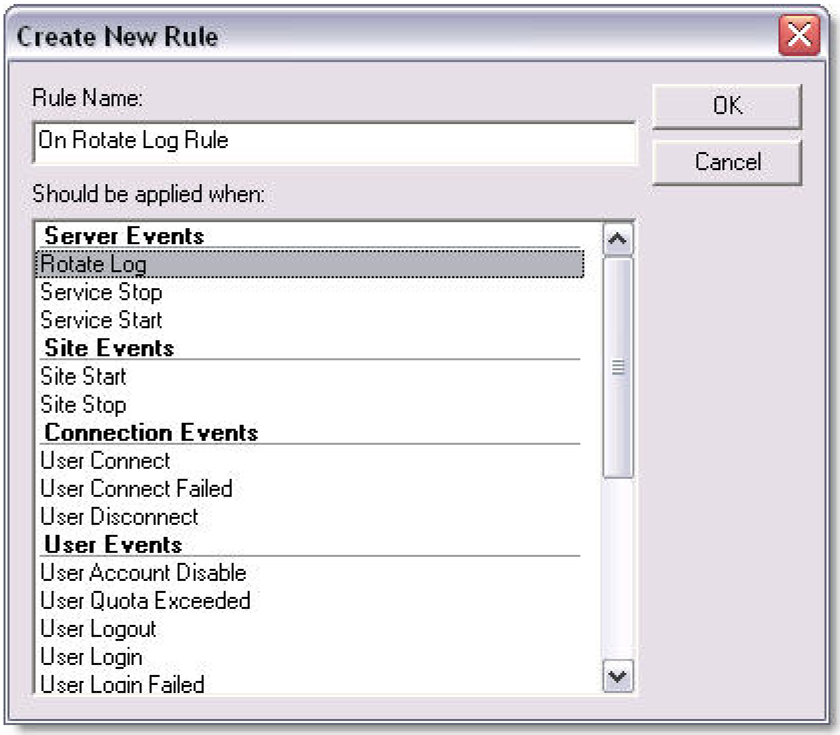
Event Rule Configuration on Secure FTP Server for Log Encryption Upon Rotation

Event rule configuration is the final step in the process to allow GlobalSCAPE Secure FTP Server to automatically encrypt log files when they are rotated. The first two steps, enabling the crypto web service and creating the **EncryptFile** Custom Command, prepared access to the encryption service and created the command to perform the encryption. This last step configures the trigger to launch the command.

These configuration steps for the **Event Rule** also include a notification provision that sends an email to the System Administrator indicating the encryption has taken place.

Event Configuration Steps:

1. Launch the Secure FTP Server GUI Administrator on the Server machine and connect to the local Secure FTP Server.
2. Select the site where the log files are to be encrypted.
3. Choose **Create New Event Rule** from the menu (or right-click on the Site node in the treeview and choose **New Event Rule**). Selecting this option opens a New Rule wizard.
4. Enter the name of the new rule in the **Rule Name** text box.
5. In the next section, select the **Rotate Log** entry under the **Server Events** category.
6. Press **OK** to continue.

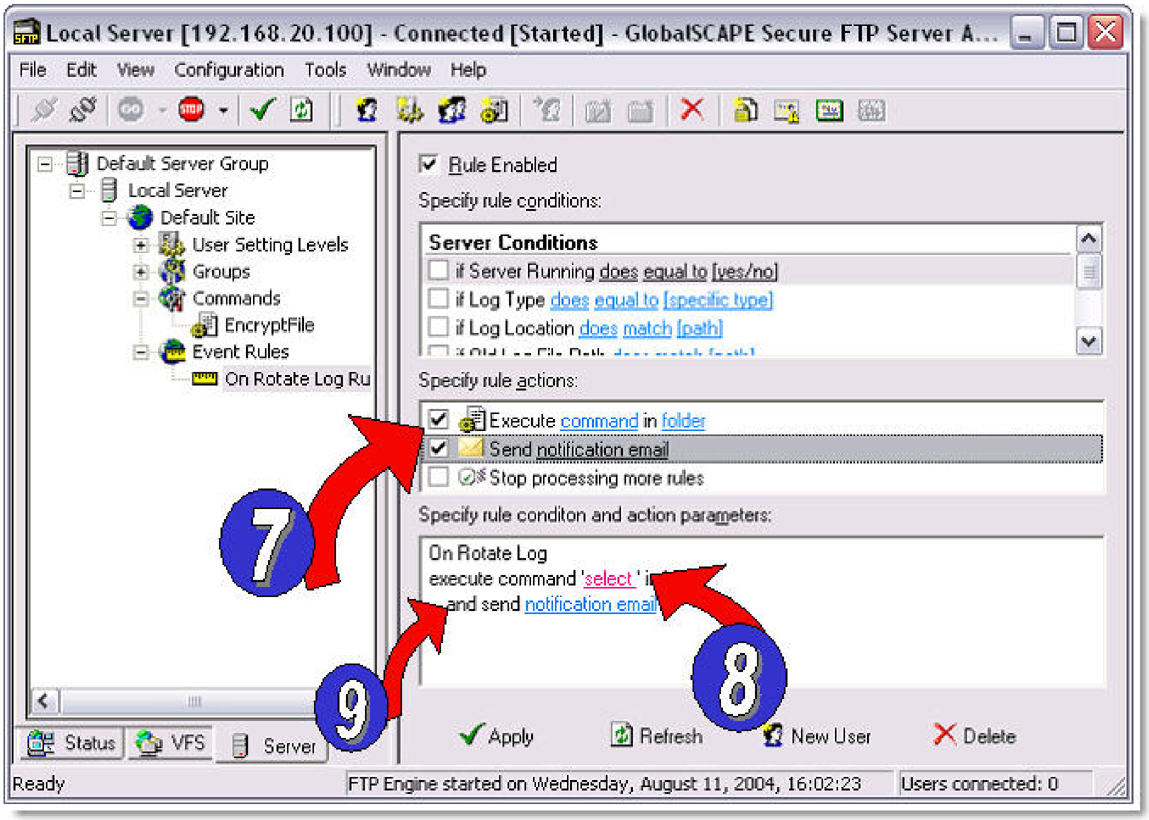


Rule Name: On Rotate Log Rule

Should be applied when:

Server Events
Rotate Log
Service Stop
Service Start
Site Events
Site Start
Site Stop
Connection Events
User Connect
User Connect Failed
User Disconnect
User Events
User Account Disable
User Quota Exceeded
User Logout
User Login
User Login Failed

7. On the Event Rule screen that appears, place a checkmark next to both **Execute command in folder** and **Send notification email** under the **Specify rule actions** area.



☑ Rule Enabled

Specify rule conditions:

Server Conditions
☐ if Server Running does equal to [yes/no]
☐ if Log Type does equal to [specific type]
☐ if Log Location does match [path]

Specify rule actions:

☑ Execute command in folder
☑ Send notification email
☐ Stop processing more rules

Specify rule conditon and action parameters:

On Rotate Log
execute command 'select' i
and send notification email

Ready | FTP Engine started on Wednesday, August 11, 2004, 16:02:23 | Users connected: 0

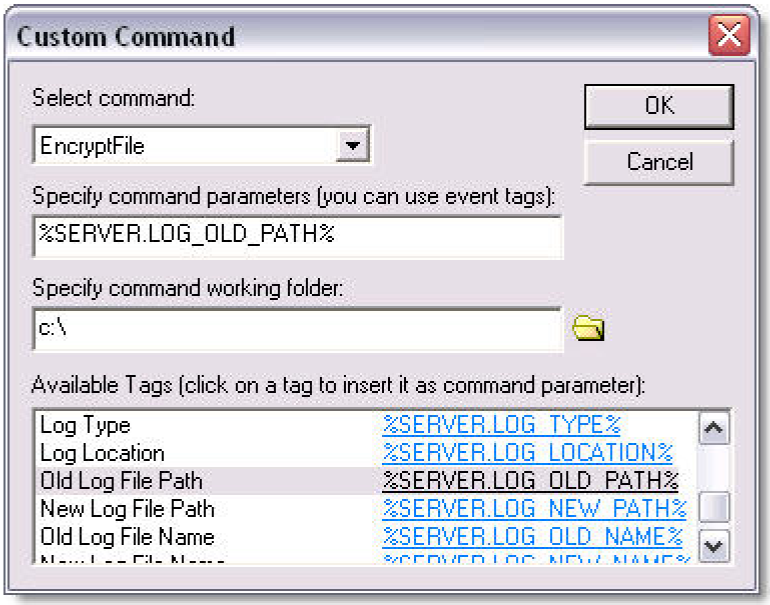
In the **Specify rule condition and action parameters** section, click on the hyperlink select.

a. In the **Custom Command** window that appears, select the **EncryptFile** command from the choices in the **Select command** drop down menu.

b. In the **Specify command parameters** text box enter the context variable for **old log file path.**

c. For the working folder, select the same folder that houses the **encryptfile.js** script.

d. Click **OK** to save the entries and close the window.



Select command: EncryptFile

Specify command parameters (you can use event tags): %SERVER.LOG_OLD_PATH%

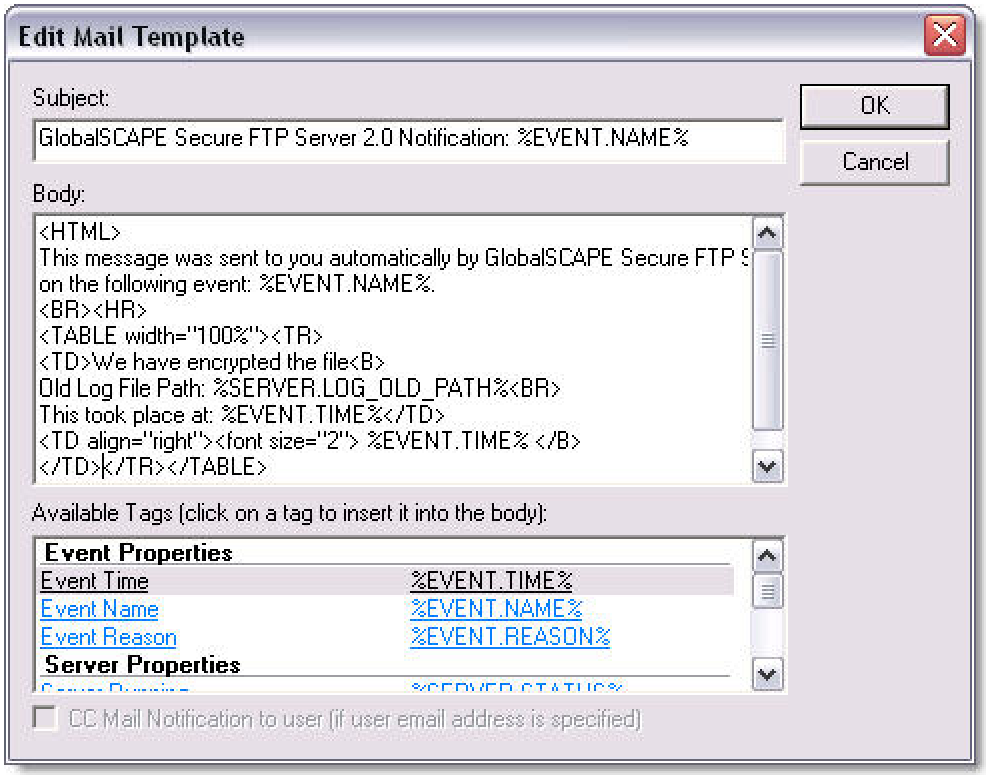
Specify command working folder: c:\

Available Tags (click on a tag to insert it as command parameter):

Log Type	%SERVER.LOG_TYPE%
Log Location	%SERVER.LOG_LOCATION%
Old Log File Path	%SERVER.LOG_OLD_PATH%
New Log File Path	%SERVER.LOG_NEW_PATH%
Old Log File Name	%SERVER.LOG_OLD_NAME%
New Log File Name	%SERVER.LOG_NEW_NAME%

Back on the Event Rule screen, in the **Specify rule condition and action parameters** section click the **notification email** hyperlink.

1. In the **Edit Mail Template** window update the email text to give descriptive information about the event that took place, such as the file being encrypted and the time at which the encryption took place.
2. Click **OK** to save the entries and close the window.



Subject:

GlobalSCAPE Secure FTP Server 2.0 Notification: %EVENT.NAME%

Body:

```
<HTML>
This message was sent to you automatically by GlobalSCAPE Secure FTP S
on the following event: %EVENT.NAME%.
<BR><HR>
<TABLE width="100%"><TR>
<TD>We have encrypted the file<B>
Old Log File Path: %SERVER.LOG_OLD_PATH%<BR>
This took place at: %EVENT.TIME%</TD>
<TD align="right"><font size="2"> %EVENT.TIME% </B>
</TD></TR></TABLE>
```

Available Tags (click on a tag to insert it into the body):

Event Properties	
Event Time	%EVENT.TIME%
Event Name	%EVENT.NAME%
Event Reason	%EVENT.REASON%
Server Properties	

□ CC Mail Notification to user (if user email address is specified)

The GlobalSCAPE Secure FTP Server is now ready to encrypt log files upon rotation.

