

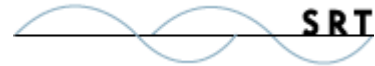


TitanFTP

S E R V E R

Administrator's Guide
Version 8

February 2010



Notices

Copyright 2010 South River Technologies, Inc. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement, OEM, or reseller agreement. The software may be used or copied only in accordance with the terms of those agreements. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of South River Technologies, Inc.

South River Technologies[®], GroupDrive Collaboration Server[®], Cornerstone MFT[™], Titan FTP Server[®], DMZedge Server[™], and WebDrive[®] are trademarks of South River Technologies, Inc. in the U.S. and other countries. Microsoft, Windows, Windows NT, Windows XP, and Windows Vista are trademarks of Microsoft Corporation, Inc. The names of other actual companies and products mentioned herein may be the trademarks of their respective owners. Any information in this document about compatible products or services should not be construed in any way to suggest SRT endorsement of that product or service.

Contact Information

South River Technologies, Inc.
2635 Riva Road
Suite 100
Annapolis, Maryland 21401
USA

Telephone: 410-266-0667
Fax: 410-266-1191

Sales Office e-mail: sales@southrivertech.com
Online Support: <http://www.srthelpdesk.com>
Support e-mail: support@southrivertech.com
Corporate Web site: www.southrivertech.com

Office Hours: 8:30 A.M. to 5:30 P.M. Eastern Time, GMT-5:00

Table of Contents

Notices.....	ii
Table of Contents.....	iii
Getting Started.....	6
System Requirements.....	6
Installation and Removal.....	7
Starting the Titan Service.....	8
Program Group Items.....	9
Terminology.....	10
Launching the Administrator.....	11
Domain Overview.....	13
Configuring the Domain.....	13
Domain Activity Tab.....	15
Domain IP/Ports In Use Tab.....	16
Configuring Servers.....	17
Servers Overview.....	17
Creating New Servers.....	18
Deleting Servers.....	19
Backing Up Servers.....	19
Restoring Servers.....	19
Server Properties.....	20
Servers General Tab.....	20
Server Advanced Tab.....	21
Server User Authentication Tab.....	23
Server UNC Accounts Tab.....	24
Server Email Server Tab.....	25
Server FTP Tab.....	26
Server Web Access.....	27
Server Connections.....	28
Server Connections General tab.....	28
Server Connections Advanced tab.....	29
Server Connections IP Access Tab.....	31
Server Connections Upload/Download Ratios Tab.....	32
Server Custom Messages Tab.....	33
Server Files/Directories.....	34
Server Files/Directories Tab.....	34
Server Files/Directories Directory Access Tab.....	36
Server Files/Directories Virtual Folders Tab.....	38
Server Files/Directories Disk Quotas Tab.....	40
Server Security.....	41
Server FTPS/SSL Tab.....	41
Server SFTP/SSH Tab.....	43
Server Flood Protection/DoS Tab.....	45
Server Logging.....	46
Server Log Tab.....	46
Server Log Settings Tab.....	47
Server Statistics Tracking Tab.....	49
Server Activity.....	51
Server Activity Tab.....	51
Server Spy User Tab.....	52
Server Spy Session Tab.....	53
Server Events.....	54
Server Event Handlers Tab.....	54
Server Flagged Events Tab.....	55
Configuring Groups.....	56
Groups Overview.....	56
Creating New Groups.....	56
Deleting Groups.....	57
Adding Users to Groups.....	57
Removing Users from Groups.....	58

Group Properties.....	59
Groups tab.....	59
Group General Tab.....	60
Group Users Tab.....	61
Group FTP Tab.....	61
Group Connections.....	62
Group Connections General tab.....	62
Group Connections Advanced tab.....	63
Group Connections IP Access Tab.....	65
Group Connections Upload/Download Ratios Tab.....	66
Group Custom Messages Tab.....	67
Group Files/Directories.....	68
Group Files/Directories Tab.....	68
Group Files/Directories Directory Access Tab.....	70
Group Files/Directories Virtual Folders Tab.....	72
Group Files/Directories Disk Quotas Tab.....	74
Group Security.....	75
Group FTPS/SSL Tab.....	75
Group SFTP/SSH.....	75
Configuring Users.....	76
User Authentication Overview.....	76
Creating New Users.....	76
Deleting Users.....	77
User Properties.....	78
Users Tab.....	78
User General Tab.....	79
User Groups Tab.....	81
User FTP Tab.....	81
User Connections.....	82
User Connections General tab.....	82
User Connections Advanced tab.....	83
User Connections IP Access Tab.....	85
User Connections Upload/Download Ratios Tab.....	86
User Custom Messages Tab.....	87
User Files/Directories.....	88
Files/Directories Tab.....	88
User Directory Access Tab.....	90
User Virtual Folders Tab.....	92
Files/Directories Disk Quotas Tab.....	94
Security.....	95
User SSL Tab.....	95
User SFTP/SSH Tab.....	96
Advanced Features.....	97
Event Handling.....	97
Inherited Settings.....	113
Shared Attributes.....	114
Custom Message Variables.....	119
CRC File Integrity Checking.....	123
User Authentication Options.....	124
Remote Administration.....	125
SSL Support.....	126
SFTP Support.....	127
Virtual Folders.....	128
srxCfg Command Line Utility.....	130
srxCOM Interface.....	157
Server Commands and Return Codes.....	182
Standard Commands.....	182
Advanced Commands.....	183
SITE Commands.....	183
FTP Return Codes.....	184
SFTP Commands Overview.....	187
SFTP Return Codes Overview.....	188

- Standard FTP Commands 190
- Advanced FTP Commands..... 213
- Tutorials 232
 - Creating a New Server 232
 - Using Titan with a Router or Firewall..... 238
 - Event Handlers 239
- Troubleshooting..... 249
 - FAQ - Frequently Asked Questions 249
 - SRT Knowledgebase 253
 - Reporting Problems 253
- Contact Information 254
- Compare Feature Sets 255
- Index 256

Getting Started

System Requirements

Supported Platforms

- Windows 7®, 32-bit or 64-bit
- Windows Vista®, 32-bit or 64-bit
- Windows Server 2008®, 32-bit or 64-bit
- Windows Server 2003®, 32-bit or 64-bit
- Windows XP Pro®, 32-bit or 64-bit

Requirements & Limitations

Titan FTP Server is a multi-threaded, dynamic FTP Server for the Windows operating system. While it is designed to handle an unlimited number of user connections and servers, it is limited by the resources of the computer; most notably, those limitations imposed by the Windows Sockets (WINSOCK) Library.

The minimum hardware requirements for Titan are:


- A 2GHz Pentium® class processor
- 2GB of RAM is required, 4GB is recommended
- Minimum 100MB of disk space for the Titan program and 100MB for each FTP Server configuration
- Minimum SVGA (800x600) resolution is required to run the Administration program

The Titan Web browser interface uses industry standard HTML and JavaScript and should run properly on most browsers. However, due to the minor differences in how various browsers interpret HTML, some graphics and features may not display correctly.

Browser Requirements

The Titan Web User Interface has been tested on the following browsers:

- Microsoft Internet Explorer v7.0 or later
- Firefox v3.5 or later
- Safari v4.0 or later



- All Windows Service Packs and Hot Fixes must be applied to the computer prior to installing Titan FTP Server.
- The Titan Web Interface is an add-on module. For more information, contact sales@southrivertechologies.com.
- For more information about the Titan Web interface, see the [Titan Web Interface User's Guide](#).

Installation and Removal

Installing Titan FTP Server

1. Make sure that you have the most recent version of Titan FTP Server. You can download the most recent version from our Web site at <http://www.titanftp.com/>
2. Double-click the installation file to start the installation process. **NOTE:** If you are installing Titan FTP Server on **Windows 7, Windows Vista, or Windows Server 2008**, you must right-click the **Installation/Setup file** and select **Run As Administrator**.
3. When the installation process is complete, you must **restart Windows**.
4. After Windows has restarted, the Titan FTP Service will be running. The Titan FTP Service runs as a system service and starts when the operating system loads. You can monitor the status of the Titan FTP Service using the Windows Services applet in the Administrative Tools folder in the Windows Control Panel.
5. Once Titan FTP Server has been installed, you can begin to configure your FTP Servers. To configure your servers, use the Titan Administrator program located in the Titan FTP Server program group.

Uninstalling/Removing Titan FTP Server

1. From the Windows Start menu, click **Control Panel** to open the Windows Control Panel.
2. Double-click on the **Add/Remove Programs** icon in the Control Panel.
3. Select **Titan FTP Server**.
4. Follow the instructions on the dialog screens to remove Titan from your machine.
5. Close the **Add/Remove Programs** applet.

You must **restart Windows** to have Titan completely removed from your machine.

A warning icon consisting of two red exclamation marks inside a square frame.


- If you are installing Titan FTP Server on Windows 7, Windows Vista or Windows Server 2008, you **MUST** run the installer AS ADMINISTRATOR; simply using an account with Administrative Privileges is not sufficient. Right-click on the Installation/Setup file and select **RUN AS ADMINISTRATOR**.
- The 32-bit version of Titan must be installed on a 32-bit operating system. The 64-bit version of Titan must be installed on a 64-bit operating system.

Starting the Titan Service

Titan FTP Server is designed to run as a system service or background process. You can configure Titan FTP Server so that it starts when Windows starts or you can configure it to be started manually.

To check and/or modify the startup setting for the Titan Service

1. Click **Start** and then click **Control Panel**.
2. Double-click **Administrative Tools** and then double-click **Services**.
3. Right-click on **Titan FTP Server Daemon**, and click **Properties**.

A small icon of a red pushpin on a silver clip, used to denote a note or important information.

The Titan Service will be installed to run under the context of the standard **LocalSystem** or **LocalService** Windows User Account. When a user connects to the Titan FTP Server, all file access is normally performed by the NT User Account on behalf of the logged in FTP client user. This means that if there are files on a UNC that Titan will be accessing, the Titan Service must be re-configured so that it uses an NT User Account that has proper NTFS permissions to the UNC share.

Program Group Items

The Titan FTP Server Installation program will install Titan FTP Server on the local computer. As part of the installation process, the Titan FTP Server Program Group will be created.

A standard Titan FTP Server installation produces a program group containing the following entries:

- **Administrator** - Launches the Titan FTP Server Administrator program. This is the main program used to configure and administer all aspects of the Titan FTP Server. Using the Administrator, you can connect to the Titan FTP Service, add/delete/modify or monitor FTP Servers. The Administrator program is also used to configure groups, users, permissions, and access options. The Titan installer also adds a shortcut to the Administrator program on the Windows desktop.
- **Buy Titan FTP Server** - Launches your browser and opens the Titan FTP Server Pricing/Purchasing page of our Web site. Titan FTP Server can be purchased online and downloaded immediately.
- **Check for Program Update**- This icon, which is installed with the retail version only, will allow users to go online and check for new releases of the software.
- **Online Help** - Launches the Titan FTP Server Help System. The Titan FTP Server Help System describes all aspects of using Titan and the Administrator program.
- **Pricing** - Launches your browser and opens the Titan FTP Server pricing page of our Web site.
- **Quick Start Guides** – Launches your browser and opens the Quick Start Guide and White Paper page on our Web site.
- **Release Notes** - Launches the Release Notes. Please review the Release Notes each time you install a new/upgrade build of Titan FTP Server. The Release Notes contain important information about bug fixes, known issues, and other important information not found in the Online Help.
- **Technical Support** - Launches your browser and opens the main Support page.
- **Titan FTP Server Homepage** - Launches your browser and opens the Titan FTP Server home page.
- **Titan Tray Applet** - Launches the tray application. You can use the tray application to monitor the status of the Titan service.
- **Uninstall** - Launches the Titan FTP Server Uninstaller. This program will uninstall Titan and remove all components from your computer.
- **Version History** - Launches your browser and displays the complete version history for Titan FTP Server. This page shows past versions and the features, changes, and fixes that appeared in those versions.

Terminology

- **Administrator** - The Administrator program allows you to configure all aspects of the Titan FTP Server. Installed as part of a standard Titan FTP Server installation, the Administrator program displays the configuration elements/options for each Domain, Server, Group, and User. The Administrator program is used to configure servers on both the Local and Remote Domains.
- **Titan FTP Server Service** - The Titan FTP Server Service is the main program/service/daemon that runs on the computer and manages the various FTP server instances that have been configured. In the Titan FTP Server Administrator, the Titan FTP Server Service is represented by the root/top node in the tree, which is labeled **Titan FTP Server Domains**.
- **Domain** - A domain is defined as the physical computer on which the Titan FTP Server Service is running. The Titan Administrator program can connect to the local domain or to a remote Titan domain. For each domain, you can define zero or more FTP server instances. Each FTP server instance is identified by a unique IP Address/Port combination. In the Titan FTP Server Administrator, domains are represented by a blue computer monitor icon located directly under the Titan FTP Server Domains node in the menu tree.
- **Local Domain** - The local computer on which the Titan FTP Server Service is executing. In most instances, the Titan FTP Server Administrator operates on the local domain. In the Titan FTP Server Administrator, the local domain is represented by a blue computer monitor icon located directly under the Titan FTP Server Domains node in the menu tree. There will only be one local domain listed.
- **Remote Domain** - A remote computer on which the Titan FTP Server Service is executing. You can connect to a remote domain using the Titan FTP Server Administrator program. See [Remote Administration](#) for more information.
- **FTP Server** - The Titan FTP Server Service manages the various FTP server instances that are defined within the local domain. Each FTP server instance is identified by a unique IP Address/Port combination, providing the ability to have multiple FTP servers all running at the same time. For example, you can configure a production FTP server to be listening on IP address 192.168.1.1 (Port 21), and also have a staging server (or test server) listening on Port 2100. If your computer has multiple NIC interfaces (192.168.1.1 and 205.1.2.3), you can set up an FTP Server to listen on 192.168.1.1:21 and configure another FTP server instance to listen on 205.1.2.3, Port 21.
- **Groups** - For each FTP server instance, Titan FTP Server provides the ability to define one or more groups. Groups are a way to categorize users who share common attributes. Define a group, add users, and then set various permissions and attributes that will be applied to any user in that group. By default, each FTP Server instance is pre-configured with a general group called **Everyone**. This is a system level group and cannot be deleted. All users are members of the **Everyone** group, so be careful when you are adjusting the permissions for the **Everyone** group.
- **Users** - For each FTP server instance, the Titan FTP Server Administrator gives you the ability to add new users and to configure the access rights granted to these users. By default, Titan automatically creates an **Anonymous** user and grants it access to the FTP server. The Anonymous user can be disabled by the System Administrator.
- **WebDrive®** - South River Technologies' **WebDrive** FTP Client; the FTP Client that maps a Drive Letter to FTP, SFTP and WebDAV Servers. FTP Client through a virtual drive.

Launching the Administrator

The Titan FTP Server Administrator program is used to configure **FTP Servers**, **Groups**, and **Users**, both locally and remotely.

To start the Administrator program, double-click the **Administrator** icon in the Titan FTP **Program Group**. There is also a shortcut to the Administrator program on your Windows desktop.

The Administrator program is designed as a standard Windows application and contains a split screen with two panes separated by a vertical resizing bar. The left pane of the screen, or **Tree Pane**, displays the overall Titan hierarchy of **Domains**, **Servers**, **Groups**, and **Users** (see [Terminology](#)). The right pane of the screen, or **Tab Pane**, displays configuration information and options based on what is currently selected in the **Tree Pane**. When you click on different items in the **Tree Pane**, the **Tab Pane** changes and different **Dialog Tabs** are displayed. Each **Dialog Tab** in the **Tab Pane** displays information and options relevant to the selected item.

Apply/Revert

Many of the Dialog Tabs in the Administrator program have two buttons labeled **Apply** and **Revert**, located at the bottom of the screen. **Apply** and **Revert** provide the ability to apply new settings or revert to previously saved settings. When a configuration option is modified on a Dialog Tab, the **Apply** and **Revert** buttons are enabled. To save changes, click **Apply**. If you make changes that you do not want to keep, click **Revert** and the current Tab will be reloaded with the currently saved settings. **Note:** If you make changes on a Dialog Tab, and then switch to another Dialog Tab, or select a different item in the Tree Pane, your changes are applied and saved automatically.

Real Time Effectiveness

In most cases, modifications that you make at the **Server**, **Group**, or **User** level become effective once you click **Apply**. When a configuration option is modified at any level, the Titan FTP Service is notified that the configuration has been modified and Titan will reload that information at the next available opportunity. For example, if someone is logged onto the server as **Anonymous**, and the Administrator disables the Anonymous account, the Titan FTP Service will accept the change and disconnect the anonymous user from the system.

Navigation

The Administrator program uses standard Windows navigation keys.

- **Menu Bar** - To access the menu options in the Administrator, hold down the <ALT> key to activate the menu bar, and then click on the underlined letter of the desired menu item.
- **Toolbar** - The toolbar contains buttons for some of the more common features of the Titan FTP Server Administrator. Different toolbar buttons are enabled and disabled depending on the item that you have selected in the Tree Pane. Commands associated with the tools on the toolbar can also be accessed from the main menu bar.

- **Context Menus** - You can also access Tree Item command options by right-clicking on that item. When you right-click on an item in the Tree, a context menu is presented that contains options specific to the selected item.

- **Switching Panes** - Use <F6> to switch or toggle between the Tree Pane and the Tab Pane. You can also select the desired pane by clicking the pane with your mouse.

- **Switching Tabs** - Use the left and right arrow keys on your keypad to switch between the various tabs in the Tab Pane.

- **System Menu** - The Administrator program has the Windows standard minimize, maximize, restore, and close functionality built into the System menu. Since some of the configuration dialogs, such as the **Server Configuration** dialog, have many tabs, it may be easier to maximize the Administrator so that all of the tabs are visible.