Script Source: encryptfile.js

```
'
********************************************************************
*******************
' * FILE: encryptfile.vbs
' * CREATED: 19 July 2004 GTH
' * PURPOSE: Invoke the "crypto" web service to encrypt a file with
' * the supplied secret key (and, optionally, algorithm)
' * USAGE:
' * cscript encryptfile.vbs filename passphrase [algorithm]
' * WHERE:
' * "filename" is the fully qualified path to the file which is to
be encrypted
' * "passphrase" is the secret text that is the key for
encryption/decryption
' * "algorithm" is an optional parameter specifying cipher
algorithm; can be:
' * DES
' * 3DES
' * RC2
' * RIJNDAEL (default if no algorithm supplied on command line)
' *
' * NOTES
' * The file is encrypted "in-place"; that is, the old contents are
overwritten
```

```
' * by the encrypted contents.
' *
' * EXAMPLE:
' * cscript encryptfile.vbs "c:\inetpub\ftproot\main site\test.txt "
SECRET 3DES
'
********************************************************************
*******************

'
' Function encryptText
'
' Invoke a webservice for encrypting a given string using specified
passphrase and
' algorithm. Algorithm can be "" (blank) to use default cipher
RIJNDAEL.
'
Function encryptText(strText, strPassphrase, strAlgorithm)
Dim SoapClient
Dim strResult

set SoapClient = WScript.CreateObject("MSSOAP.SoapClient30")


SoapClient.ClientProperty("ServerHTTPRequest") = True
' NOTE: Change the URL below to your .NET web service location.
SoapClient.mssoapinit "http://localhost/crypto/crypto.asmx?wsdl"

strResult = SoapClient.Encrypt( strText, strPassphrase, strAlgorithm
)
encryptText = strResult
End Function


' MAIN
Dim oArgs
Dim strFileName, strPassPhrase, strAlgorithm
Dim oFSO, oFile

Dim strContents

Set oArgs = WScript.Arguments
If ( oArgs.length < 2 ) Then
WScript.Echo "Invalid usage!"
WScript.Echo "USAGE:"
WScript.Echo " encryptfile.vbs filename passphrase [algorithm]"
WScript.Quit 255
End If

strFileName = oArgs(0)
strPassphrase = oArgs(1)
If ( oArgs.length > 2 ) Then
strAlgorithm = UCase(oArgs(2))
End If

WScript.Echo "Going to encrypt '" & strFileName & "' with a
passphrase of '" & strPassPhrase & "', using algorithm '" &
```

```
strAlgorithm & "'"

Set oFSO = CreateObject("Scripting.FileSystemObject")
If not oFSO.FileExists( strFileName ) Then
WScript.Echo "Could not find file '" & strFileName & "'!"
WScript.Quit 255
End If

Set oFile = oFSO.OpenTextFile( strFileName, 1 ) ' 1 = for reading
strContents = oFile.ReadAll()
oFile.Close

strContents = encryptText( strContents, strPassphrase, strAlgorithm
)
WScript.Echo "New file contents: " & vbCrLf & strContents & vbCrLf
Set oFile = oFSO.OpenTextFile( strFileName, 2 ) ' 2 = for writing
oFile.Write(strContents)
oFile.Close

WScript.Echo "Finished encrypting '" & strFileName & "'"
Set oFile = nothing
Set oFSO = nothing
```

Script Source: Decryptfile.js

```
'
********************************************************************
*******************
' * FILE: encryptfile.vbs
' * CREATED: 19 July 2004 GTH
' * PURPOSE: Invoke the "crypto" web service to encrypt a file with
' * the supplied secret key (and, optionally, algorithm)
' * USAGE:
' * cscript decryptfile.vbs filename passphrase [algorithm]
' * WHERE:
' * "filename" is the fully qualified path to the file which is to
be encrypted
' * "passphrase" is the secret text that is the key for
encryption/decryption
' * "algorithm" is an optional parameter specifying cipher
algorithm; can be:
' * DES
' * 3DES
' * RC2
' * RIJNDAEL (default if no algorithm supplied on command line)
' *
' * NOTES
' * The file is decrypted "in-place"; that is, the old contents are
overwritten
' * by the decrypted contents.
' *
' * EXAMPLE:
' * cscript decryptfile.vbs "c:\inetpub\ftproot\main site\test.txt "
SECRET 3DES
```

```
'
********************************************************************
*******************

'
' Function decryptText
'
' Invoke a webservice for decrypting a given string using specified
passphrase and
' algorithm. Algorithm can be "" (blank) to use default cipher
RIJNDAEL.
'
Function decryptText(strText, strPassphrase, strAlgorithm)
Dim SoapClient
Dim strResult

set SoapClient = WScript.CreateObject("MSSOAP.SoapClient30")

SoapClient.ClientProperty("ServerHTTPRequest") = True
' NOTE: Change the URL below to your .NET web service location.
SoapClient.mssoapinit "http://localhost/crypto/crypto.asmx?wsdl"

strResult = SoapClient.Decrypt( strText, strPassphrase, strAlgorithm
)
decryptText = strResult
End Function


' MAIN
Dim oArgs
Dim strFileName, strPassPhrase, strAlgorithm
Dim oFSO, oFile
Dim strContents

Set oArgs = WScript.Arguments
If ( oArgs.length < 2 ) Then
WScript.Echo "Invalid usage!"
WScript.Echo "USAGE:"
WScript.Echo " encryptfile.vbs filename passphrase [algorithm]"
WScript.Quit 255
End If

strFileName = oArgs(0)
strPassphrase = oArgs(1)
If ( oArgs.length > 2 ) Then
strAlgorithm = UCase(oArgs(2))
End If

WScript.Echo "Going to decrypt '" & strFileName & "' with a
passphrase of '" & strPassPhrase & "', using algorithm '" &
strAlgorithm & "'"

Set oFSO = CreateObject("Scripting.FileSystemObject")
If not oFSO.FileExists( strFileName ) Then
WScript.Echo "Could not find file '" & strFileName & "'!"
WScript.Quit 255
End If
```

```
Set oFile = oFSO.OpenTextFile( strFileName, 1 ) ' 1 = for reading
strContents = oFile.ReadAll()
oFile.Close

strContents = decryptText( strContents, strPassphrase, strAlgorithm
)
WScript.Echo "New file contents: " & vbCrLf & strContents & vbCrLf
Set oFile = oFSO.OpenTextFile( strFileName, 2 ) ' 2 = for writing
oFile.Write(strContents)
oFile.Close

WScript.Echo "Finished decrypting '" & strFileName & "'"
Set oFile = nothing
Set oFSO = nothing
```

Script Source: crypto.asmx

```
<%@ WebService Language="c#" Codebehind="crypto.asmx.cs"
Class="crypto.CryptoService" %>
```

Script Source: crypto.asmx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Web;
using System.Web.Services;
using System.Security.Cryptography;
using System.Text;
using System.IO;

namespace crypto
{
/// <summary>
/// Summary description for Service1.
/// </summary>
public class CryptoService : System.Web.Services.WebService
{

// private members
string txtIV = "";
byte[] plainText;
byte[] cipherText;
int iKeySize;
byte[] keyBytes;
byte[] initVector;
byte[] hashText;


public CryptoService()
```

```
{
//CODEGEN: This call is required by the ASP.NET Web Services
Designer
InitializeComponent();
}

#region Component Designer generated code

//Required by the Web Services Designer
private IContainer components = null;

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
}

/// <summary>
/// Clean up any resources being used.
/// </summary>
protected override void Dispose( bool disposing )
{
if(disposing && components != null)
{
components.Dispose();
}
base.Dispose(disposing);
}

#endregion


//
//
// NOTES: a "string" return value on a webservice method will
properly base64 encode
// the response as necessary, allowing for binary values.
//
[WebMethod]
public string Encrypt(string strInput, string strPassword, string
strAlgorithm )
{

SymmetricAlgorithm Alg;

string strEncrypted = "";
if ( strAlgorithm.Equals("") )
{
strAlgorithm = "RIJNDAEL"; // default cipher
}
try
{
Alg = initAlgorithm( strAlgorithm, strInput, strPassword );
if ( Alg != null )
```

```
{
// Now it is time to encrypt.

ICryptoTransform encryptor = Alg.CreateEncryptor( keyBytes,
initVector );
MemoryStream ms = new MemoryStream();
CryptoStream cs = new CryptoStream( ms, encryptor,
CryptoStreamMode.Write);

// Start encrypting
cs.Write( plainText, 0, plainText.Length );
cs.FlushFinalBlock();

// Convert our encrypted data from a memory stream into a byte
array.
byte[] cipherTextBytes = ms.ToArray();
ms.Close();
cs.Close();

strEncrypted = Convert.ToBase64String( cipherTextBytes );
}
else
{
throw new Exception("Could not instantiate crypto service
provider.");
}
}
catch (Exception e)
{
throw e;
}
return strEncrypted;
}

[WebMethod]
public string Decrypt(string strInput, string strPassword, string
strAlgorithm )
{
SymmetricAlgorithm Alg;
// String strUnencoded = Convert.FromBase64String( strInput );

string strDecrypted = "";
if ( strAlgorithm.Equals("") )
{
strAlgorithm = "RIJNDAEL"; // default cipher
}
try
{
Alg = initAlgorithm( strAlgorithm, strInput, strPassword );
if ( Alg != null )
{
ICryptoTransform decryptor = Alg.CreateDecryptor( keyBytes,
initVector );
byte[] baInput = Convert.FromBase64String(strInput);
MemoryStream ms = new MemoryStream( baInput, false );
CryptoStream cs = new CryptoStream( ms, decryptor,
CryptoStreamMode.Read);
// Since at this point we don't know what the size of decrypted data
// will be, allocate the buffer long enough to hold ciphertext;
// plaintext is never longer than ciphertext.
byte[] plainTextBytes = new byte[strInput.Length];

// Start decrypting.
int decryptedByteCount = cs.Read(plainTextBytes, 0,
plainTextBytes.Length);

ms.Close();
cs.Close();

// Convert decrypted data into a string.
strDecrypted = Encoding.ASCII.GetString(plainTextBytes, 0,
decryptedByteCount);
}
else
{
throw new Exception("Could not instantiate crypto service
provider.");
}
}
catch (Exception e)
{
throw e;
}
return strDecrypted;
}


#region Class Utility Methods
// Crypto Factory
private System.Security.Cryptography.SymmetricAlgorithm
initAlgorithm( string strAlgorithm,
string strInput, string strPassword )
{
SymmetricAlgorithm sa;
// for a given input, return a concrete instance of the specified
algorithm
if ( strAlgorithm.ToUpper().Equals( "DES" ) )
{
sa = new System.Security.Cryptography.DESCryptoServiceProvider();
}
else if ( strAlgorithm.ToUpper().Equals( "3DES" ) )
{
sa = new
System.Security.Cryptography.TripleDESCryptoServiceProvider();
}
else if ( strAlgorithm.ToUpper().Equals( "RC2" ) )
{
sa = new System.Security.Cryptography.RC2CryptoServiceProvider();
}
else if ( strAlgorithm.ToUpper().Equals( "RIJNDAEL" ) )
{
sa = new System.Security.Cryptography.RijndaelManaged();
}
```

```
else
{
return null;
}
// get the plain text and cipher text (we use this function to go
either way).
plainText = Encoding.ASCII.GetBytes( strInput );

// set key size to 128 bit. Make sure it is valid for this
algorithm.
iKeySize = 128 / 8;
// Set up the key for encoding, using the provided password and key
size.
byte[] salt = Encoding.ASCII.GetBytes( "gscp-salt" );
PasswordDeriveBytes password = new PasswordDeriveBytes( strPassword,
salt );
keyBytes = password.GetBytes( iKeySize );

// For the initialization vector, ENCRYPT and DECRYPT must use the
same thing. We
// do a common trick here and use the password itself as the
initialization vector.
// Make sure it is the proper length for the algorithm.
int iPassLength = strPassword.Length;
sa.IV = password.GetBytes( sa.BlockSize / 8 );
initVector = sa.IV;

// Set up the key and cipher mode
sa.Mode = CipherMode.CBC;
try
{
sa.Key = keyBytes;
}
catch(Exception e)
{
if ( e.Message.IndexOf("valid size", 1) > 0 )
{
sa = null;
}
}

return sa;
}

private byte[] stringcharToByteArray(string str)
{
char[] s = str.ToCharArray();
byte[] b = new Byte[s.Length];
for ( int i = 0; i < s.Length; i++ )
{
b[i] = Convert.ToByte(s[i]);
}
return b;
}