Domain Overview

In Titan, the term domain refers to the physical computer on which Titan is installed. The primary use of the domain is to provide a grouping for the FTP server or servers that run on that computer. You will use the [Titan FTP Server Administrator](#) program to connect to the domain and to configure your FTP servers.

The first time the Administrator is executed, the **Local Domain Wizard** will be launched. The **Local Domain Wizard** ensures that your computer is properly configured. Along with other configuration options, you will need to specify the username and password that you will use for local administration. Save this information because each time you run the Administrator program and connect to the local domain, you will be prompted for the username and password for authentication.

Configuring the Domain

Local Domain Properties


- **Local Domain Name** - This name represents the local domain. By default, this is the same name as the physical computer. However, you can change this name to any text name. The local domain is not displayed to the client by default, but you can configure custom messages to display the name of the domain. See [Custom Variables](#) for more information. Once you have connected, the domain name is displayed in the Tree Pane of the Administrator.
- **Local Domain Description** - This text box provides you the option to further describe the domain.
- **Data Directory** - The domain **Data Directory** setting defines the default storage location for FTP Server data. This value is used to prime the FTP Server **Data Directory** entry in the **FTP Server Wizard**. For each new FTP Server, the domain data directory and the new server name will be concatenated to produce the full path to the data directory where the FTP Server data will be stored. This value can be either a fully qualified path such as **C:\Mydata** or a UNC name such as **\\Server\Share\MyData**.
- **Log Directory** - The domain **Log Directory** setting defines the default storage location for FTP Server logs. This value will be used to prime the FTP Server **Log Directory** entry in the **FTP Server Wizard**. For each new FTP server, the domain log directory and the new server name will be concatenated to produce the full path to the log directory where the FTP Server logs will be stored. This value can be either a fully qualified path such as **C:\MyLogs** or a UNC name such as **\\Server\Share\MyLogs**.
- **Start FTP Service when Windows Starts** - This option allows you to configure whether or not the Titan FTP Server Service will start automatically when Windows starts. If this option is enabled, the Titan FTP Server Service starts automatically when Windows starts. Once the Service starts, any FTP servers that you have configured to start when the Titan Service starts will also launch. The Titan FTP Server Service is installed as a Windows Service and follows the rules for starting automatically.
- **Run Tray Applet when Windows Starts** - This option allows for the automatic launching of the Tray applet used to **display/start/stop** the FTP Service.

Local Administration Properties

- **Administrator Username** - Enter the username that will be used to log on to the local domain. This username can be any combination of letters/numbers, but cannot contain spaces. The maximum limit for the administrator username is 128 characters. This username is **case sensitive**.
- **Administrator Password** - Enter the password that will be used in conjunction with the Administrator username to confirm access to the local domain. The password can be any combination of letters/numbers, but cannot contain spaces. The maximum limit for the password is 128 characters. The Administrator password is **case sensitive**.
- **Local Administration Port** - The Titan FTP Server Administrator communicates with the Titan FTP Server Service using a connection on the **localhost/127.0.0.1 loopback** address. The local administration port specifies the port number to be used with the loopback address to be used by the Titan FTP Server Service for listening for the Administrator.

Remote Administration Properties

- **Allow Remote Administration** - You can use the Titan FTP Server Administrator to connect to a Titan FTP Server Service installed on a remote computer, either on your intranet, or over the Internet. Before you can connect to this local domain from a remote location, you need to enable the **Remote Administration** feature for the local domain. Enabling remote administration will allow you to connect to this local domain and make changes from another location.
- **Administration IP Address** - This is the address that the local domain will listen on for remote administration connections.
- **Administration Port** - This is the port that the local domain will listen on for remote administration connections. This port cannot be the same port that is used for local administration. For example, if you have remote administration enabled, the Titan FTP Server Service will open two separate listener sockets, one on the **loopback:localport** for local administration and one on the **remoteaddress:remoteport** for remote administration.

An icon of two orange pushpins on a grey background, indicating a pinned note or important information.

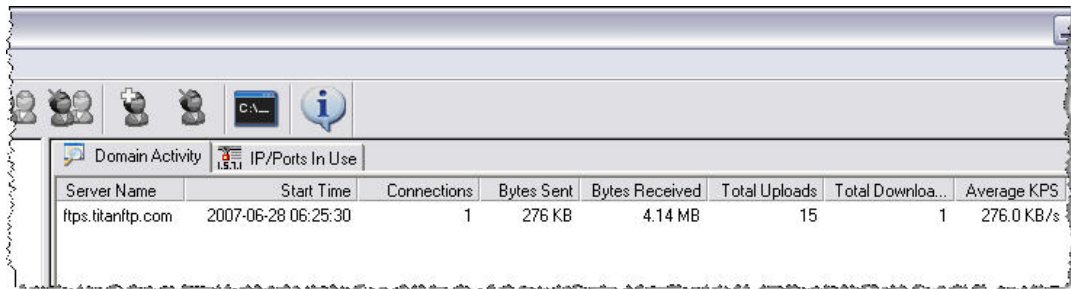
- Remote administration features are available in the Enterprise edition of Titan FTP Server.
- When specifying a Directory or Path in Titan, only use fully qualified Local Paths or UNC shares. Do not specify a mapped drive because these are not accessible from within the Titan Service.

Domain Activity Tab

The **Domain Activity** tab will display a list of servers currently configured and running on the domain to which the Administrator application is connected. If the Titan Administrator application is connected to a remote domain, servers configured and running on the remote domain will be displayed on this tab.

To access the **Domain Activity** tab, click **Domain Activity** in the tree pane and then click the **Domain Activity** tab.

For each server configured and running on the domain, this list will display the time the server was started, in local time, the number of active connection, total number of bytes sent from the server to the client(s), the total number of bytes received by the server from the client(s) since the server was started, the total number of files uploaded to the server, the total number of files downloaded from the server, and the overall average Kilobytes per second processed (sent and received) during execution.



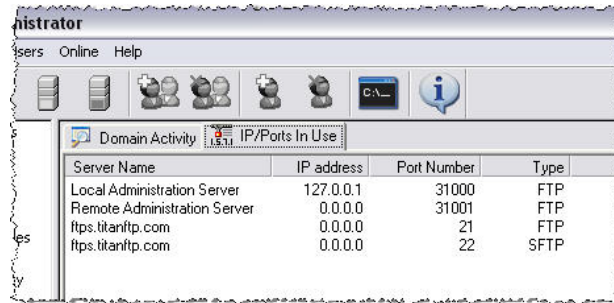
Server Name	Start Time	Connections	Bytes Sent	Bytes Received	Total Uploads	Total Downloa...	Average KPS
ftps.titanftp.com	2007-06-28 06:25:30	1	276 KB	4.14 MB	15	1	276.0 KB/s

- **Server Name** - This is the name of the server as defined on the domain.
- **Start Time** - This column contains the date and time that the server was started, in local 24-hour time.
- **Connections** - This column displays the total number of active connections on the server.
- **Bytes Sent** - This column displays the total number of bytes that have been sent by the server to the various client connections.
- **Bytes Received** - This column displays the total number of bytes that have been received by the server from the various client connections.
- **Total Uploads** - This column displays the total number of files that have been successfully uploaded to the server since the server was started.
- **Total Downloads** - This column displays the total number of files that have been successfully downloaded from the server since the server was started.
- **Average KPS** - This column displays the average kilobytes per second for data transferred to and from the server since the server was started.

Domain IP/Ports In Use Tab

The **IP/Ports In Use** tab displays the IP addresses and ports that are currently in use by the Titan system.

To access the **IP/Ports In Use** tab, click **Domain Activity** in the tree pane and then click the **IP/Ports In Use** tab.



- **Local Administration Server** - This is an internal system server that is used to interact with the Local Titan **Administrator**. By default, this internal server will listen on IP address **127.0.0.1 (localhost)**, port **31000**.
- **Remote Administration Server** - This is an internal system server that is used to interact with the Titan Service from a remote Titan Administration console. If **Remote Administration** is enabled on the domain, this server will listen on **port 31001** for incoming connections from the Remote Titan Administration utility.

There will be zero or more servers listed, depending on how many servers are configured on this domain. For each server defined on the domain, FTP, FTPS and/or SFTP may be listed.

This utility will not show non-Titan IP addresses and ports that are in use. To display a complete list of all **IP addresses** and **ports** that are in use on the local computer, open a **command prompt** and use the **NETSTAT** utility. The **-b** command line argument can be used with **NETSTAT** to display the process/executable that is currently using the IP/port combination. This is a good way to diagnose which application may be using a port.



Use the command line utility **NETSTAT** with the **-b** argument to display a list of IP addresses and ports that are in use on the computer. The **-b** argument will display the executable that is using the IP address/port. This is a good way to diagnose which application may have a port in use.

Configuring Servers

Servers Overview

Titan supports the ability to configure multiple server instances under a single domain or physical computer. Each server instance listens on its own distinct IP address/port combination, which provides the ability to have a virtually unlimited number of servers running simultaneously.