#endregion
```

```
    }
    }
```

Encrypting Server-Side Files

These instructions provide three methods for document encryption. The first method uses a manually executed encryption command. It executes the specified file and then terminates. The second procedure automatically encrypts any file that is uploaded. This method is also event based so the encryption process occurs without operator assistance. The third method also automatically encrypts uploaded files but includes a security feature to safeguard files that might be vulnerable during the encryption process.

Why Encrypt Files?

File encryption is used to safeguard files from unauthorized access or tampering. Established and emerging security requirements recognize the need to encrypt files. Legal provisions, business objectives, or the sensitive nature of the data transmitted or stored provide the basis to limit access to file information. Protection for files in transit exists in the form of SSH and SSL secured transfers. However, once a file is securely transferred, the data is still at risk since it resides unencrypted on the server until it is removed. Another factor needing resolution is ordinary file deletion. Regular deletion does not completely destroy data. If the machine is hacked or if the physical security of the machine is compromised, unencrypted files may be downloaded or data recovery utilities could be used to restore previously deleted unencrypted files. Using a combination of encrypting and wiping techniques for all files can reduce the impact of having a system compromised.

Encryption Tools

These examples use PGP Command Line by Network Associates
http://web.mit.edu/network/pgp.html

This program is only free for non-commercial use. If your use of PGP is non-commercial then you should be able to use this version and these examples with almost no change.

For commercial purposes you can purchase a copy of PGP Command line from PGP Corporation
http://www.pgp.com

Another alternative is the Gnu Privacy Guard program, which is not restricted to non-commercial use
http://www.pgpi.org/products/gnupg/

Note:
This how-to instruction assumes a basic working knowledge of PGP or Public/Private key encryption. The public key used for encryption must be signed by a trusted source to avoid prompts that could stall the encryption process.

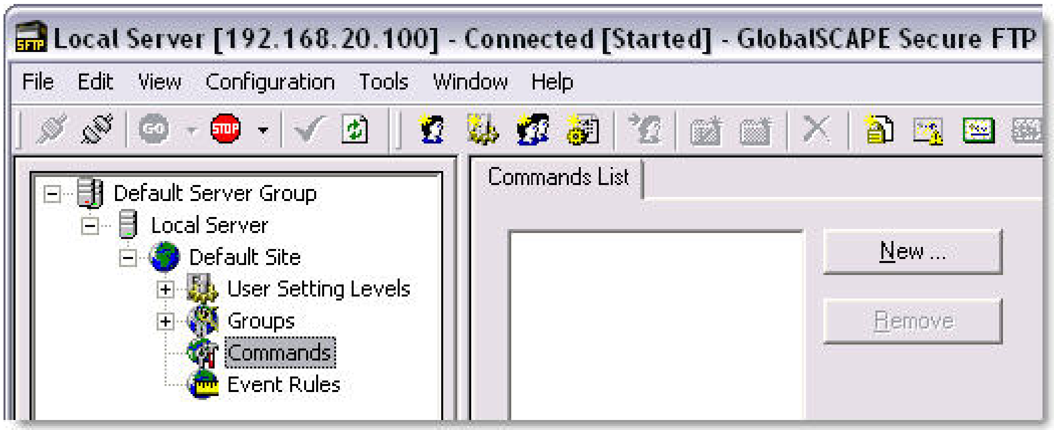
Method One - Manual Encryption by Custom Command

A custom command is the easiest way to handle encrypting files on a server. This section has two parts. The first part explains how to create a custom command and the second part explains how to execute the new encryption custom command.

PART ONE - Custom Command Creation

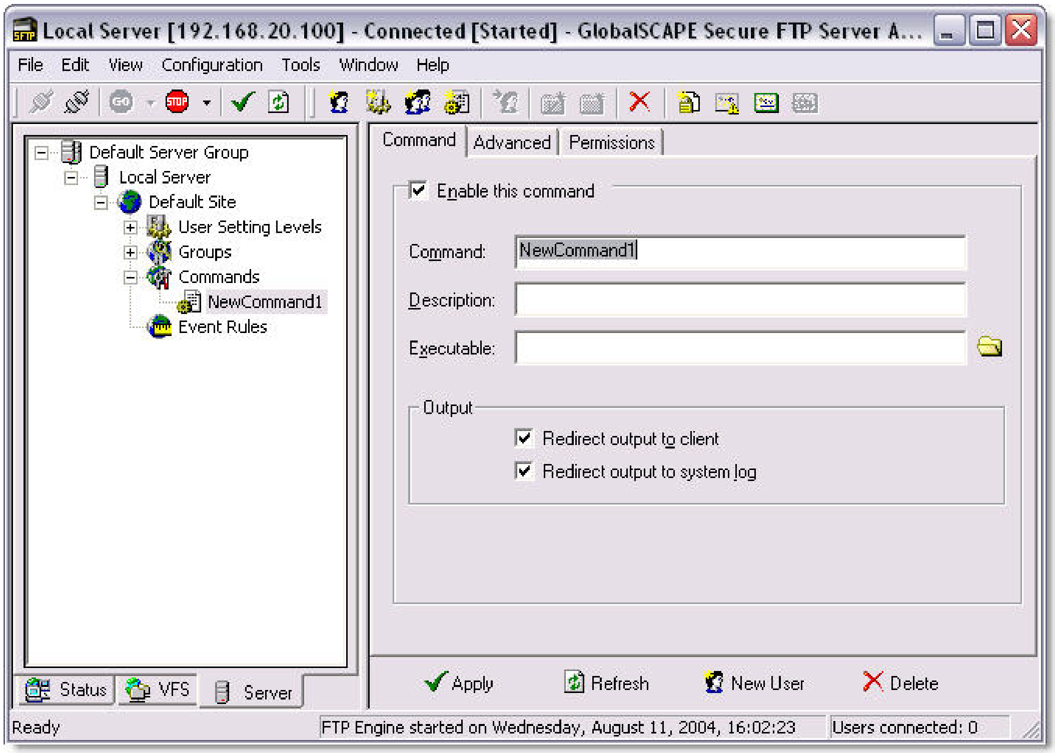
Create a custom command by following the regular process shown below.

1. Launch the Secure FTP Server GUI Administrator on the Server machine and connect to the local Secure FTP Server.
2. Expand the **Local Server** tree to view the **Commands** icon.
3. Select the **Commands** icon to display the **Commands List** window.
4. Press the **New** button to add a new command.


Local Server [192.168.20.100] - Connected [Started] - GlobalSCAPE Secure FTP
File Edit View Configuration Tools Window Help
Default Server Group
Local Server
Default Site
User Setting Levels
Groups
Commands
Event Rules
Commands List
New ...
Remove

The **Command** window displays.

5. Enter **pgp_encrypt** in the **Command** text box.
6. Enter the location of the PGP command line in the **Executable** text box.


Local Server [192.168.20.100] - Connected [Started] - GlobalSCAPE Secure FTP Server A...
File Edit View Configuration Tools Window Help
Default Server Group
Local Server
Default Site
User Setting Levels
Groups
Commands
NewCommand1
Event Rules
Status VFS Server
Command Advanced Permissions
Output
Apply Refresh New User Delete

☑ Enable this command

Command: NewCommand1

Description:

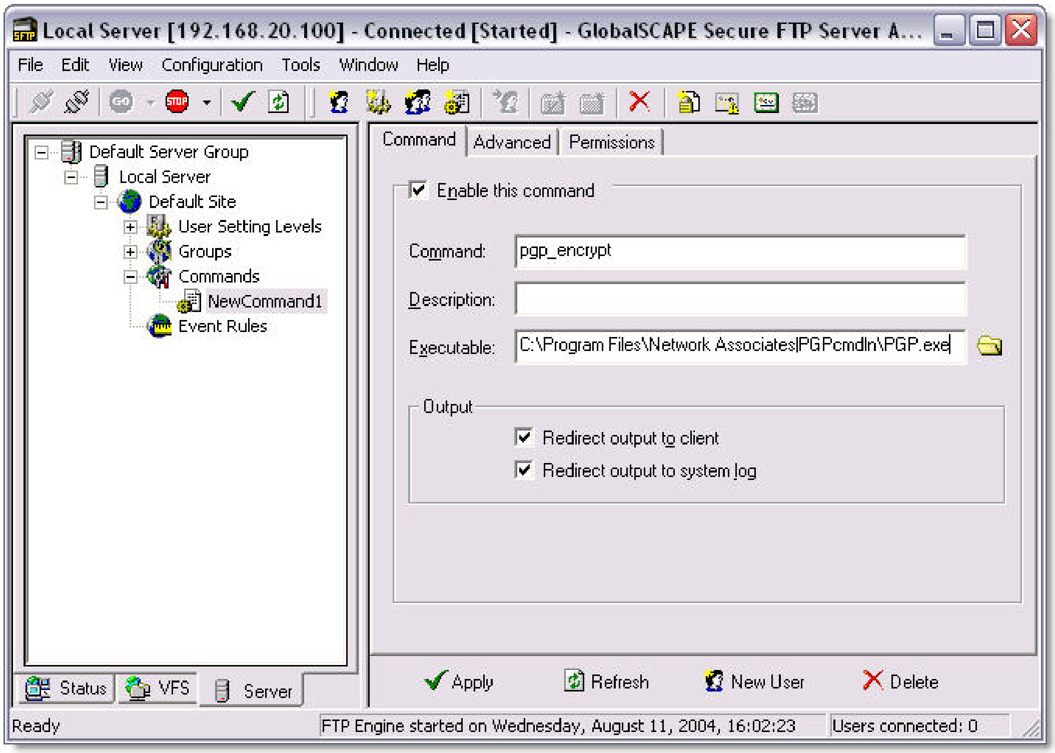
Executable:

☑ Redirect output to client

☑ Redirect output to system log

Ready | FTP Engine started on Wednesday, August 11, 2004, 16:02:23 | Users connected: 0

In this example the executable line reads:
C:\Program files\Network Associates\PGPcmdln\PGP.exe


Local Server [192.168.20.100] - Connected [Started] - GlobalSCAPE Secure FTP Server A...
File Edit View Configuration Tools Window Help
Default Server Group
Local Server
Default Site
User Setting Levels
Groups
Commands
NewCommand1
Event Rules
Command
Advanced
Permissions
Status
VFS
Server
Apply
Refresh
New User
Delete
Ready
FTP Engine started on Wednesday, August 11, 2004, 16:02:23
Users connected: 0

☑ Enable this command

Command: pgp_encrypt

Description:

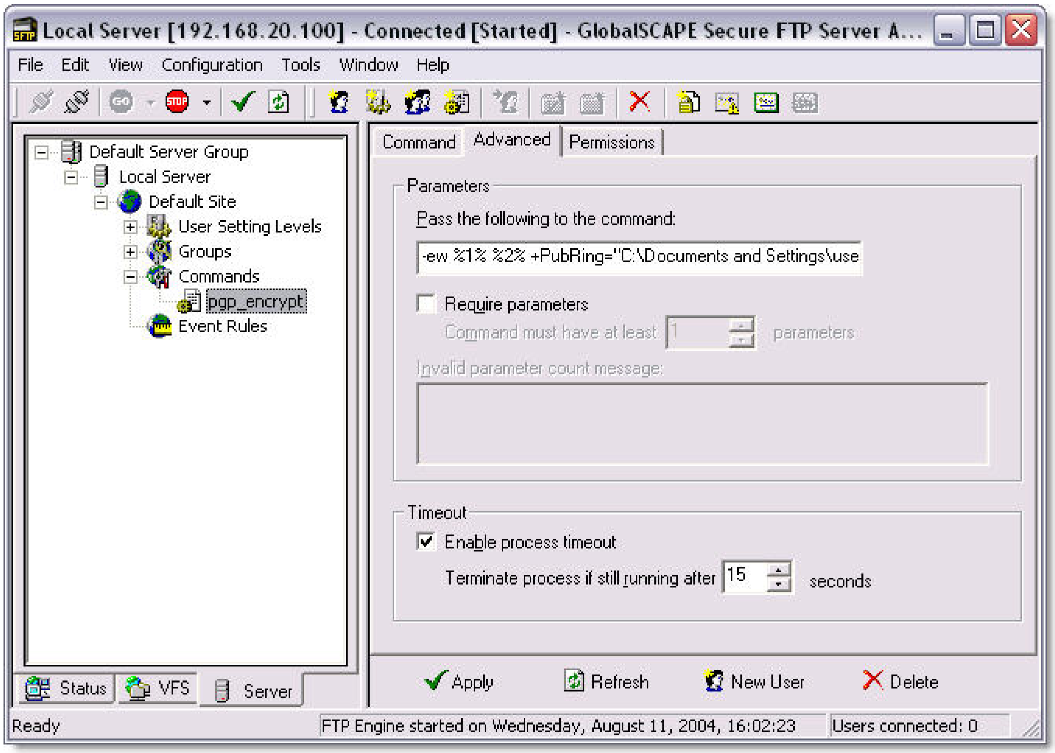
Executable: C:\Program Files\Network Associates\PGPcmdln\PGP.exe

Output

☑ Redirect output to client

☑ Redirect output to system log

7. Select the **Advanced** tab on the right side of the window to configure the parameters for encryption.


Local Server [192.168.20.100] - Connected [Started] - GlobalSCAPE Secure FTP Server A...
File Edit View Configuration Tools Window Help
Default Server Group
Local Server
Default Site
User Setting Levels
Groups
Commands
pgp_encrypt
Event Rules
Command
Advanced
Permissions
Status
VFS
Server
Apply
Refresh
New User
Delete
Ready
FTP Engine started on Wednesday, August 11, 2004, 16:02:23
Users connected: 0

Parameters

Pass the following to the command:

-ew %1% %2% +PubRing="C:\Documents and Settings\use

□ Require parameters

Command must have at least 1 parameters

Invalid parameter count message:

Timeout

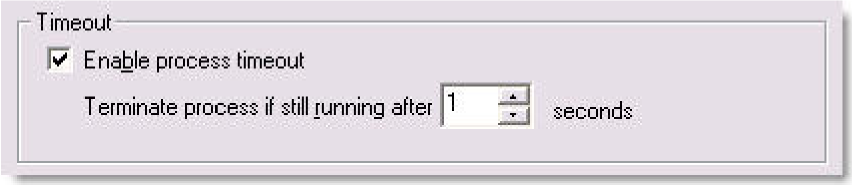
☑ Enable process timeout

Terminate process if still running after 15 seconds

8. Enter the Network Associate's PGP parameter command in the **Pass the following to the command:** text box:

-ew %1% %2% +PubRing="C:\Documents and Settings\<user>\Application Data\PGP\pubring.pkr
The -ew flag specifies two commands. The <e> command encrypts the file and the <w> command wipes the file. The wiping operation erases the file to preclude file recovery utilities from extracting any useful information from the disk denying access to the file's original contents. The +PubRing path must be changed to point to the file containing the list of public keys. The key ring location must be specified. Under normal usage explicitly setting the public key ring wouldn't be necessary, but when Secure Server launches PGP, the path environment variable will be missing. Specifying the +PubRing variable bypasses this problem by indicating the location of the file.

9. Set the Timeout variable to a reasonable value. Encrypting larger files requires a larger timeout value. One hundred and twenty seconds (or more) might be realistic if extremely large files are being encrypted.



Timeout

☑ Enable process timeout

Terminate process if still running after 1 seconds

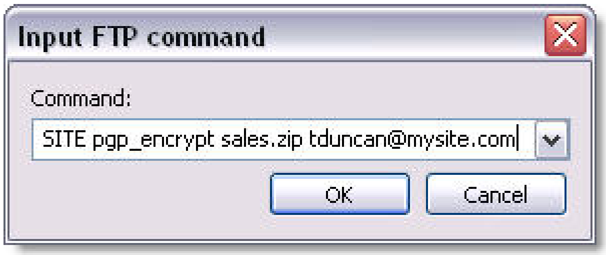
10. Select the Permissions tab to set necessary permissions for proper users to execute the encryption command.
11. Click Apply to set the rule and Permissions.

The Custom Command is complete.

PART TWO - Encryption Execution

1. In order to encrypt your files you must use an FTP client that supports sending raw commands to the FTP Server. CuteFTP Professional 6 supports this requirement.

   In CuteFTP Professional 6 go to the Tools Menu and select Enter FTP Command.



Input FTP command

Command:

```
SITE pgp_encrypt sales.zip tduncan@mysite.com
```

   The FTP command contains four parts:
   - SITE Command - Tells the server to execute a custom command
   - Custom Command - Follows the SITE command. pgp_encrypt is the name of the encryption custom command
   - File Name - Enter the name of the file to encrypt, e.g., sales.zip
   - Public Key Identifier - The identifier for the public key that should be used in the encryption e.g., **tduncan@mysite.com**.
2. Press the OK button to encrypt the file.
3. The client will reflect acknowledgement when the encryption is complete.

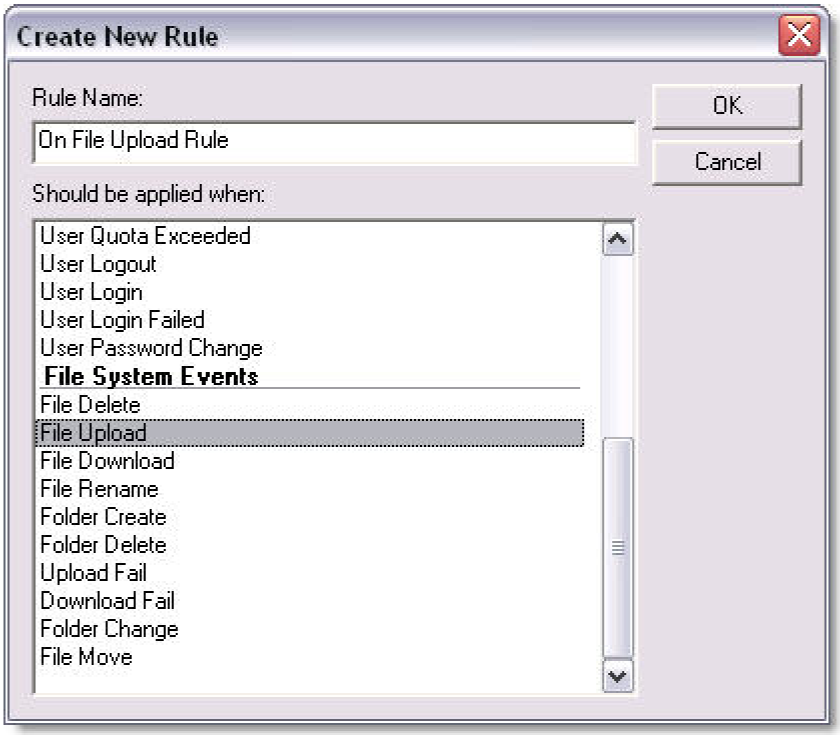
After execution, the original file should be wiped and only the encrypted file will be present.

Method Two - Automatic Encryption on Upload

Method Two builds on the previous process by adding an Event Rule that automatically launches the encryption custom command when a file is uploaded.

Event Configuration Steps:

1. Start the Secure FTP Server GUI Administrator on the Server machine and connect to the local Secure FTP Server.
2. Select the site where the files are to be encrypted.
3. Choose Create New Event Rule from the Configuration drop down menu (or right-click on the Site node in the tree view and choose New Event Rule).
4. Enter the name of the new rule in the Rule Name text box.
5. In the Should be applied when: section, double click the File Upload entry under the File System Events category. This will display the Event Rule screen.



Create New Rule

Rule Name: On File Upload Rule

Should be applied when: File Upload

6. On the Event Rule screen that appears, place a checkmark next to the Execute command in folder checkbox in the Specify rule actions area.
7. In the Specify rule condition and action parameters section, click on the pgp_encrypt "%FS.PATH%" tduncan@mysite.com hyperlink.
   a. In the Custom Command window that appears, select pgp_encrypt command from the choices in the Select command drop down menu.
   b. In the Specify command parameters text box there should be two parameters. The first is "%FS.PATH%" and the second is the name of the person whose public certificate should be used for the encryption.

Note:
The quote marks " " surrounding the parameter must be included.

   c. For the working folder, specify the location of the custom command's executable.

d. Click OK to save the entries and close the window.

Secure FTP Server is now ready to encrypt files upon upload.

If everything was properly set then whenever a file is uploaded it will be immediately encrypted and the original file wiped. Small files should appear to be encrypted almost immediately. Large files take longer and canat times be seen side-by-side with the unencrypted versions. Please see the security notes below for additional information.

Method Three - Enhanced Automatic Encryption on Upload

This method is an advanced implementation of the file encryption scheme used in Method Two.

The process used in this method seeks to limit potential file vulnerability during encryption. The larger the file, the longer it will take to encrypt. With long duration encryption, it is possible that a user could begin downloading the unencrypted version of the file. When this happens the server locks the file and the wipe portion of the encrypt/wipe operation fails. This means two versions of the file exist in the upload directory: one encrypted the other unencrypted. Using Method Three reduces the risk of possible download by invoking a batch file to move the file. The Method Three process moves the file to another directory in the same partition, encrypts the data, and then moves the encrypted file back to its original upload location.

This example specifically directs that files move within the same partition. The batch process starts by creating a CRYPTDIR directory off of the root of the current partition as the location where encryption occurs. Staying within the same partition is CRITICAL to maximize the security of the data. Moving data within the same partition only takes milliseconds. This is because the data does not have to be read and rewritten -- the file allocation table is simply modified to reflect the file's new directory.

Moving across a partition is actually two separate operations: A copy followed by a move. This can take seconds and possibly even minutes for large files. If a download occurred during the move process, the file would be locked and the purpose of this process would be defeated. Another reason to stay within the same partition is because the "delete" portion of the "copy/delete" operation during a move is insecure. Moving a file across a partition still leaves the file contents on the old partition. This residual data is susceptible to data recovery utilities. Proper operation of the PGP encryption command described in this method will only wipe the unencrypted file information in the new location of the file.

These instructions show how to implement the advanced auto-encryption routine. There are three main parts to the instructions:

Secure batch file

Place the following code in the appropriate directory to perform the 'move' job explained in the introduction to this section.

```
REM File: Secure.bat
REM %1 is the path of the file
REM %2 is the file name
REM %3 is the public key identifier
md \CryptDir
cd \CryptDirattrib -r-h %2
attrib -r-h %2.pgp
del %2
del %2.pgp
cd %1
```

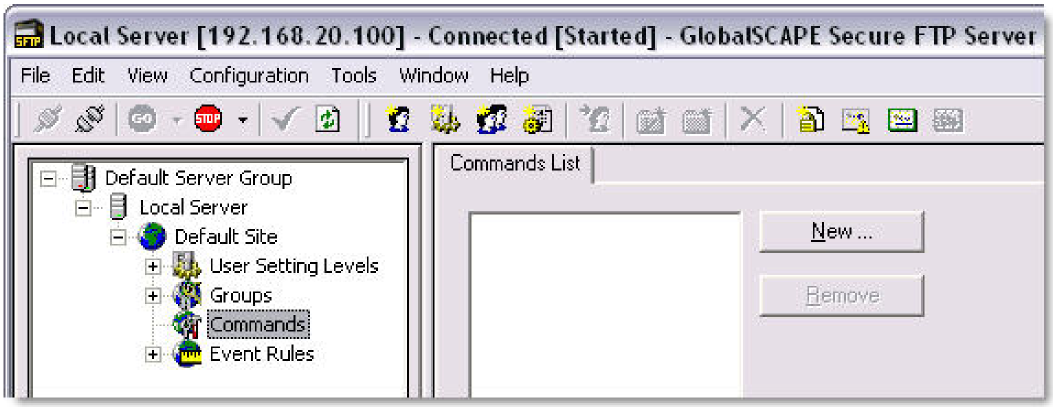
```
move %2 \CryptDir
cd \CryptDir"C:\Program Files\Network Associates\PGPcmdln\PGP.exe" -
ew %2 %3
+PubRing="C:\Documents and Settings\user\Application
Data\PGP\pubring.pkr"
cd %1
move \CryptDir\%2.pgp %2.pgp
```

The batch file is constructed to always remain on the same partition as the file.

Create a Custom Command

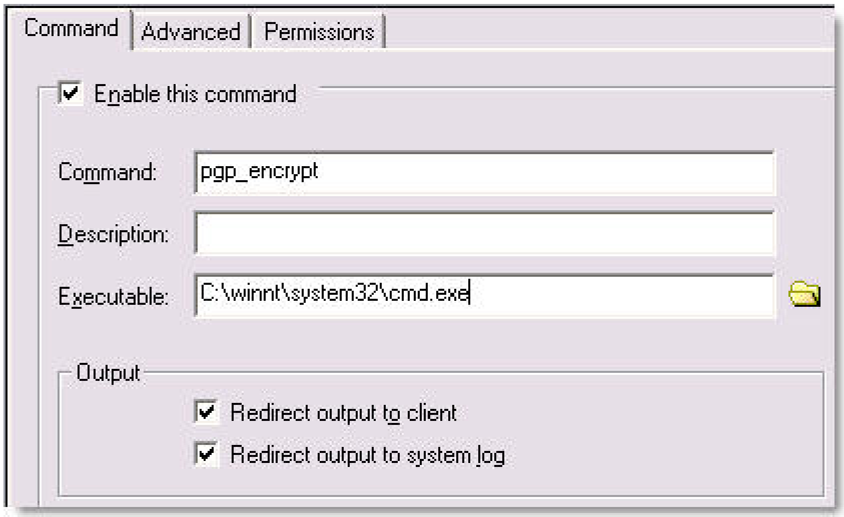
Create a custom command by following the regular process shown below.

1. Launch the Secure FTP Server GUI Administrator on the Server machine and connect to the local Secure FTP Server.
2. Expand the Local Server tree to view the Commands icon.
3. Select the Commands icon to display the Commands List window.



4. Press the New button to add a new command.

   The Command window will be displayed.
5. Enter pgp_encrypt in the Command text box.
6. Enter the location of the batch command line in the Executable text box.



☑ Enable this command

Command: pgp_encrypt

Description:

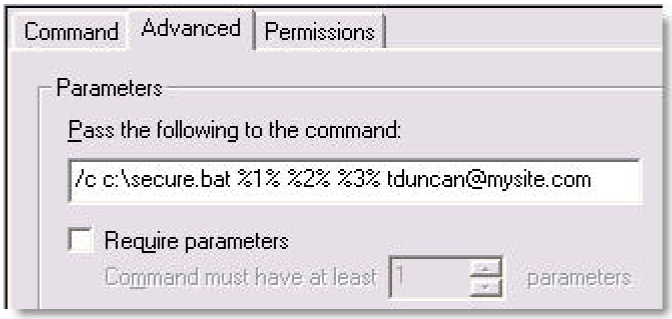
Executable: C:\winnt\system32\cmd.exe

☑ Redirect output to client

☑ Redirect output to system log

In this example, the executable line reads: **C:\winnt\system 32\cmd.exe**

7. Select the **Advanced** tab on the right side of the window to configure the parameters for encryption.
8. Enter the batch file parameter command in the **Pass the following to the command:** text box:



Parameters

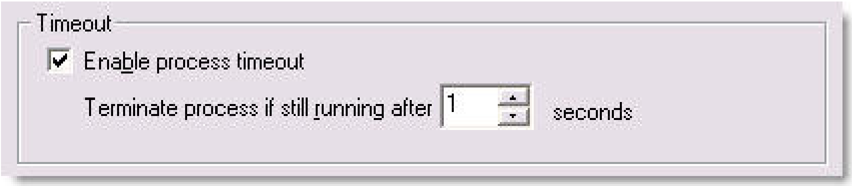
Pass the following to the command: /c c:\secure.bat %1% %2% %3% tduncan@mysite.com

□ Require parameters

Command must have at least 1 parameters

The parameter **tduncan@mysite.com** is the location of the PGP key used.

9. Set the **Timeout** variable to a reasonable value. Encrypting larger files requires a larger timeout value. One hundred and twenty seconds (or more) might be realistic if extremely large files are being encrypted.



Timeout

☑ Enable process timeout

Terminate process if still running after 1 seconds

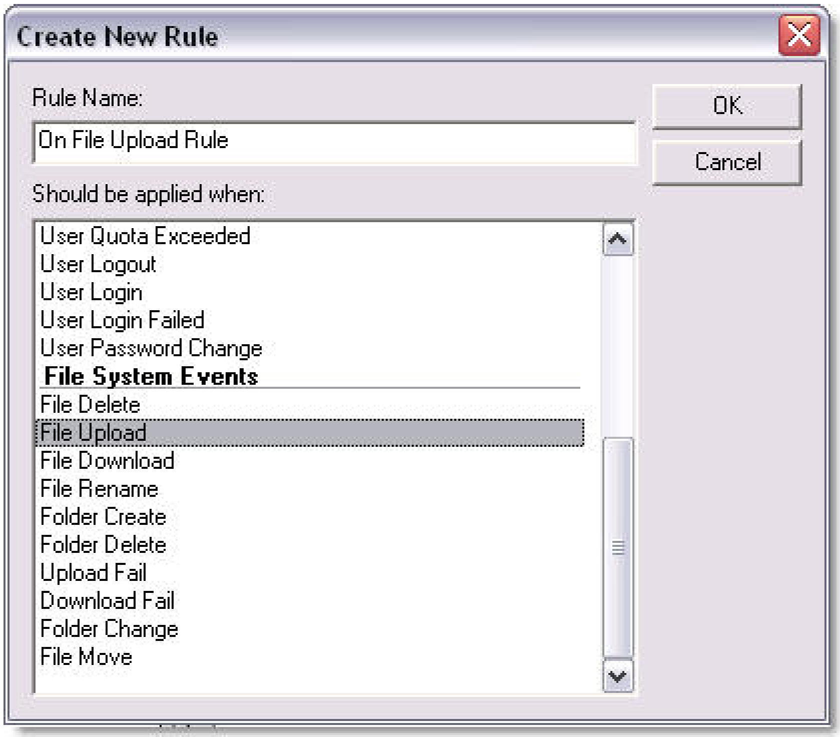
10. Select the **Permissions** tab to set necessary permissions for proper users to execute the encryption command.
11. Click **Apply** to set the rule and **Permissions**.

The Custom Command is complete.

Specify an Event Rule to trigger the encryption action

Event Configuration Steps:

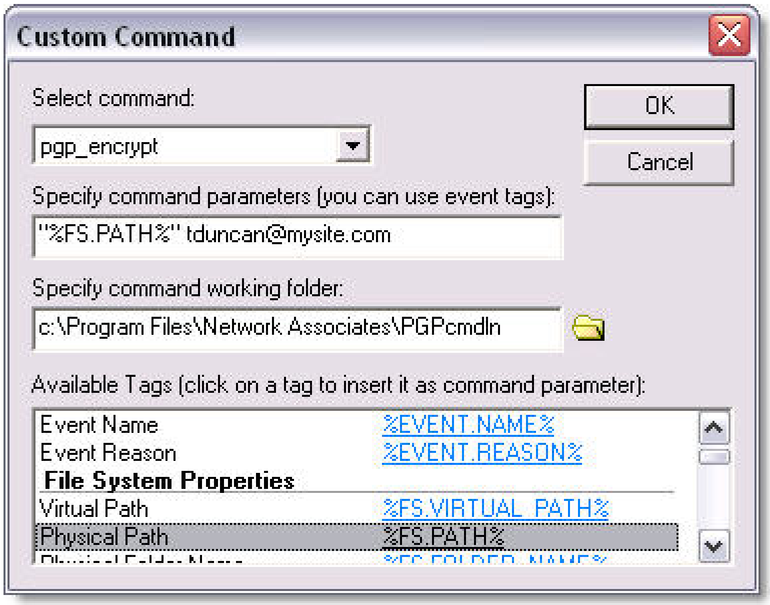
1. Start the Secure FTP Server GUI Administrator on the Server machine and connect to the local Secure FTP Server.
2. Select the site where the files are to be encrypted.
3. Choose **Create New Event Rule** from the **Configuration** drop down menu (or right-click on the Site node in the tree view and choose **New Event Rule**).
4. Enter the name of the new rule in the **Rule Name** text box.



Rule Name: On File Upload Rule

Should be applied when: File Upload

5. In the **Should be applied when:** section, double click the **File Upload** entry under the **File System Events** category. This will display the Event Rule screen.
6. On the Event Rule screen that appears, place a checkmark next to the **Execute command in folder** checkbox in the **Specify rule actions** area.
7. In the **Specify rule condition and action parameters** section, click on the **pgp_encrypt "%FS.FOLDER_NAME%" "%FS.FILE_NAME%"** hyperlink.
   a. In the **Custom Command** window that appears, select pgp_encrypt command from the choices in the **Select command** drop down menu.



Select command: pgp_encrypt

Specify command parameters (you can use event tags): "%FS.PATH%" tduncan@mysite.com

Specify command working folder: c:\Program Files\Network Associates\PGPcmdln

Available Tags (click on a tag to insert it as command parameter):

Event Name	%EVENT.NAME%
Event Reason	%EVENT.REASON%
File System Properties	
Virtual Path	%FS.VIRTUAL_PATH%
Physical Path	%FS.PATH%
Physical Folder Name	%FS.FOLDER_NAME%

   b. In the **Specify command parameters** text box there should be two parameters: **"%FS.FOLDER_NAME%"** and **"%FS.FILE_NAME%"**.

c. For the working folder, specify the location of the custom command's executable. In the example it is:

   C:\Program Files\Network Associates\PGPcmdln

d. Click **OK** to save the entries and close the window.

Secure FTP Server is now ready to encrypt files upon upload. Some tweaking may be needed such as ensuring all paths are correct. As a final suggestion, the **secure.bat** file can be moved out of the root directory.

How to Manage Temporary Users for One-Time Downloads

Secure FTP Server can support e-commerce merchants who need to limit customer downloads from their website to a single copy of a file. This example uses PHP scripts to create a temporary user account for the purchaser and restrict the purchaser to a single download of the product. This is particularly useful for digital document distribution. This example uses Secure FTP Server's COM interface, two custom PHP scripts, a custom site command, and an event rule.

Technology Used in These Directions

These instructions use PHP scripting techniques applied to GlobalSCAPE's secure FTP Server to extend the functionality of the system. For this customization to operate properly, PHP version 4.3.3 (or above) should be installed on the same machine as the Secure FTP Server software. Although these instructions are created with PHP, users can develop an equivalent script in an alternate server-side scripting language such as ASP, ASP.NET, JSP, PERL, or COLD FUSION.

These instructions require the installation of two PHP scripts. The first script is launched by the web server when the customer makes a purchase. This script creates a temporary user account, gives permissions, and sends an e-mail to the customer. The second script is launched after the customer has performed a successful download and permits no further downloads.

Create a temporary account

Use a PHP script to create a temporary account for a customer to download a product.

Insert the first script, **CreateTempUser.php**, into the appropriate directory. This scripts asks, using the COM interface, Secure FTP Server to create a temp user account, giving the user 'download only' permission to a specified folder. After the temporary user account is created, the script sends an email notification to the customer. This email indicates a temporary FTP account has been created and provides a user name and password to for the customer to log in. For the email process to operate correctly, the SMTP server specified in PHP.ini needs to be configured to allow access by the Secure FTP server.

Script Assumptions
This script assumes:

a. The Web Server and the FTP Server are on the same machine; specifically, the e-commerce web server has Secure FTP Server installed and configured. If the Secure FTP Server is located on a different machine, the **local host** value needs to be changed to the IP address of the FTP Server and the FTP Server must be configured to allow remote administration.

b. The Administrator username is **root** and the password is **hello**.

Source Script: CreateTempUser.php