Each server can be configured to store data in its own separate data directory, either on your local hard drive, or on a shared network drive. Titan FTP Server supports both standard DOS path syntax and UNC paths. **NOTE:** Do not use mapped drives or paths that point to a mapped drive because these are not accessible from the Titan service; use a UNC path instead.

To create a new server configuration, simply launch the New Server Wizard from the main menu, toolbar or via the right-click context menu for the domain under which you want the server to reside. The New Server Wizard will walk you through the steps involved in configuring your server. Once the New Server Wizard has completed, your new server will be created and you can start using it immediately. For more information about how to create new servers, see the [Creating a New Server Tutorial](#).

The New Server Wizard will help you setup the initial configuration for your server, but there are additional configuration options available once the server is created. Using the Administrator program, server properties can be modified on the fly.

1. Launch the **Titan Administrator**.
2. Connect to the **domain**.
3. Select the **server** in the Tree Pane.
4. Once the desired server has been highlighted in the Tree Pane, a list of Dialog Tabs will be displayed in the **Tab Pane**. The numerous configuration options are grouped under the various tabs. Select the **tab** and make the necessary changes. Once you have made the configuration changes, click **Apply** to save the changes. The Administrator will save the configuration changes and notify the Titan FTP Server Service that it needs to re-configure the server with the new settings.

[Deleting FTP Servers](#) is also available from the Administrator program. You cannot delete an FTP Server if it is running. You must stop the server before deleting it. Use the right-click context menu to stop the server. Once the server has been deleted, all associated Groups and Users will also be deleted from the system. **Note:** in order to protect from any possible data loss, the Titan FTP Server Administrator program will **not** delete the contents of the FTP Server Data directory or the Log directory; you must delete that information manually if it is no longer needed.

Creating New Servers

New servers can be configured at any time using the **New Server Wizard**. The **New Server Wizard** can be launched from within the Titan **Server Administrator**:

- Right-click the **Local** or **Remote Domain** icon in tree pane (left pane) and select **New Server Wizard** from the context menu.

or

- From the menu bar, select **Server > New Server Wizard**.

The **New Server Wizard** walks you through the steps required to configure a new server and set it online. You can modify the server configuration after the server has been created. The **New Server Wizard** helps you to set up the initial configuration for your server. There are more features and configuration options available once the server is created. Use the Titan **Server Administrator** to modify server properties.

To modify Server Properties Using the Titan Administrator:

1. Launch the **Titan Administrator**.
2. Connect to the **Domain**.
3. Select the **Server** in the **Tree Pane**.
4. Once the desired server has been highlighted in the **Tree Pane**, a list of **Dialog Tabs** will be displayed in the **Tab Pane**. Configuration options are grouped under the tabs. Select the **tab** and make the necessary changes. Once you have made the configuration changes, click **Apply** to save the changes. The **Administrator** will save the configuration changes and notify the Titan FTP Server Service that it needs to re-configure the server with the new settings.

Deleting Servers


Servers can only be deleted if they are not running. Make sure that you have stopped the server before attempting to delete it from the system. Use the right-click context menu to stop the server.

To delete an existing server using the Titan Administrator:

- Right-click on the desired server in left pane (tree pane) of the Titan Administrator and select **Delete Server** from the menu. You will be prompted to confirm the deletion of the server.

or

- Single-click on the desired server in the left pane of the Administration program. From the main menu, select **Server > Delete Server**. You will be prompted to confirm the deletion of the server.

A warning icon consisting of two red flags on a silver pole.

Deletion of a Server configuration is **permanent**. It is recommended that you only delete a server configuration if you are absolutely sure you will no longer be using it. Once a Server configuration has been deleted, all associated permissions, groups and user information will also be removed.

Backing Up Servers

To back up your Titan server configuration, select the **Server** in the tree pane. From the **Server** menu, select **Backup**. Use the drop-down arrow to change the default location. Click **Save** to save your .reg file.

Restoring Servers

To restore your Titan server configuration select the **Server** in the tree pane. From the **Server** menu, select **Restore**. Browse to your server configuration and click **Open**. Click **Yes** to accept the prompt or **No** to keep your current settings.

Server Properties

Servers General Tab

The **Servers General** tab is used to manage basic configuration options for the server.

To access the **Servers General** tab, click the **server** in the tree pane and then click the **Servers General** tab.

- **Server Name** - The name of the server.

- **Server Description** - A text description for the server.

- **IP Address** - The IP Address that the server will listen on. You can type a specific IP address, or you can select **Any Available IP Address**. We recommend that you select **Any Available IP Address** if your computer has a dynamic IP addressing scheme.

- **Data Directory** - The base directory where all FTP data will be stored. **Note:** Titan FTP Server runs as a **Windows Service** that, by default, does not have access to shared network resources or mapped drive letters because these are based on the currently authorized Windows user. If you plan to configure your server Data Directory to be a network resource or UNC, you will need to reconfigure the Titan FTP Server Service/Daemon manually so that it logs in using a Windows User account that has access to network resources. The default Windows Service account, **Local System**, does not have access to network resources.

- **Server Time Zone** - Allows you to specify the time zone that the server will "virtually" exist in. By default, this is the **local time zone**. Modifying the time zone will alter how the file dates and times are displayed to the user via an FTP client.

- **Adjust for Daylight Saving Time** - When selected, compensates for "Daylight Saving Time". This option is enabled by default.

- **Notes** - Allows you to enter any server-specific notes.

- **Start Server when Service Starts** - Select the check box to enable. When enabled, the server will automatically be started when the Titan FTP Server starts. This option is enabled by default.

Server Advanced Tab

The **Server Advanced** tab is used to manage advanced configuration options for the server.

To access the **Server Advanced** tab, click the **server** in the tree pane and then click the **Server Advanced** tab.

● **Server is behind a Router/Firewall** - Enable this feature if your server is sitting behind a router or firewall. This will allow you to specify the **router IP address** or **host name** (such as a dynamic DNS host name). When this feature is enabled, the router IP address will be used for **PASV mode** connections instead of the local internal server IP address. This feature is extremely helpful if you want to run an FTP server from behind a router or firewall. **NOTE:** As part of this configuration, your router/firewall must be configured to allow port-forwarding of traffic to the Titan Server.

● **External IP Address of Router/Firewall** - Enter the external/public IP address of your corporate router/firewall. This IP address will be returned as part of the **PASV** response to FTP clients when they connect. This allows FTP clients to open data connections to your FTP server. **NOTE:** Your router/firewall must be configured to allow port-forwarding of traffic to the Titan Server. Enable this feature if your server is sitting behind a router or firewall. This will allow you to specify the router IP address or host name (such as a dynamic DNS).

● **Use Internal Server IP in PASV Response for Local Clients** - Enable this feature to have Titan use the Internal LAN IP of the Titan box in response to the **PASV** command from an FTP client on the local LAN. This allows local LAN clients to run in Passive mode while also allowing external clients to use Passive mode.

● **Enable Mode Z Support** - Enable this feature if you would like to provide support for Mode Z Compression. When used in conjunction with a Mode Z enabled client, such as [WebDrive](#), data will be compressed prior to being transferred. This greatly increases performance in most cases.


● **Default Compression Level** - Set the default compression level for compressed Mode Z transfers. This value can later be changed by the FTP client.

● **Enable UTF-8 Encoding Support** - If enabled, Titan FTP Server will support UTF-8 encoding of files and directories. UTF-8 is enabled through the **OPTS** command. If UTF-8 support is enabled, Titan FTP Server will return all directory and filename information encoded in UTF-8 format.

● **Create Standard UNIX Directories** - If enabled, the standard **/bin**, **/incoming**, **/pub**, and **/usr** UNIX directories will be created. User accounts will be created under the **/usr** directory.

• **Directory Case Preservation** - This feature dictates how folder-case is handled. You can choose to have all names converted to **uppercase**, **lowercase**, or you can choose to have **case preserved**. **Note:** since Windows does not distinguish between case, Titan will not be able to store two folders with identical names, even if their case is different, for example, **FOLDER1** and **folder1**.

• **File Case Preservation** - This feature dictates how file-case is handled. You can choose to have all names converted to **uppercase**, **lowercase**, or you can choose to have **case preserved**. **Note:** since Windows does not distinguish between case, Titan will not be able to store two files with identical names, even if their case is different, for example, **FILE.TXT** and **file.txt**.

Two orange pushpins are pinned to the left side of a grey rectangular box.

- If you enable the **Server is Behind a Router** option, you should also enable the **Allow Passive Mode Connections** and **Limit Passive Mode Port Range** options found on the **Server -> Connections Advanced** tab.
- If you would like more information about configuring Titan Server with a **router/firewall**, please see the [Titan Router/Firewall Configuration Quick Start Guide](#).

Server User Authentication Tab

The **User Authentication** tab is used to manage user authentication configuration options for the server.

To access the **User Authentication** tab, click the **server** in the tree pane and then click the **User Authentication** tab.

● **User Authentication Method** - This field will display the currently defined user authentication method that is being used by Titan. The User Authentication method is defined during server creation. See [User Authentication Options](#) for more information about the various authentication methods supported by Titan.