```
/*==================================================================
=========
*= File: CreateTempUser.php
*= Created: 07/20/2004 WHT
*= Purpose:
*= Create a new temporary user and send email notification,
*= the user name begin with temp_
*= NOTES:
*=    Command Paramaters:
*=     User email address, download folder
*=
*= USAGE:
*= php -f CreateTempUser.php <user email address> <the download
folder>
*= EXAMPLE:
*= php -f CreateTempUser.php user@mail.com /Pub/
*= NOTE
*=    REQUIRES PHP 4.3.3 or above version
*=
*===================================================================
========
*/

//Check if command line parameters is set
if ($argc != 3)
exit;
//Create Server Object
$SFTPServer = new COM("SFTPCOMInterface.CIServer");
if (isset($SFTPServer) && $SFTPServer != false)
{
//Connect to Server Engine
$SFTPServer->Connect("localhost", 1000, "root", "hello");
//Get the site list
$Sites = $SFTPServer->Sites();
//Choose the first site
$Site = $Sites->Item(0);
if (isset($Site) && $Site != NULL)
{
//Create a new user name, the prefix of user name is temp_
$username = "temp_" . mt_rand(1, 8000);
//Create password
$tempstr = time() . mt_rand(1, 8000);
$userpass = substr(md5($tempstr), 0, 8);
//Create the user
$Site->CreateUser($username, $userpass, 0, $username);
//Give the download permission of the specified folder to this user
$NewPermission = $Site->GetBlankPermission($argv[2], $username);
$NewPermission->FileDownload = TRUE;
$Site->SetPermission($NewPermission);
$UserSettings = $Site->GetUserSettings($username);
//Set the user's email address
$UserSettings->Email = $argv[1];
//Enable the home directory
$UserSettings->SetHomeDir(1);
//Set the directory
$UserSettings->SetHomeDirString($argv[2]);
//Apply the change
$SFTPServer->ApplyChanges();
//Send mail to notify the user
$title = "Your temporary FTP account is set up";
$body = "Dear Sir or Madam:\n\nYour temporary FTP account is set
up.\n".
"The user name is $username and password is $userpass.\nThis
account".
"will be deleted immediately after you download a
file.\n\n\nSincerely yours";
@mail($argv[1], $title, $body);
}
exit;
}
?>
```

Prevent subsequent use

This section describes the process and provides the tool to prevent subsequent downloads from the temporary user account after the first successful download. The instructions show how to insert the second PHP script, create a custom site command, and create an event trigger that uses the command to launch the script.

1. Copy the script named **downloaded.php** to a location on the publicly accessible server that is outside the VFS. Provide appropriate permissions so the service account running GlobalSCAPE Secure Server can read and execute that script.

   Script Assumptions
   This script must reside on the same machine as the Secure FTP Server since the FTP Server initiates the script not the web server.

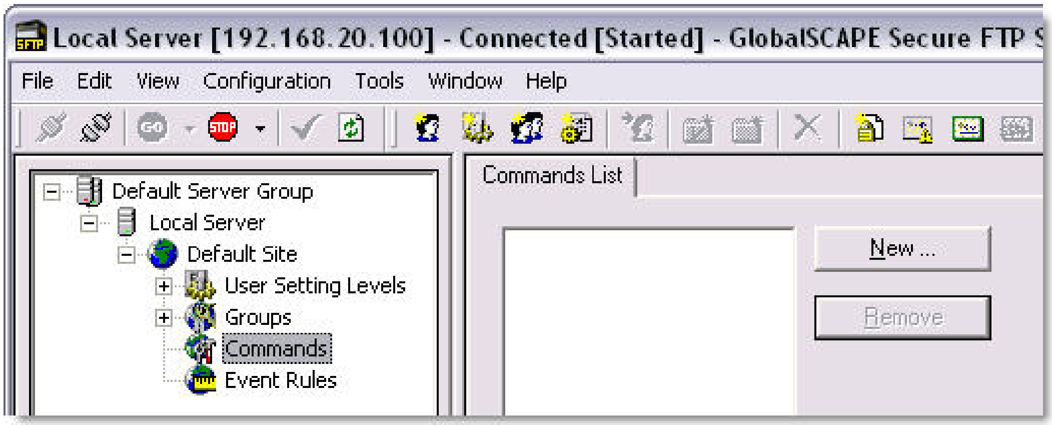
   Source Script: downloaded.php

```
<?php
/*==============================================================
=========
*= File: downloaded.php
*= Created: 07/20/2004 WHT
*= Purpose:
*= Set the max download size for the specified user to 0 after this
user
*= successfully downloaded file from FTP server.
*= NOTES:
*=    Event Rule: On File Download
*=    Command Paramaters:
*=     -f downloaded.php "%USER.LOGIN%"
*= USAGE:
*= php -f downloaded.php <user name>
*= EXAMPLE:
*= php -f downloaded.php temp_1110
*= NOTE
*=    REQUIRES PHP 4.3.3 or above version
*=
*===================================================================
========
*/
//Check if user name is set
if (($argc != 2) || empty($argv[1]))
exit;
//Create Server Object
$SFTPServer = new COM("SFTPCOMInterface.CIServer");
if (isset($SFTPServer) && $SFTPServer != false)
{
//Connect to Server Engine
$SFTPServer->Connect("localhost", 1000, "root", "hello");
//Get the site list
$Sites = $SFTPServer->Sites();
//Choose the first site
$Site = $Sites->Item(0);
if (isset($Site) && $Site != NULL && substr($argv[1], 0, 5) ==
"temp_")
{
//$result = $Site->RemoveUser($argv[1]);
$UserSettings = $Site->GetUserSettings($argv[1]);
//Set max download size to 0
$UserSettings->SetHasMaxDownloadSize(1);
$UserSettings->SetMaxDownloadSize(0);
//Apply the change
$SFTPServer->ApplyChanges();
}
}
exit;
```

?>

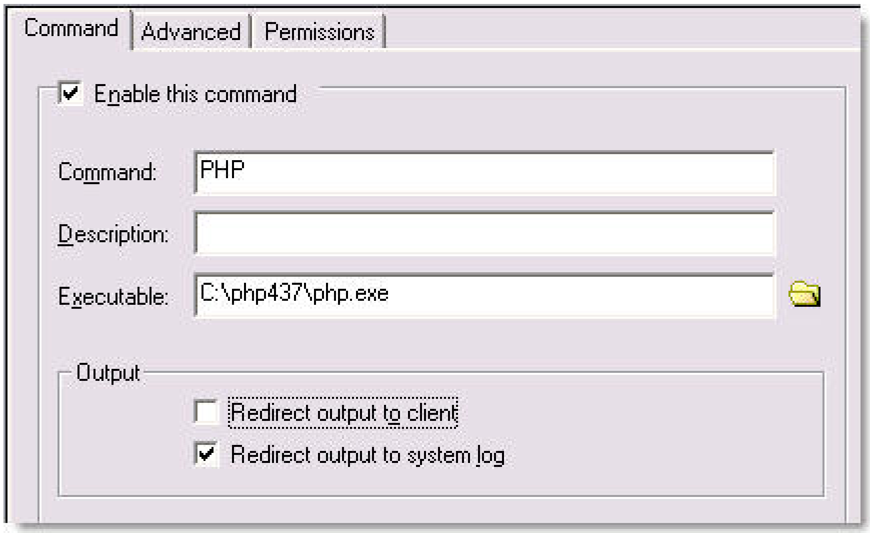
The next step is to create a custom site command that launches the **downloaded.php** script using the **php.exe** command. Create a custom command by following the process shown below.

1. Launch the Secure FTP Server GUI Administrator on the Server machine and connect to the local Secure FTP Server.
2. Expand the **Local Server** tree to view the **Commands** icon.


Local Server [192.168.20.100] - Connected [Started] - GlobalSCAPE Secure FTP S
File Edit View Configuration Tools Window Help
Default Server Group
Local Server
Default Site
User Setting Levels
Groups
Commands
Event Rules
Commands List
New ...
Remove

3. Select the **Commands** icon to display the **Commands List** window.
4. Press the **New** button to add a new command.

   The **Command** window will be displayed.
5. Enter **PHP** in the **Command** text box.
6. Enter the location of the PHP script command line in the **Executable** text box.
7. Uncheck the **Redirect output to client** box.



☑ Enable this command

Command: PHP

Description:

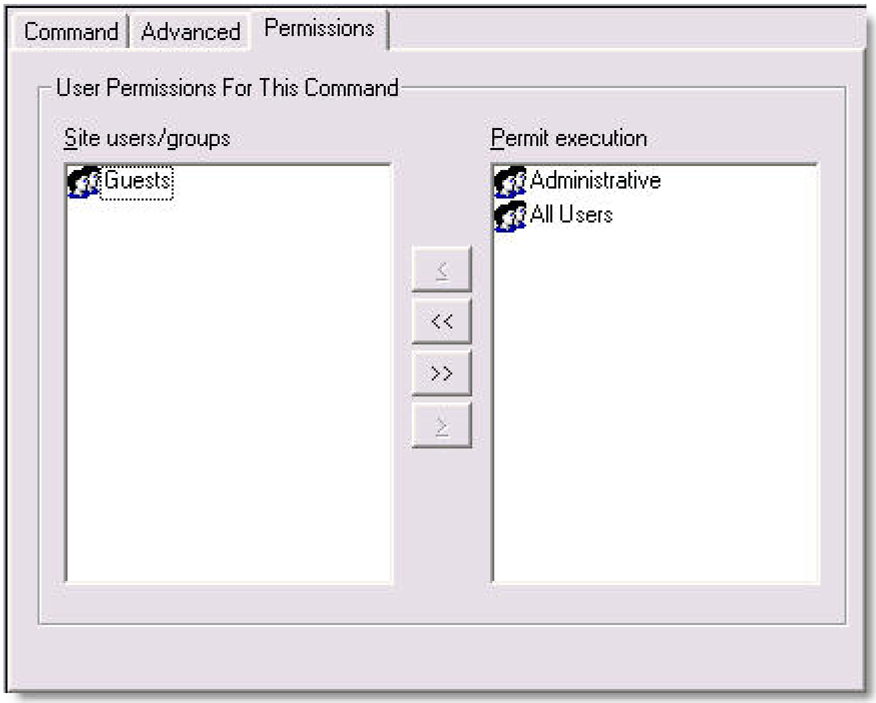
Executable: C:\php437\php.exe

□ Redirect output to client

☑ Redirect output to system log

   In this example the executable line reads: **C:\php437\php.exe**

8. Select the **Advanced** tab on the right side of the window to configure the parameters.
9. Place a check in the box **Require parameters** check box.
10. Make sure there is a "**1**" in the **Command must have at least ____ parameters** box.
11. Select the **Permissions** tab to set necessary permissions for proper users to execute the encryption command.



12. Click **Apply** to set the rule and **Permissions**.

    The Custom Command is complete.

    The last sequence specifies an event rule to trigger the command

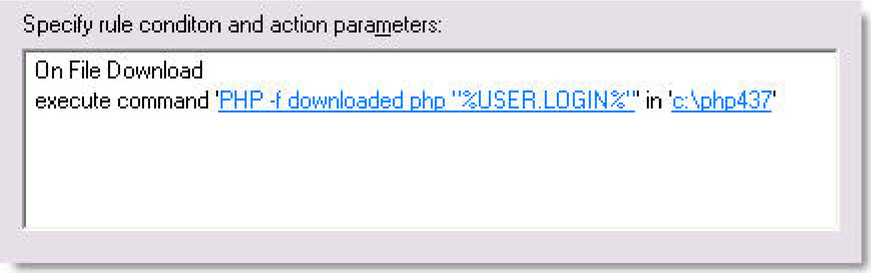
13. Start the Secure FTP Server GUI Administrator on the Server machine and connect to the local Secure FTP Server.
14. Select the site where the event will take place.
15. Choose **Create New Event Rule** from the **Configuration** drop down menu (or right-click on the Site node in the tree view and choose **New Event Rule**).
16. Enter the name of the new rule in the **Rule Name** text box.
17. In the **Should be applied when:** section, double click the **File Download** entry under the **File System Events** category.

    The **Event Rule** screen displays.

    On the Event Rule screen that appears, place a checkmark next to the **Execute command in folder** checkbox in the **Specify rule actions** area.

18. In the **Specify rule condition and action parameters** section on the Event Rule screen, click on the hyperlink:

    'PHP -f downloaded.php "%USER.LOGIN%"



Specify rule conditon and action parameters:

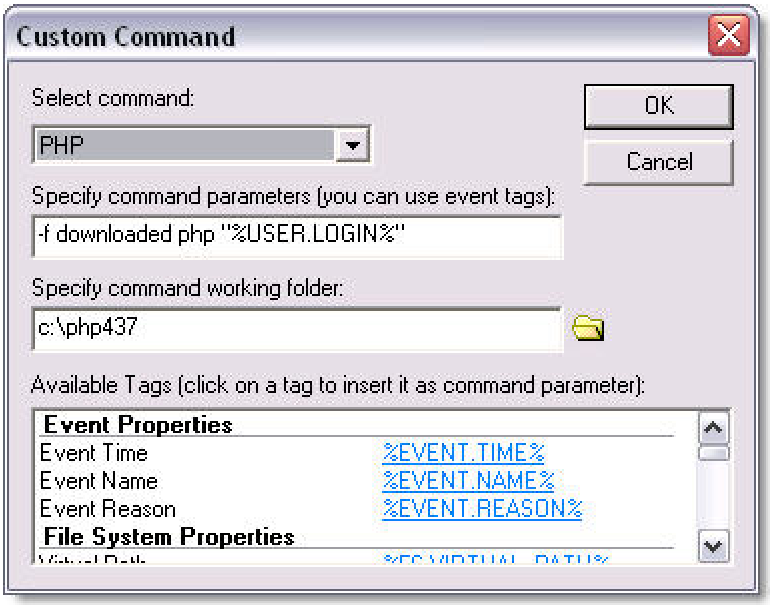
On File Download
execute command 'PHP -f downloaded.php "%USER.LOGIN%"' in 'c:\php437'

The **Custom Command** window appears.

1. In the Custom Command window, select the **PHP** command from the choices in the **Select command** drop down menu.
2. In the **Specify command parameters** text box there should be two parameters:

   -f uploaded.php and "%USER.LOGIN%"

Note:
The quote marks " " surrounding the parameter must be included.



Select command: PHP

Specify command parameters (you can use event tags): -f downloaded php "%USER.LOGIN%"

Specify command working folder: c:\php437

Available Tags (click on a tag to insert it as command parameter):

Event Properties	
Event Time	%EVENT.TIME%
Event Name	%EVENT.NAME%
Event Reason	%EVENT.REASON%
File System Properties	
Virtual Path	%FS.VIRTUAL_PATH%

3. For the working folder, specify the location of the custom command's executable. In this case it is: **C:\php437**
4. Click **OK** to save the entries and close the window.

Secure FTP Server is now configured to create a temporary user account, provide the customer email notification of the account setup, and permit the customer only one successful download of the purchased product.

Note:
Since a user can't be deleted using the COM interface when the user is still logged in, the PHP script simply resets the temporary user account parameter 'max download size' to 0 after a successful download. The temporary user account must be deleted later **manually** using the Secure FTP Server GUI Administrator.

Note:
Be sure to review, customize, and test the tools and information provided in these directions prior to using them in any non-test environment.

Moving Files to Remote Locations after an upload

Publicly accessible server should be configured to move incoming transactions to a location inside the corporate firewall. This provides additional protection for the files since they can rely on secured corporate intranets.

Secure FTP Server can automatically move these files from a publicly accessible server to a more secure server inside the enterprise firewall. This added measure of security is achieved by using a script that manipulates the COM interface of the CuteFTP **transfer engine** and moves the files with the integral file system or any of the FTP, FTPS, SFTP, or HTTP protocols.

Factors to consider when moving files after an upload

Security for the enterprise intranet is based on an effective firewall separating the intranet from external Internet threats. Moving files from a publicly accessible server to the private intranet server requires a communication channel through the firewall. Obviously, this channel must be as secure as possible to reduce the risk of compromise to the enterprise intranet.

Access to this communications channel can be configured at the network link level via the firewall so only the "inside" network card of the publicly accessible server can route packets through the channel. Access at the FTP Server level should establish user authentication to make secure transfers between the two systems.

Example configurations with a high level of security

- The Intranet FTP server is configured to run FTP over SSH (SFTP), and a user is configured with public key authentication. The private key is stored on the publicly accessible server in a location that is inaccessible by the FTP Service's virtual file system (VFS) - this reduces the risk of compromise of the key. After a file is uploaded, the publicly accessible server forwards the incoming file to the server using an SSH client, such as GlobalSCAPE CuteFTP Professional, which is configured to use the private key that corresponds to the public key stored on the Intranet server. The firewall is configured to allow communication between these two servers only on port 22 (the SSH protocol port) and the servers' IP addresses.
- The Intranet server can also be configured to use FTP over SSL, combined with requiring client certificates for authentication purposes. Analogous to the SSH private key above, the publicly accessible server stores the client certificate in a secure location and pushes the file to the Intranet server using a standard FTP client that supports FTP over SSL, such as GlobalSCAPE CuteFTP Professional. The FTP protocol requires at least two ports to be opened on the firewall; the specific port number depends upon the SSL connection type (IMPLICIT versus EXPLICIT) and the transfer mode (PORT versus PASSIVE). The simplest configuration would be to use implicit SSL in PASV mode, with the server specifying a "passive mode port range." The Firewall would then be configured to open up between the two servers port 990 (for the initial connection) as well as the ports established in the "passive mode port range" setting of the Intranet server.

- A windows share is created on the Intranet server. The publicly accessible server moves the uploaded file to the Intranet server using a mapped dive (UNC path). The Intranet server is configured so the VFS maps to this network share. In this configuration, Intranet users can use standard FTP clients, such as GlobalSCAPE CuteFTP Home, to review the uploaded files immediately after they are uploaded.

Scripting CuteFTP Professional to Move a File After Upload

The method to improve security for moving files uploaded from an Internet accessible server to a remote destination employing FTP, FTPS, SFTP protocols uses GlobalSCAPE CuteFTP Professional in conjunction with a script (hosted by Windows Scripting Host). The script manipulates the COM interface of the CuteFTP Professional transfer engine based upon the input parameters in order to move a file from the publicly accessible server to the Intranet server using either the filesystem or any of the FTP, FTPS, SFTP, or HTTP protocols.

The script is invoked from an **event rule** that uses the context information of a completed transfer to identify the file to transfer. The **custom site command** that drives the event rule also hardcodes the destination information for the file (remote host, username, password, and remote folder); however, extensions to the script should be readily identifiable to parameterize this portion as well using additional site commands or modifications to the script itself.

Configuration of the publicly accessible server

The GlobalSCAPE server on the intranet must be configured to allow incoming client connections from the publicly accessible FTP server.

Create a user account and determine a connection method for this data brokering. Elements that should be considered in this situation are:

- Create a user account that must log in over SFTP using public key authentication (the "brokering user").
- Create the public/private key pair. Associate the public key with the brokering use, and place the private key ("Identity File") on a secure location of the publicly accessible FTP server (in a folder outside the VFS structure).
- Limit the brokering user to log in **only** from the internal IP address of the publicly accessible server. Allow for multiple concurrent connections, as the publicly accessible FTP server will be servicing multiple simultaneous clients and it is likely that multiple file transfers must be brokered to the Intranet FTP server concurrently.
- Create the desired folder structure on the Intranet FTP server to receive the data brokered by the publicly accessible server. Include upload permissions for the brokering user.

Script Directions

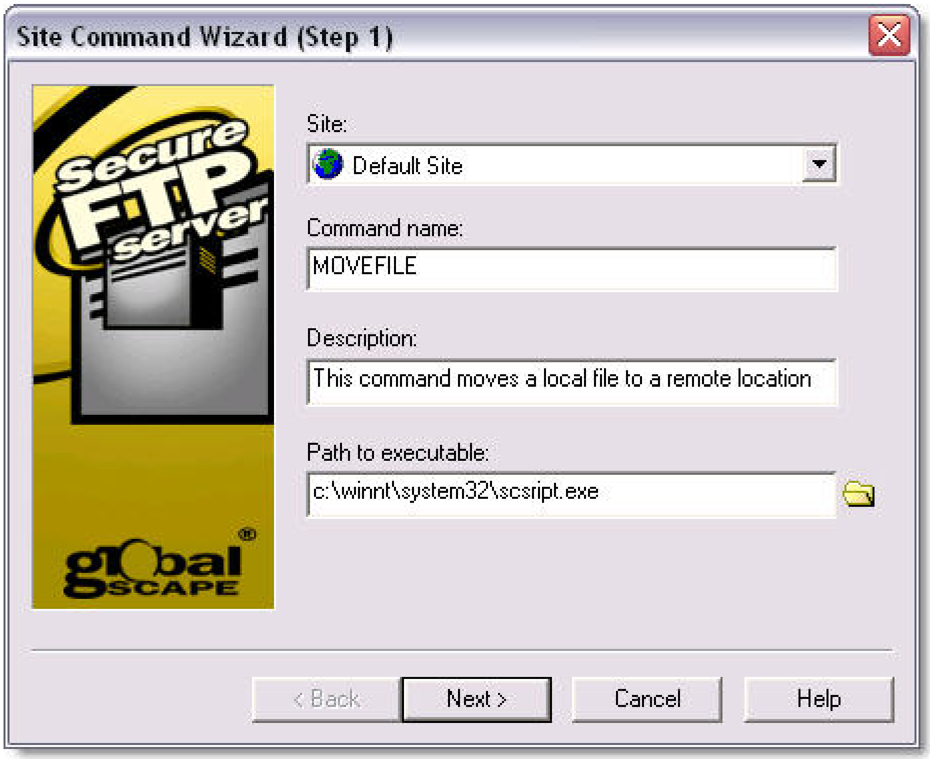
Copy the **movefile** script (see below) to a location on the publicly accessible server outside the VFS. Provide appropriate permissions so the service account running GlobalSCAPE Secure FTP Server can read and execute the script.

Custom Site Command Creation

Create a **custom site command** to launch the **movefile** script.

1. Launch the Secure FTP Server GUI Administrator on the Server machine and connect to the local Secure FTP Server.
2. Click on the site for which you want log files encrypted and choose "Create New Command" from the configuration menu (or press CTRL+M on the keyboard). The Site Command Wizard will appear.

3. Enter **MOVEFILE** in the **Command name:** text box.
4. Specify **cscript.exe** in the **Path to executable:** text box as the Microsoft Windows® command-line Scripting Host.



Site Command Wizard (Step 1)

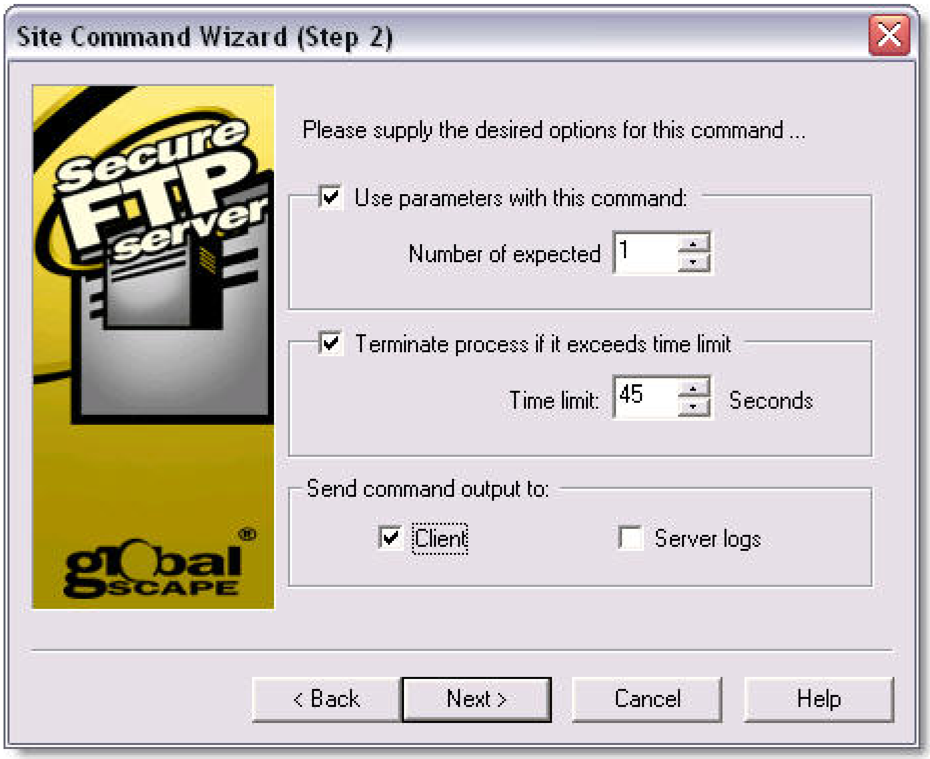
Site: Default Site

Command name: MOVEFILE

Description: This command moves a local file to a remote location

Path to executable: c:\winnt\system32\scsript.exe

When this is entered, click **Next** to continue.



Site Command Wizard (Step 2)

Please supply the desired options for this command ...

☑ Use parameters with this command:

Number of expected 1

☑ Terminate process if it exceeds time limit

Time limit: 45 Seconds

Send command output to:

☑ Client ☐ Server logs

5. On the Site Command Wizard (Step 2) page:

a. Place a check the **Use parameters** box and specify that one parameter is needed.

b. Place a check in **Terminate process** box to prevent a 'runaway process.' The example shows that Secure FTP Server will terminate this command if it takes over 45 seconds to complete.

c. Check the **Client** box so the output will be sent to the client.

When this is done, click **Next** to continue.

6. In Step 3 of the Wizard, the parameters to the CSCRIPT executable are specified.

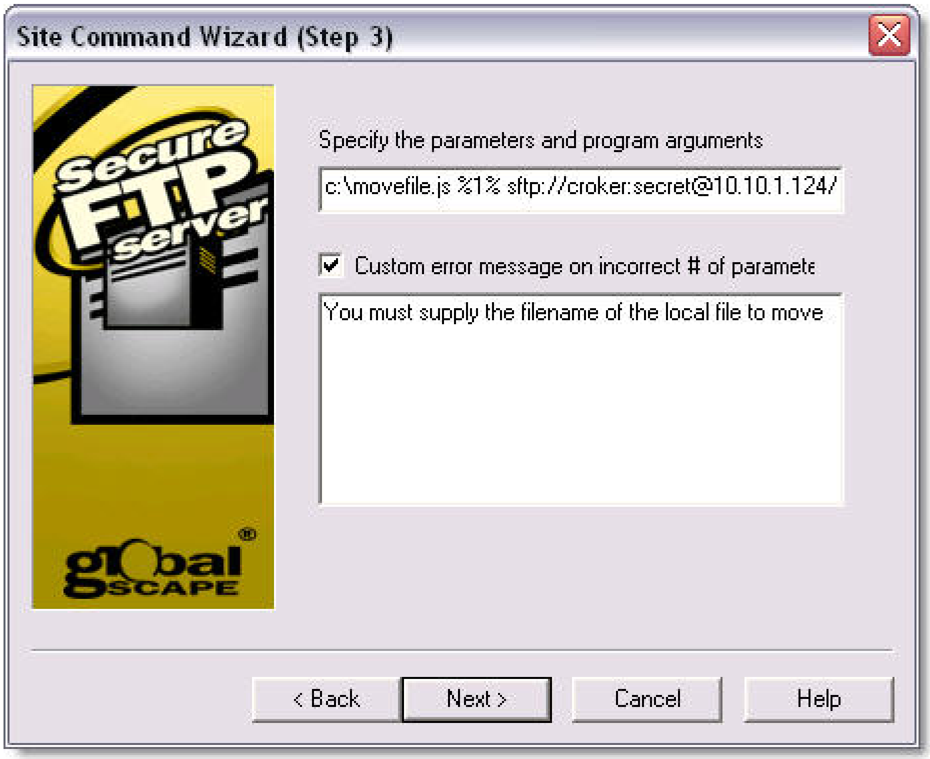
   a. In the **Specify the parameters and program arguments** text box enter the path to the **movefile.js** file.

      Parameterize the command to allow for a variable local file name . Assign permissions such that the brokering user may execute this command. As an example, the text box could read: c:\movefile.js %1% sftp://broker@10.10.1.124/mirror/

   b. The next parameter identifies the file to move.

   c. The last parameter is the destination of the designated file.

   d. The example also shows that a custom error message will be displayed in case someone invokes this command without supplying a file name parameter.



Site Command Wizard (Step 3)

Specify the parameters and program arguments

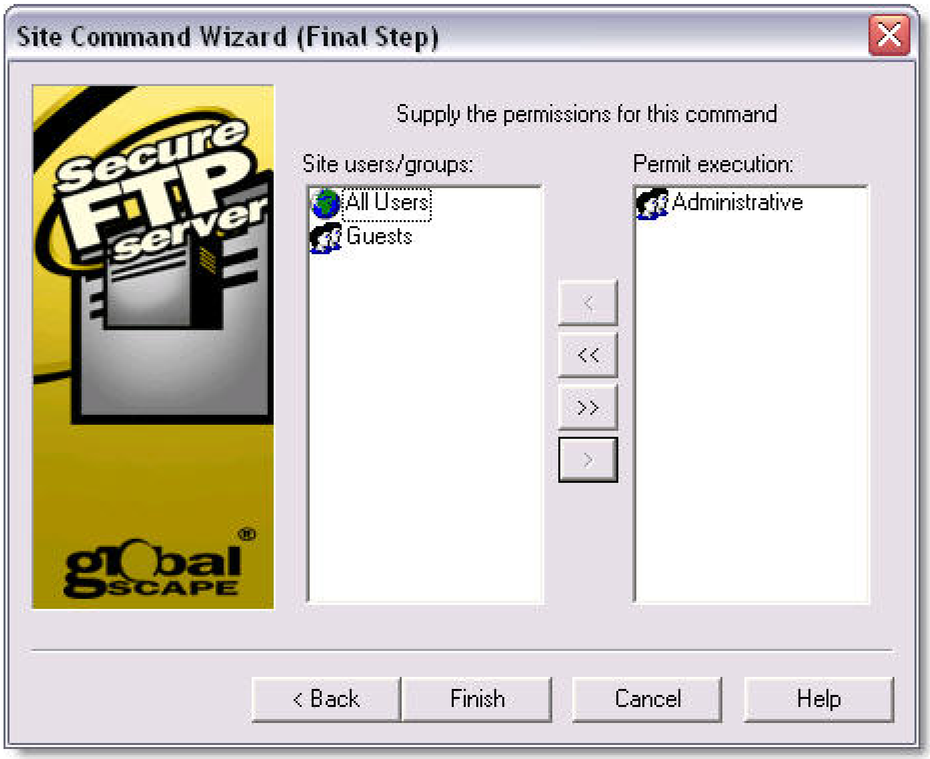
c:\movefile.js %1% sftp://croker:secret@10.10.1.124/

☑ Custom error message on incorrect # of paramete

You must supply the filename of the local file to move

7. The last screen of the command setup wizard sets permissions. The example indicates that only the **Administrative** group can access this custom command. This restriction will prevent a typical FTP user from invoking this function.

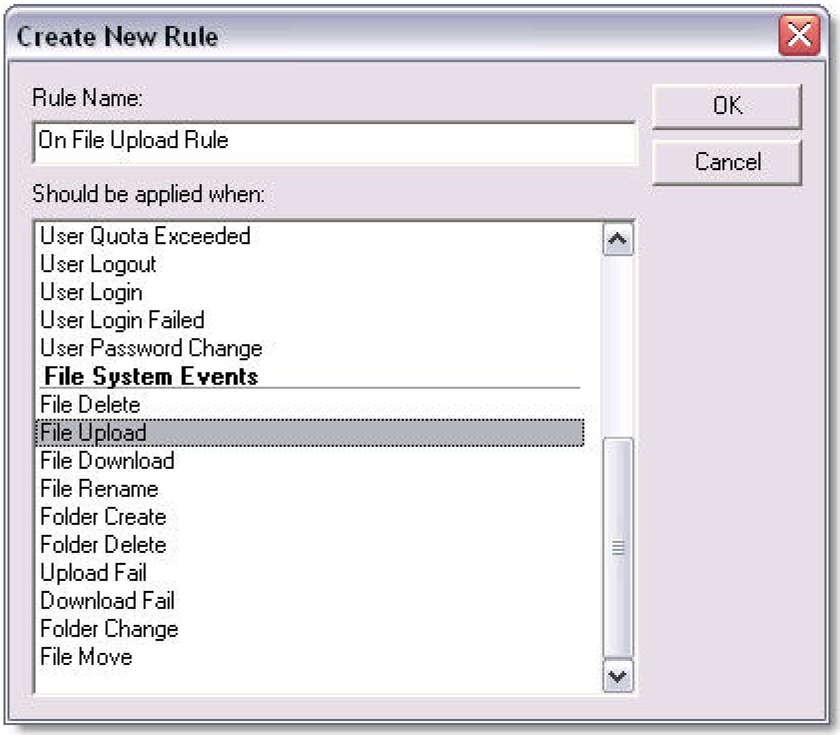
8. If desired, this custom command can be verified by logging into the server as an Administrative user and issuing the raw command "SITE MOVEFILE <filename>".



Event Rule Creation

This Event Rule creation step configures the trigger to launch the site command. Event Configuration Steps:

1. Launch the Secure FTP Server GUI Administrator on the Server machine and connect to the local Secure FTP Server.
2. Choose **Create New Event Rule** from the menu (or right-click on the Site node in the treeview and choose **New Event Rule**). Selecting this option opens a New Rule wizard.
3. Enter the name of the new rule in the **Rule Name** text box.
4. In the next section, select the **File Upload** entry under the **Server Events** category.


Create New Rule

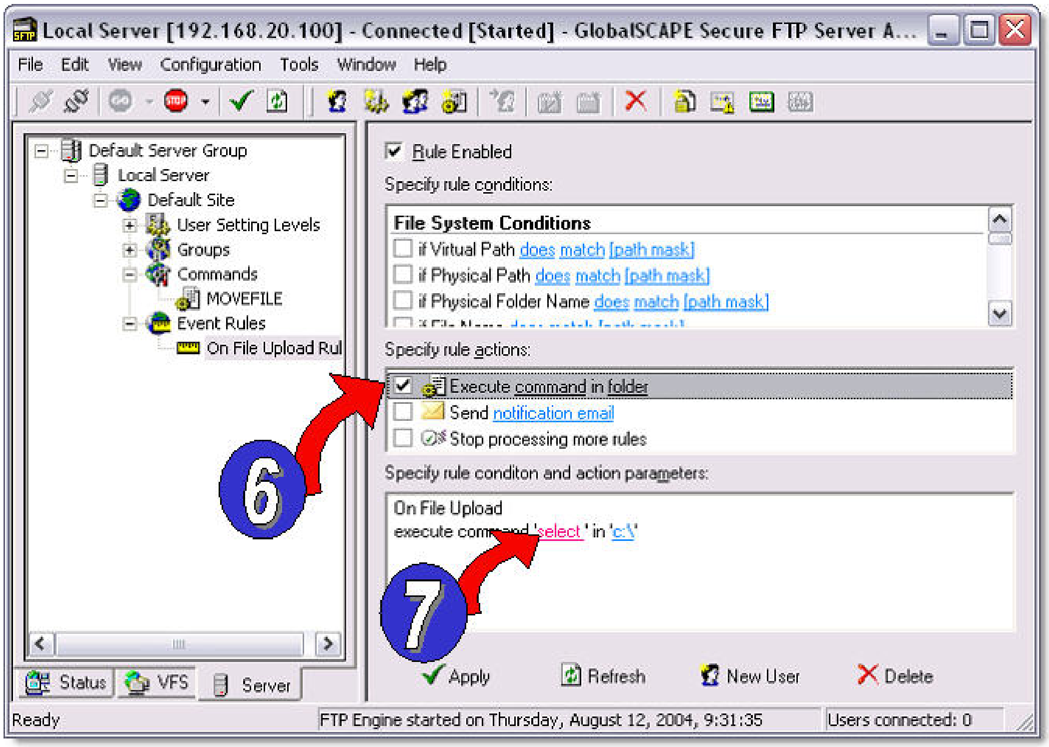
Rule Name:

On File Upload Rule

Should be applied when:

File Upload

5. Press **OK** to continue.


Local Server [192.168.20.100] - Connected [Started] - GlobalSCAPE Secure FTP Server A...

☑ Rule Enabled

Specify rule conditions:

File System Conditions

☐ if Virtual Path does match [path mask]

☐ if Physical Path does match [path mask]

☐ if Physical Folder Name does match [path mask]

Specify rule actions:

☑ Execute command in folder

☐ Send notification email

☐ Stop processing more rules

Specify rule conditon and action parameters:

On File Upload
execute command 'select' in 'c:\'

Ready | FTP Engine started on Thursday, August 12, 2004, 9:31:35 | Users connected: 0

6. On the Event Rule screen that appears, place a checkmark next to the **Execute command in folder** in the **Specify rule actions** area.
7. In the **Specify rule condition and action parameters** section, click on the hyperlink select.
   a. In the **Custom Command** window that appears, select the **MOVEFILE** command from the choices in the **Select command** drop down menu.

b. In the **Specify command parameters** text box enter the context variable for **Physical Path** or select **Physical Path** from the **Available Tags** selection box.

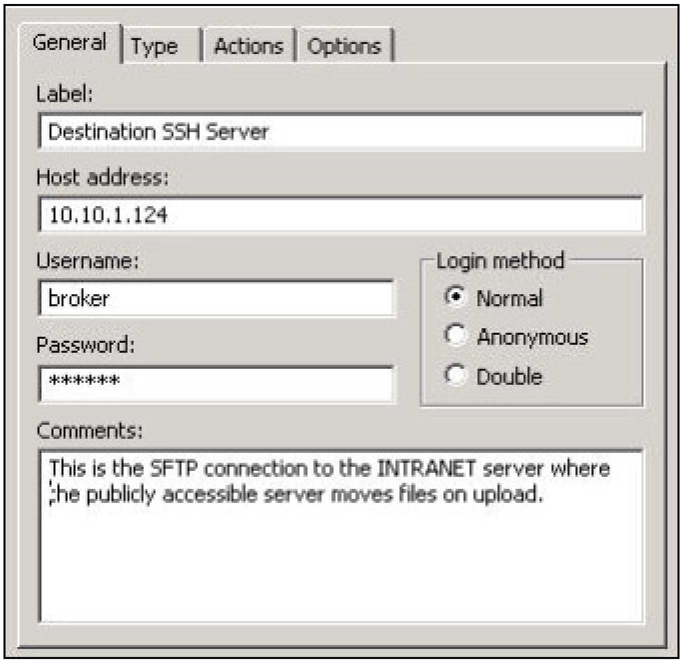
c. For the working folder, select the same folder that houses the **movefile.js** script.
d. Click **OK** to save the entries and close the window.

CuteFTP Professional Configuration

1. Configure the firewall to allow bi-directional communication over port 22 for the Intranet server and the internal IP address of the publicly accessible server.

   The publicly accessible server must have the CuteFTP Professional Transfer Engine installed and configured to interact with the Intranet FTP server.

2. Install and run CuteFTP Professional on the publicly accessible server.
3. Create a connection to the Intranet server and provide the appropriate public key ("Identity File") for authentication. Establish the connection to confirm that it works.



Label: Destination SSH Server

Host address: 10.10.1.124

Username: broker

Password: ******

Login method: ● Normal ○ Anonymous ○ Double

Comments: This is the SFTP connection to the INTRANET server where the publicly accessible server moves files on upload.

NOTES:
Secure FTP Server stores a list of trusted SSH server public keys in a file called **certs_ssh2.crt**

These public keys are found in the following folder:
%userprofile%\Application Data\GlobalSCAPE\CuteFTP Pro\6.0\Security
(where "%userprofile%" is the interactive user profile path, e.g., "c:\documents and settings\foo")
Since the CuteFTP Professional Transfer Engine will be launched as the user account under which the Secure FTP Server service is running, it is imperative that this file be in the appropriate location. If the Secure FTP Server service is running as a user account, place the file in that user's profile directory; if, however, the Secure FTP Server service is running as the **local system account** (which is the default configuration), you should place that file into the "All Users" profile area.

c:\documents and settings\all users\Application Data\GlobalSCAPE\CuteFTP Pro\6.0\Security
This concept also applies to related elements, such as public keys used to connect to a remote SSH server: CuteFTP Professional must be configured to use paths which are accessible by the account under which the Secure FTP Server service is running.

Conclusion

After this process is complete, every file that is uploaded to the publicly accessible Secure FTP Server will be securely brokered to the Intranet Secure FTP Server.

A variety of enhancements are imminently possible given the extensible nature of this solution, and are left as an exercise to the reader. Possible ideas include:

- Delete the local file after a successful transfer to the Intranet server.
- Parameterize the Custom Command and add additional Event Rules that transfer files to other locations or use different login credentials based upon path or filename information of the uploaded file.
- Extend the **movefile** script so that it sends an email notification upon successful transfer.
- Extend the Custom Command so that it invokes an **encryption** script prior to the invocation of the **movefile** script.

Script source: movefiles.js