● **Authentication Server Setup** - This button will launch the **User Authentication Configuration Wizard**. The User Authentication Configuration Wizard will guide you through configuration options for the various methods of user authentication that are supported by Titan. **NOTE:** Once you select a User Authentication Database in Titan, you cannot change to a different method once the [New Server wizard](#) has completed.

● **Auto Assign Home Directory For New Users** - This option is enabled when you select **Native Titan User Authentication** as your user authentication method. When selected, Titan will automatically generate users' home directories. Home directories will be under `/usr/<username>` for each user.



If you would like more information about **User Authentication**, please see the [User Authentication](#) topic.

Server UNC Accounts Tab

The **UNC Accounts** tab is used to define a list of domain usernames and passwords that will be used for authentication when Titan needs to access a remote UNC share.

To access the **UNC Accounts** tab, click the **server** in the tree pane and then click the **UNC Accounts** tab.

Because the Titan Service usually runs under the context of the **LocalSystem NT Account** defined for the local computer, it does not normally have rights to access a UNC resource that is located on a remote server. When Titan attempts to access a file/folder stored on a UNC share, it will attempt to connect/authenticate itself against the remote UNC by sending over a UNC username and password along with the UNC.

How Titan uses the UNC Accounts List

Titan will check the users in the list one at a time until it authenticates against the UNC share. This list is not intended to be a list of your FTP users. You will likely only need to add one username to the UNC Accounts tab. **The UNC account should have all of the permissions any of your users will need on the UNC share.** The permissions of the UNC user can be further restricted by Titan, but Titan cannot elevate the permissions of the UNC user. For example, a user may have write access in Titan but if the UNC user does not have write access, the user will not be able to write to files.

NOTE: If you are using NT Impersonation, the UNC Accounts tab will be disabled. UNC accounts are not used in conjunction with NT Impersonation. When you use NT Impersonation, the access rights of individual users will be used to authenticate against UNC shares.

- **Username** - Type a **domain username** that will be used for authentication against the remote UNC share. The username can be simply a **username**, or **username@domain** or **domain\username**.
- **Password** - Type the corresponding password that will be used for authentication against the remote UNC share.
- **Add/Remove** - Use these buttons to add a new Username/Password to the UNC Accounts list, or to remove the currently selected UNC Account from the list.
- **UNC Account List** - Contains the list of domain accounts that will be used for authentication against the remote UNC share. Titan will present each username/password to the UNC server until it receives a successful authorization.



If you would like information about configuring Titan using UNC paths for data storage and scalability, see the [Titan Using UNC Paths for Data Storage & Scalability Quick Start Guide](#).

Server Email Server Tab

The **Email Server** tab is used to configure mail server settings used by Titan.

To access the **Email Server** tab, click the **server** in the tree pane and then click the **Email Server** tab.

- **SMTP Server IP or Hostname** - Type the **IP address** or **host name** of the SMTP server used for sending email.
- **Mail Server Username**- Type a valid **Username** that will be used for authentication to the remote SMTP server.
- **Mail Server Password** - Type a corresponding **Password** for the username.
- **Test Connection** - Click **Test Connection** to test the connection settings to the SMTP server. If Titan is unable to connect to the server, or unable to authenticate to the server using your credentials, an error will be displayed.



If your email SMTP server does not require a username/password for authentication, Titan may not be able to connect to your server. We recommend that you specify a valid email username and password for authentication against the SMTP server.

Server FTP Tab

The **FTP** tab is used to configure the general FTP settings used by Titan.

To access the **FTP** tab, click the **server** in the tree pane and then click the **FTP** tab.

- **Enable FTP access on this server** - This option enabled/disables FTP services on the Titan server. If this option is disabled, FTP (and FTPS) services will not be enabled.

- **FTP port**- Enter the port to be used for accepting FTP connections. The default FTP port is **port 21**.

- **FTP send buffer size** - This value defines the size of the buffer that is used to send data to the remote client during file downloads. If you experience very slow transmission rates between Titan and the FTP client, it could be that there is a high latency rate on your network. Increasing/decreasing this value could help improve performance on high latency networks.

- **FTP receive buffer size** - This value defines the size of the buffer that is used to receive data from the remote FTP client during file uploads. Increasing or decreasing this value could help improve performance on high latency networks.

- **NLST Returns File names and directory/folder names** – When enabled, this option forces **NLST** to return directory/folder names in addition to returning file names.

- **Exclusively lock file during upload** – When enabled, Titan will create an exclusive lock on the server file while it is being uploaded from the client.

- **Allow Anonymous Access** - When selected, enables the **Anonymous** user account. This option is enabled by default.

- **Check Anonymous Password** - Requires that the password for **Anonymous** access be of the **x@y** format (usually specifying an e-mail address). Checking is only performed to ensure that the format is correct. No checking is performed to ensure that the email address and/or domain are valid.

Server Web Access

The **HTTP/HTTPS** tab is used to configure the settings for the Titan Web Interface.

To access the **HTTP/HTTPS** tab, click the **Server** in the tree pane and then click the **HTTP/HTTPS** tab.

Enable HTTP access on this server - Select the check box to enable HTTP protocol on this server.

- **IP Address** - Use the drop-down arrow to select your IP address. **Any Available IP Address** indicates that the server will listen on all IP addresses that are configured on the computer, along with the local IP address of 127.0.0.0, also known as **localhost**.
- **Port** - Type in your port number. The default port is port 80.


Enable HTTPS/SSL access on this server - Select the check box to enable HTTPS protocol on this server.

- **IP Address** - Use the drop-down arrow to select your IP address. **Any Available IP Address** indicates that the server will listen on all IP addresses that are configured on the computer, along with the local IP address of 127.0.0.0, also known as **localhost**.
- **Port** - Type in your port number. The default port is port 443.
- **Use the following certificate for this server** - Use the drop-down arrow to select the certificate to be used for this server.
- **Certificate Management** - Launches the Certificate Manager, which can be used to create, import, and manage certificates.
- **Enter the password associated with this certificate** - Type in the certificate password.

WAN Address - Type the external domain name or IP address that clients will use to connect to this server, for example, **myserver.com**.

Web Services

Web Interface Module - Select the check box to enable the Titan FTP Server Web interface.



- The Titan Web Interface is an optional module, contact sales@southrivertech.com for more information.
- If you would like more information about configuring Titan Server SSL settings, see the [Titan FTP/SSL and Public Key Certificate Quick Start Guide](#).
- For more information about the Titan FTP Server Web Interface, see the [Titan FTP Server Web Interface User's Guide](#).

Server Connections

Server Connections General tab

The **Connections General** tab is used to configure general connection settings. These values can be set at the **Server**, **Group**, and **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access the **Connections General** tab at the **Server** level, expand the **Server** in the tree pane, click **Connections**, and then click the **Connections General** tab.

To access these settings at the **Group** or **User** level, in the tree pane click the **Group** or **User**, click **Connections**, and then click the **Connections General** tab.

- **Use Inherited Setting** - This check box appears on the **Group** and **User** levels. When selected, the settings will be inherited. Clear this check box to override the inherited values.

- **Max Concurrent Connections** - Specifies the maximum number of concurrent sessions that may be established.

- **Idle Connection Time-out** - Specifies the maximum amount of time, in minutes, to wait before dropping a user due to inactivity.

- **Max Connections/IP** - Specifies the maximum number of concurrent connections a user can establish from any given IP address.

- **Max Upload Speed** - Specifies the maximum KPS (**K**ilobytes **P**er **S**econd) upload speed that the server will allow from the user. If the user attempts to exceed this bandwidth allotment, the server will pause the transfer and slow it down to the proper speed.

- **Max Download Speed** - Specifies the maximum KPS (**K**ilobytes **P**er **S**econd) download speed that the server will allow data to be sent to the user.

- **Max Uploads/Session** - Specifies the maximum number of files that can be uploaded per session. Once this limit has been reached, the user will not be able to upload/replace any files until they log out and then log back in.

- **Max Downloads/Session** - Specifies the maximum number of files that can be downloaded per session. Once this limit has been reached, the user will not be able to download any files until they log out and then log back in.

- **Max File Upload Size** - Specifies the maximum file size that can be uploaded by the user. Any attempt to upload a larger file will be aborted and the file will be deleted from the system.

- **Max File Download Size** - Specifies the maximum file size that can be downloaded by this user. Any attempts to download files larger than this value will be denied.

Server Connections Advanced tab

The **Connections Advanced** tab is used to configure advanced connection settings. These values can be set at the **Server**, **Group**, and **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access the **Connections Advanced** tab at the **Server** level, expand the **Server** in the tree pane, click **Connections**, and then click the **Connections Advanced** tab.

To access these settings at the **Group** or **User** level, in the tree pane click the **Group** or **User**, click **Connections**, and then click the **Connections Advanced** tab.

- **Use Inherited Setting** - This check box appears on the **Group** and **User** levels. When selected, the values will be inherited. Clear this check box to override the inherited values.