```
/*===================================================================
=========
*= File: movefile.js
*= Created: 04/30/2004 GTH
*= Purpose:
*= Perform stress test suite of uploads to the Secure FTP Server
*= according to basic specs of DELL so that we can calculate the
*= average transactions per second / hour / day etc. that can be
*= expected by our server.
*= NOTES:
*= * this should use FTP Professional 6 TE or better.
*=
*= USAGE:
*= cscript stress_test <remote location> [username] [password]
*= EXAMPLE:
*= cscript movefile c:\temp.dat
ftps://foo:bar@192.168.20.64/uploads/
*= NOTE
*=    REQUIRES WINDOWS SCRIPTING HOST 5.0 (comes with IE 5.0) OR
HIGHER
*=
*= You can use connections of the following form for other protocols
*= ftp:// FTP
*= ftps:// FTP over SSL (implicit)
*= sftp:// FTP over SSH (SFTP)
*= http:// HTTP
*= https:// HTTP over SSL
*= file:// Filesystem (local or UNC path; also supports web sharing)
*===================================================================
========
*/
/*=========================== CONSTANTS
===========================*/
var _MIN_ARGS = 2; // we need at least three arguments to run.
var _MAX_ARGS = 4;
/*============ VARIABLES ============*/
var oTE; // this will hold the TEConnection object
var sLocalFile; // which local file to upload to the remote server.
(Param(0))
var sConnectionString; // the host name of the remote server to
connect to. (Param(1)
// and possibly 2&3)
var sLogin; // the FTP login to the remote host (Param(2))
var sPassword; // the FTP password to the remote host (Param(3))
var sRemoteFolder; // the remote folder to which we upload.
(Param(1))
/* ===== ENTRY POINT ===== */
main();
//
// Function getArgs()
// This function will get the command line arguments passed to this
script,
// and populate the appropriate global variables.
// Builds the ConnectionString from remote location and, optionally,
username/password.
// Returns FILE or TE to indicate which method of movement is
required for the operation.
// TE refers to the PRO 6 TE, while FILE refers to simple file
movement (FileSystemObject).
// Throws an exception if things are not quite right.
//
function getArgs()
{
var oArgs = WScript.Arguments;
var sTempConnection = "ftp://";  // assume FTP unless otherwise
specified.
var sConnectionPart;
var iPos;
var sMethod;
if ( oArgs.length >= _MIN_ARGS )
{
sLocalFile = oArgs(0);
sConnectionString= oArgs(1);
```

```
// any more arguments supplied?
// WScript.Echo("Argument Length = " + oArgs.length );
iPos = sConnectionString.indexOf("..//");
sMethod = sConnectionString.substr( 0, iPos - 1 );
// WScript.Echo("\n\tsMethod is '" + sMethod + "'\n");
// Determine if we are going over FILE system; we already assume FTP
if not.
if ( sMethod.toLowerCase().indexOf("file") != -1 )
{
sMethod = "FILE";
}
else
{
sMethod = "TE";
}
// WScript.Echo("\n\tsMethod is '" + sMethod + "'\n");
// If we are using the TE to transfer files, we must look at
additional
// command line arguments to ensure we have what we need.
if ( oArgs.length > _MIN_ARGS )
{
// if so, must be exact expected amount; we don't support
// partial parameter lists.
if ( (oArgs.length == _MAX_ARGS) && ( sMethod.equals("TE") ) )
{
sLogin = oArgs(2);
sPassword = oArgs(3);
iPos = sConnectionString.indexOf("..//");
if ( iPos >= 0 )
{
// remote location specifies a protocol, so use that
// instead of assumed ftp://
sConnectionPart = sConnectionString.substr( iPos + 2 );
sTempConnection = sConnectionString.substr( 0, iPos + 2 );
}
else
{
sConnectionPart = sConnectionString;
}
sTempConnection += sLogin + ":";
sTempConnection += sPassword + "@";
sTempConnection += sConnectionPart;
}
else if ( sMethod.equals("FILE") )
{
```

```
// parse off the filepath part of the destination spec
iPos = sConnectionString.indexOf("..//");
if ( iPos >= 0 )
{
sTempConnection = sConnectionString.substr( 0, iPos + 2 );
}
else
{
sTempConnection = sConnectionString;
}
}
else
{
throw "Invalid arguments!"; // too many args
}
sConnectionString = sTempConnection;
}
return sMethod;
}
throw "Invalid arguments!";
}
//
//
// Function upload
// This function will upload the file to the current remote folder
// sLocalFile is the source file on the local machine.
// sRemoteFile is the destination.  It will be replaced if extant.
//
function upload( sLocalFile, sRemoteFile )
{
if ( oTE.RemoteExists( sRemoteFile ) )
{
oTE.RemoteRemove( sRemoteFile );
}
oTE.Upload( sLocalFile, sRemoteFile , 1 );
}
//
// Function Connect
// This function instantiates the TE and connects to the server.
// returns TRUE if successful.
function connect()
{
var sListing_local, sListing_remote; // compare listings to see if
anything is downloaded
```

```
var arLocal, arRemote, arDiff; // arrays of the listings, for easier
comparison.
var i,j,k,l;
oTE = new ActiveXObject("CuteFTPPro.TEConnection")
oTE.Host = sConnectionString;
try
{
oTE.Connect();
}
catch(e)
{
WScript.Echo("Connection Error ('" + sConnectionString + "'): " +
e);
}
return (oTE.IsConnected);
}
//
// Function usage
// This function shows the usage of this script.
//
function usage()
{
WScript.Echo("Incorrect usage! Usage:\n");
WScript.Echo("cscript movefile <file to move> <remote location>
[username] [password]\n");
WScript.Echo("Example:\n");
WScript.Echo("cscript movefile c:\\temp.dat
ftps://foo:bar@192.168.20.64/uploads/\n");
}
//
// Function doMoveFTP
// This function will invoke the CuteFTP PRO TE and move the
// local file to the remote location, both specified in the
// command line argument(s).
//
function doMoveFTP()
{
sMethod = connect();
sRemoteFullFile = oTE.RemoteFolder + sLocalFile.substr(
sLocalFile.lastIndexOf('\\') + 1);
WScript.Echo("\nMoving '" + sLocalFile + "' to '" + sRemoteFullFile
+ "'\n");
upload( sLocalFile, sRemoteFullFile );
// sTemp = oTE.GetResult();
oTE.Disconnect();
oTE.Close( "EXIT" );
```

```
}
//
// Function doMoveFileSystem
// This function will use the Scripting.FileSystemObject to move
// the local file to the remote location, both specified in the
// command line argument(s).
//
function doMoveFileSystem()
{
var oFS, sDestFile;
sDestFile = sConnectionString.substr( "file://".length );
WScript.Echo("\nCopying '" + sLocalFile + "' to '" + sDestFile +
"'\n");
oFS = WScript.CreateObject("Scripting.FileSystemObject");
oFS.CopyFile( sLocalFile, sDestFile, true); // force an overwrite of
the destination.
}
//
// Function main
// This is the main function.
//
function main()
{
var i, tm1, tm2;
var sRemoteFullFile;
var sTemp;
var sMethod;
if ( sMethod = getArgs() )
{
// WScript.Echo("Your movement method is '" + sMethod + "'");
if ( sMethod == "TE" )
{
doMoveFTP();
}
else if ( sMethod == "FILE" )
{
doMoveFileSystem();
}
}
else
{
usage();
}
}
```

14

Index