- **Disable account after X invalid password attempts** - When enabled, the user account will be disabled after the specified number of consecutive incorrect password attempts.

- **Kick User after X consecutive bad commands** - When enabled, the user connection will be dropped after the user types the specified number of invalid commands.
 - **Disable user account afterwards** - When selected, the user's account will be disabled after they are dropped from the system.
 - **Ban access from this IP Address once this happens** - When selected, the user's IP address will be added to the **IP Access Restrictions** list and will no longer have access to the server.

- **Allow PASV Mode Connections** - When enabled, allows the server to be placed in **PASV** mode by the client. If this feature is disabled and the client attempts to issue a **PASV** command, they will receive a **502 Not Implemented** response.

- **Allow EPSV Mode Connections** - Similar to the **PASV** command, but used for IP v6 addressing. When enabled, allows the server to be placed in **EPSV** mode by the client. If this feature is disabled and the client attempts to issue an **EPSV** command, they will receive a **502 Not Implemented** response.
 - **Limit PASV Port Range** - Allows you to specify a specific range of ports that the server will use when the user issues a **PASV** command. This is useful if the FTP Server is behind a firewall/router and you only want to open a specified range of ports for use by the server.

- **Delete Partially Uploaded Files** - When enabled, the server will delete any files that are not successfully uploaded. For example, if a **STOR** or **STOU** does not complete successfully, the file will be deleted from the server.

- **Block Anti-Timeout Schemes** - When enabled, the server will ignore **NOOP** commands as attempts are made to keep the connection alive.

● **Block FTP Bounce Attacks and FXP** - When enabled, the server will not accept any data connections from IP addresses other than that of the user's primary connection.

● **Allow change of password (SITE PSWD)** - When enabled, users are permitted to change their password using the **SITE PSWD** command.

Server Connections IP Access Tab

The **IP Access** tab is used to configure IP Access restrictions. These values can be set at the **Server**, **Group**, and **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access the **IP Access** tab at the **Server** level, in the tree pane expand the Server, click **Connections**, and then click the **IP Access** tab.

To access these settings at the **Group** or **User** level, in the tree pane select the **Group** or **User**, click **Connections**, and then click the **IP Access** tab.

- **Use Inherited Setting** - This check box appears on the **Group** and **User** levels. If checked, the values will be inherited. Clear this check box to override the inherited values.

- **Enable IP Access Restrictions** - When enabled, IP Access restrictions will be applied at this level whenever a user attempts to connect.

- **Grant/Deny access by default** - Select the default action that will be applied at this level when a connection attempt is made.

- **Except the addresses listed below** - (Exception List) Enter a list of IP addresses that will be the exception to the default rule. For example, you can **Deny Access by default** and then type a single IP address. This will be the only IP address that the user will be permitted to connect from.

Server Connections Upload/Download Ratios Tab

The **Upload/Download Ratios tab** is used to configure Upload/Download ratios. These values may be set at the **Server**, **Group**, or **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access the **Upload/Download Ratios** tab at the **Server** level, in the tree pane expand the **Server** and click **Connections**, and then click the **Upload/Download Ratios** tab.

To access these settings at the **Group** or **User** level, in the tree pane select the **Group** or **User**, click **Connections**, and then click the **Upload/Download Ratios** tab.

● **Use Inherited Setting** - This check box appears on the **Group** and **User** levels. When selected, the values will be inherited. Clear this check box to override the inherited values.

● **Enable UL/DL Ratios** - When enabled, the user will have **Upload/Download ratios** applied to their sessions.

- **Count # of Files Per Session** - Ratios will be applied at the file level on a per-session basis. Each time the user disconnects from the server, the counters are reset. If the user has multiple concurrent sessions open, statistics are applied for each individual session.
- **Count KBytes Per Session** - Ratios will be applied at the file size level on a per-session basis. Each time the user disconnects from the server, the counters will be reset. If the user has multiple concurrent sessions open, statistics are applied for each individual session.
- **Count # of Files across All User Sessions** - Ratios will be applied at the file level across all concurrent sessions for the user.
- **Count KBytes across All User Sessions** - Ratios will be applied at the file size level across all concurrent sessions for the user.

● **Ratios** - Enter the UL to DL ratio for the user. For example, you may want to require that the user upload two files for every one file that they download, so the ratio would be **Uploads 2, Downloads 1**. You may also want to require that the user upload 1MB of data for each 1MB of data that they download, so the ratio would be **Upload 1000, Download 1000**, and select the **Count KBytes** option.

● **Free Files List** - Type a list of files/file types that will be excluded from the ratios.

Server Custom Messages Tab

The **Messages tab** is used to configure custom messages at the **Server**, **Group**, and **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access these settings at the **Server** level, in the tree pane expand the **Server**, click **Connections**, and then click the **Messages** tab.

To access these settings at the **Group** or **User** level, in the tree pane select the **Group** or **User**, click **Connections**, and then click the **Messages** tab.

- **Custom Message Type** - Select the message to customize. The current message will be displayed in the **Message Text** field.
- **Use Default Setting** - At the **Server** level, you can select this check box to use the default message for the selected Message Type. Clear this check box to customize the message.
- **Use Inherited Setting** - At the **Group** and **User** level, you can select this check box to use the inherited message for the selected **Message type**. Clear this check box to customize the message.
- **Custom Message** - Type the message to be displayed when this event occurs. Custom messages are limited to **1024 bytes**.

Custom Message Variables

Server Files/Directories

Server Files/Directories Tab

The **Files/Directories** tab is used to configure general file/directory settings. These values can be set at the **Server**, **Group**, and **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access these settings on the **Server** Level, in the tree pane expand the **server**, click **Files/Directories**, and then click the **Files/Directories** tab.

To access these settings at the **Group** or **User** level, in the tree pane select the **Group** or **User**, click **Files/Directories**, and then click the **Files/Directories** tab.

- **Use Inherited Setting** - These configuration options appear on the **Group** and **User** levels. When selected, the values will be inherited. Clear this check box to override the inherited values.

- **Lock User(s) in Home Directory** - Enable this feature to prevent the user from leaving the user's home directory and venturing "up" the tree. If the user's Home Directory is `c:\usr\test1\` and this feature is enabled, then the user's Home Directory will appear as the root "/" when a **PWD** is performed from the client. The user will not be able to **CWD** or **CDUP** from the user's home directory. If this feature is not enabled, then the user's Home Directory will appear as `/usr/test1/` and the user will be able to **CWD** or **CDUP** to other directories in the system, provided that the user has adequate permissions.

- **Show Hidden Files** - When enabled, Titan will display hidden files in the directory listings that are sent to the client.

- **Hide directories users cannot enter** - When enabled, Titan will not display any folder/directory entries that the user does not have adequate rights to.

- **Allow modification of file dates/times via MDTM command** - When enabled, the user will be permitted to modify file dates/times by issuing the **SITE MDTM** command from the client.

- **Allow modification of file dates/times via MFMT command** - When enabled, the user will be permitted to modify file dates/times by issuing the **SITE MFMT** command from the client.

- **Allow modification of file dates/times via MFCT command** - When enabled, the user will be permitted to modify file dates/times by issuing the **SITE MFCT** command from the client.

- **STOU Prefix** - The **STOU** command requires that all unique filenames have a prefix. Use the text box to customize the prefix.

- **STOU Extension**- Use the text box to customize the STOU suffix (file extension).
- **Ban the following file types** - Select this check box to ban certain file types. Use the text box to specify a list of file types that are prohibited from the server. Users will not be permitted to upload or rename a file that matches this filter. You must **separate multiple entries** with a **semicolon**.

Server Files/Directories Directory Access Tab

The **Directory Access** tab provides you with the ability to grant or deny access to folders on the server. These settings can be configured at the **Server**, **Group**, and **User levels**. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access these settings at the **Server** level, in the tree pane expand the **server**, click **Files/Directories**, and then click the **Directory Access** tab.

To access these settings at the **Group** or **User** level, in the tree pane select the **Group** or **User**, click **Files/Directories**, and then click the **Directory Access** tab.

Directory Access is broken down into two categories, **File Permissions** and **Folder Permissions**:

● File Permissions

- **Read/Download Files** - Allows users to download files from the server. (**RETR**)
- **Write/Upload New Files** - Allows users to upload files to the server (**STOR**, **STOU**). **Note:** This right applies to **new** files only; it does not allow the user to replace/append existing files.
- **Append/Replace Files** - Allows users to upload/replace/append existing files on the server. **Note:** This rule applies only to **existing** files; this rule does not apply to new files.
- **Delete Files** - Allows users to delete existing files from the server (**DELE**).
- **Rename Files** - Allows users to rename existing files on the server (**RNFR**).

● Folder Permissions

- **Create Subdirectories** - Allows users to create subdirectories within the current folder/directory (**MKD**).
- **Remove Subdirectories** - Allows users to remove subdirectories from the current folder/directory (**RMD**). **Note:** The sub-directory must be empty to remove it. This permission usually coincides to the **Delete Files** permission. The combination of these two permissions will permit members to delete directory trees.
- **Can View Directory Listing** - Allows users to generate a directory listing of the contents of the folder (**LIST/NLST**).
- **Apply Rights to Subdirectories** - Enable this check box to have these permissions, both **File** and **Folder**, propagated to all subdirectories of the specified path.

Titan FTP Server will apply permissions as follows:

1. Directory Access rules for the **USER** are loaded first.
2. Directory Access rules for all **GROUPS** in which the user is a member of are loaded next. These **GROUPS** are loaded in the order in which they appear in the **Groups tab** of the **User Configuration dialog**. If a duplicate folder is encountered for which there are already **Dir Access Perms** specified, the **SUM** of the permissions is used.
3. Directory Access rules for the **SERVER** are loaded last. Again, if duplicates are located, they are summed together.

Example:

User A is a member of **Group 1** and **Group 2** for **Server S**.

User A has the following permissions: **/pub/** - **Read** permissions

Group 1 has the following permissions: **/pub/** - **Write** permissions

Group 2 has the following permissions: **/pub/** - **NO ACCESS AT ALL**

Server S has the following permissions: **/pub/** - **LIST** permissions

Outcome: **User A** will have **READ**, **WRITE** and **LIST** permissions to the **/pub/** folder.

Server Files/Directories Virtual Folders Tab

Virtual folders are used to link or map external folders into a user's directory space. For Windows users, think of a virtual folder as a Windows shortcut. The link appears in one location whereas the data lives elsewhere. For UNIX users, virtual folders are very similar to symbolic links. Virtual Folders are commonly used to map network shares or folders from different drive letters into the server directory structure.

Virtual folders can be added at the **Server**, **Group**, or **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access the **Virtual Folders** tab at the **Server** level, in the tree pane expand the **Server**, click **Files/Directories**, and then click the **Virtual Folders** tab.

To access the **Virtual Folders** tab at the **Group** or **User** level, in the tree pane select the **Group** or **User**, click **Files/Directories**, and then click the **Virtual Folders** tab.


Using Virtual Folders

1. Select the **Server/Group/User** where you wish to add the Virtual Folder.
2. Select the **Files/Directories** node.
3. Select the **Virtual Folders** tab.
4. Click on the **Add** button to display the **New Virtual Folder** wizard.
5. To select the fully qualified path, browse to the actual/real **physical** folder. You may select a folder on your local computer, or you may choose a network folder that has been previously shared. If you are mapping a UNC share, make sure that the account under which the Titan FTP Service is running has access to the UNC. Click **Next**.
6. After you have selected a **Physical Path**, the **Virtual Path** will be filled in automatically. Click **Next**.
7. Select the default **Access Rights** for this new Virtual Folder using the check boxes. Click **Next**.
8. The Actual Path of the folder is displayed and the Virtual Path is displayed. You can change the **Folder Name** as it will appear under the virtual path, or you can leave the default name (which is the same as the Actual Path name). Click **Finish** to generate the virtual folder mapping.
9. The Virtual Path and the Actual Path are now displayed in the **Virtual Folders** tab. Click **Apply**.

Note: Virtual folder updates are not real-time. If a user is currently connected to the server, and you make changes to the Virtual Folder list, users will need to log out and then log back in to the system to see the virtual folder changes.

Note: If mapping a UNC share, make sure that the account under which the Titan Service is running has access to the UNC. Otherwise, you will need to add the appropriate username and password under the [UNC Accounts](#) tab.

[More information about using virtual folders](#)



- If you attempt to create a virtual folder to a mapped network drive, Titan will replace the drive mapping with the actual UNC name. This is done because the Titan Service does not have access to mapped drives, only to UNC shares.
- If you would like more information about configuring group level virtual folders in Titan, see the Using Group Level Folders [Quick Start Guide](#).

Server Files/Directories Disk Quotas Tab

The **Disk Quotas tab** is used to configure disk quota limits. These values may be set at the **Server**, **Group**, or **User level**. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access these settings at the **Server** level, in the tree pane expand the **Server**, click **Files/Directories**, and then click the **Disk Quotas** tab.

To access these settings at the **Group** or **User** level, in the tree pane expand the **Server**, expand the **Group** or **User**, click **Files/Directories**, and then click the **Disk Quotas** tab.

- **Use Inherited Setting** - This check box appears on the **Group** and **User** levels. When selected, the values will be inherited. Clear this check box to override the inherited values.

- **Enable Disk Quotas** - When enabled, the server will have a limited amount of storage space.

- **Current Disk Usage** - Shows the amount of storage currently being used by this server.

- **Maximum Disk Usage** - Specifies the maximum number of Kilobytes of data that the server will be permitted to store.

- **Recalculate** - Recalculates the current disk usage.

- **Explore** - Launches Explorer so that you can browse the server's **data directory**. This is useful if you need to purge invalid files or if you want to see where space is being used.

- **Free Files List** - Type a list of files/file types that will not be included in the quota calculations.

Server Security

Server FTPS/SSL Tab

The **FTPS/SSL tab** is used to configure secure FTP (FTPS, or FTP over SSL) settings used by Titan. For more information about the SSL features supported by Titan, See [SSL Support](#).


To access the **FTPS/SSL tab**, in the tree pane, expand the **server**, click **Security**, and then in the tab pane, click the **FTPS/SSL tab**.

- **Enable SSL/TLS access on this server** - This option enables/disables basic FTPS services.
- **Protocol Version** - Select the highest/maximum supported SSL version used by Titan. The minimum supported protocol is SSL v3.0. Titan does not support SSL v2 and will reject any client connection presenting SSL v2 messages.
- **Enable explicit SSL/TLS access (AUTH SSL)** - When this option is enabled, Titan will allow incoming connections on the standard FTP port and once connected, the remote FTPS client may issue the AUTH SSL or AUTH TLS command to initiate the secure handshake process with the Titan server.
- **Enable implicit SSL/TLS** - When this option is enabled, Titan will accept secure connections on the Implicit SSL/TLS port. Any inbound connection on this port will "imply" that it is secure and Titan will immediately initiate a secure handshake with the remote client before any FTP commands are accepted from the client.
- **Implicit SSL/TLS port** - Select the port number that Titan will use for inbound SSL connections. The default Implicit SSL port is **port 990**.
- **Encrypt data channel by default** - Protect the data channel by encrypting all communications by default. If a client makes a PROT command, that value will override this setting.
- **Enable CCC (Clear Command Channel)** - This option enables Titan support for the CCC command. The CCC command can be issued by a remote FTPS client and will cause Titan to fall out of secure mode and back into unsecure mode. This option is useful for clients who only need to secure the **authentication** portion of the session. Once the **USER/PASS** has completed, some clients will use CCC to return to unsecure mode, which is faster.
- **Enable secure site-to-site (FXP) file transfers (CPSV/SSCN)** - Enables **FXP site-to-site** transfers. FXP indicates a direct server-to-server file transfer.
- **Require all FTP connections to be secure** - Prevents normal FTP connections from being established. When enabled, the user must use **Explicit** or **Implicit** SSL or the connection will be terminated.

Server Certificate Settings

- **Require trusted certificates from clients who connect securely** - this feature requires that all FTPS clients provide a trusted certificate to connect. This is the most secure method of connecting but it requires that trusted keys be distributed offline to each user.
- **Use the following certificate for this server** - Select a certificate to use for this server.

- **Certificate Management** - Click **Certificate Management** to create or import certificates.
- **Certificate Password** - Type the **Password** for the selected certificate.
- **Certificate Store Folder** - This is the location where Titan will store all certificates for this server. **Note:** Local paths and UNC shares are supported; do not use a mapped drive because mapped network drives are not accessible from the Titan service.



- Titan does not support SSL v2 and will reject any client that attempts a connection using SSL v2.
- If you would like more information about configuring Titan Server FTPS/SSL settings, see the Titan FTPS/SSL & Certificate Management [Quick Start Guide](#).
- FTPS/SSL features are only available in the Enterprise edition of Titan Server.

Server SFTP/SSH Tab

The **SFTP/SSH tab** is used to configure SFTP/SSH settings for this server. See [SFTP Support](#) for more information.

To access the **SFTP/SSH tab**, in the tree pane, expand the **server**, click **Security**, and then in the tab pane, click the **SFTP/SSH tab**.

- **Enable SFTP on this server** - Enables/disables SFTP connections for the server. SFTP, SSH's Secure File Transfer Protocol, is a special subsystem of SSH and is different from FTP and FTPS.
- **SFTP Port** - Port used for SFTP connections. Default SFTP port is **port 22**.
- **SFTP Version** - Select the highest version of SFTP that the server will accept. Clients will negotiate with the server and the highest version that both the client and the server support will be used. Titan currently supports versions 3, 4, 5, and 6 of the SFTP protocol.
- **Use zlib compression** - Enable zlib compression to increase transfer throughput.
- **Exclusively lock file during upload** - When enabled, Titan will create an exclusive lock on the server file while it is being uploaded from the client.


Cipher/MAC Settings

- **Cipher preferences** - Select one or more encryption ciphers that will be supported by the server. During the SSH handshake, Titan presents a list of supported ciphers to the SFTP client. The client will do the same and the two parties will negotiate on a commonly supported cipher. **Note:** the cipher used to send data from the client to the server may be different from the cipher used to send data from the server to the client.
- **MAC preferences** - Select one or more MAC (Message Authentication Code) algorithms that will be supported by the server. The list of supported MACs is presented to the SFTP client during the handshake.

Server host key settings

- **Require trusted host keys when accessing this server** - Enable this feature to require that a user presents a valid host key when connecting to the system. When this feature is enabled, Titan will not allow password or keyboard-interactive authentication methods, only Public-Key authentication.

- **Use the following host key for this server** - Select an existing host key to be used by the server. If no host keys are available, use the **Host Key Management** utility to create a new server host key pair.
- **Host Key Management** - Launches the Titan **Host Key Management** utility that allows you to create, import, export, and manage SSH host keys used by Titan.
- **Host key password** - Enter the password used to secure the **private key** portion of the selected host key.
- **Host Key Folder** - Enter the **fully qualified path** where Titan SSH host keys will be stored. **Note:** Local paths and UNC shares are supported; do not use a mapped drive because mapped network drives are not accessible from the Titan service.

A small icon of a red flag on a silver pole, indicating a warning or important note.

- Titan will not accept host key pairs that are not secured by a password. Passwords used to secure the private key portion of a host key pair must be at least 4 characters in length.
- Titan does not accept connections from SSH v1.x clients.
- If you would like more information about configuring Titan Server SFTP settings & Host Key Management, please see the Titan SFTP & Host Key Management [Quick Start Guide](#).

Server Flood Protection/DoS Tab

The **Flood Protection/DoS Settings** tab is used to manage flood protection configuration options for the server.

To access the **Flood Protection/DoS** tab, in the tree pane, expand the **server**, and click **Security**, and then in the tab pane, click the **Flood Protection/DoS** tab.

Flood protection is designed to limit a hacker's ability to flood your server with multiple connections over a short period. Many times this is used to produce a DoS (Denial of Service) attack, which is designed to cripple the server so that it is unable to service existing or new connections properly.

Titan has the ability to track incoming connections based on IP address and "time since last connection". If Titan finds that a client IP address has attempted to connect to the Titan server more than X number of times in Y seconds, Titan flags this client IP address as **Flooding the server** and will close the incoming connection and will also prevent any future connections from that IP address.

Titan can ban an IP address forever, which adds the IP address to the **IP Access denial list** (see [Server > Connections > IP Access](#)), or it can ban the IP for a given period. Use the **Ban IP Address Forever** feature carefully, because by using this feature you could inadvertently ban a valid client. Sometimes a client will use multiple connections to transfer many very tiny files to the Titan server. If the files are transferred very quickly, this will cause the client to open and close many connections. Titan may erroneously flag this as a flooding attempt and ban the client IP from connecting.

- **Enable Flood Protection (DoS/Hammering)** - Enable this feature to have Titan track incoming connections and look for flooding/DoS attacks. This feature is enabled by default.

- **X Connections received from an IP address within Y Seconds** - Set the thresholds for the minimum number of connections and minimum seconds that must elapse before the client IP is flagged as flooding. It is recommended that the **Number of Connections** is set high and the **Number Of Seconds** is set low to prevent incorrect flagging of valid clients. The default setting is 200 connections received from an IP address within 5 seconds.

- **Ban IP Address Forever** - If this option is selected, Titan will add the client IP address to the list of IP addresses that are banned from accessing the server. For more information on the banned IP list, see [Server > Connections > IP Access](#).

- **Ban IP Address for X Minutes** - If this option is selected, connections from the client IP address will be prevented for the predefined number of minutes. Once that period has expired, the IP address is removed from the banned list and the client will be allowed to connect. The default setting is 60 minutes.

Server Logging

Server Log Tab

The **Server Log** tab is used to view the Server Log in real-time.

To access the **Server Log** tab, in the tree pane expand the **Server**, click **Logging**, and then click the **Server Log** tab.

- **Auto-refresh** - Enable this option to have the logfile viewer automatically refresh the screen after the specified interval has elapsed. For efficiency, the most recent entries are listed (200). For a complete listing of the entire logfile, click **View Entire Logfile**.
- **Refresh** - Click **Refresh** to force an immediate refresh of the log viewer.
- **View Entire Logfile** - Opens the Logfile in a text editor.
- **Clear Log Window** - Clears the Log Window. To view the erased text, click **View Entire Logfile**.

Server Log Settings Tab

The **Log Settings** tab is used to configure the logging options for the server.

To access the **Log Settings** tab, in the tree pane expand the **Server**, click **Logging**, and then click the **Log Settings** tab.

- **Enable Logging to File** - Select this check box to enable logging to a disk file. This is highly recommended.

- **Enable Logging to Screen** - Select this check box to enable logging to the **Activity** screen in the Titan **Administrator**.

- **Log Directory** - Specifies the location where log files will be stored.

- **Explore Log Directory** - Launches Windows Explorer so that you can browse the contents of the Log Directory for this server.

- **Log File Format**- Use the drop-down arrow to select the output format for the server log file.
 - **Plain Text Format** - Log entries will be formatted in plain text format (default).
 - **W3C Extended Log File Format** - Log entries formatted according to W3C standard. **NOTE:** The W3C Standard is to record times in GMT.


- **Log Fields** - Select the log fields to be included in each log entry (**Date**, **Time**, **ServerID/Socket#**, **Message**).

- **Information Level** - Choose the level of information to be recorded in the log file.
 - **General Information** - Minimal information will be recorded in the log file. Errors and Warnings will appear and some high-level server/connection information.
 - **Verbose/Detailed Information - Recommended setting.** Most server and connection information will be recorded to the log file.
 - **Debug Level Information** - The most verbose/detailed level of recording. All detailed and debugging information is recorded. It is recommended that you **do not** use this setting unless instructed to by a technical support representative because it can inhibit performance of your server.

- **Word Wrap** - When enabled, each line in the log file is limited to the specified number of characters.

- **Rotate Log** - Select the rotation schedule for log files. The rotation schedule dictates how often a new log file is created. Selecting "never" is highly discouraged since the log files can become rather large.

• **Rotate Log Now** - Click **Rotate Log Now** to rotate the log immediately. A new Log file will be created based on the current date. If a Log file already exists for the current date, a number will be appended to the Log file until a unique name is found.



- If Anti-Virus software is installed on the same computer as the Titan service, it is highly recommended that the anti-virus software be configured so that it **does not** actively scan the Titan log file subdirectories. Contention between the anti-virus software actively scanning the Titan log files and the Titan service attempting to write to those files could cause performance issues with the Titan server.
- W3C Extended Log File Format: The W3C specifies that the default is to list the times in GMT so that W3C report utilities can convert it to any local time needed. When you use plain text format, time displays correctly for the appropriate time zone.

Server Statistics Tracking Tab

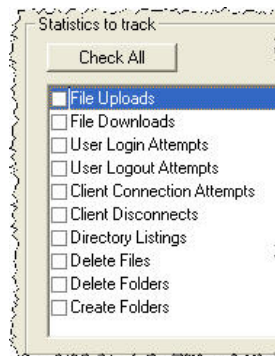
The **Statistics Tracking** tab is used to configure statistics tracking for the server. Statistics are tracked using an ODBC-compliant database, such as SQL Server.

NOTE: SRT supports Titan configurations using SQL Server 2005 or later or SQL Server 2005 Express or later, test or production environment. No other databases are supported.

To access the **Statistics Tracking** tab, in the tree pane expand the **Server**, click **Logging**, and then click the **Statistics Tracking** tab.

Please note that in many cases creating an [Event Handler](#) is a better approach to tracking and responding to server events.

- **Enable Statistics Tracking** - Select the check box to enable statistics tracking for the server.
- **ODBC Datasource** - Use the browse "..." button to select your **ODBC datasource**. Click **Test** to make sure the datasource is valid. You **must** use a **SYSTEM** datasource, not a user datasource. **NOTE:** SRT supports Titan configurations using SQL Server 2005 or later or SQL Server 2005 Express or later, test or production environment. No other databases are supported.
- **Prune/Purge old statistics every** - Allows you to delete old statistics so that the database does not become bloated.
- **Archive old statistics before pruning** - When this option is enabled, the contents of the statistics table will be backed up to an archive table before the statistics table is pruned.
- **Statistics to track** - Use the check boxes to select statistics that you would like to track.



The database table will be named **sr_stats** and will contain the following columns:

Column Name	Data Type	Purpose
serverid	number	A unique identifier associated with each server on the domain.
datestamp	date/time	Contains the date and time of the statistic.
userid	number	A unique identifier associated with each user on the system.
sessionid	text	A unique cookie identifying the client session. This is usually the ID of the Windows Socket.
ipaddress	text	The IP address of the client.
action	text	The action being performed. Usually this will be the FTP/SFTP command that has been issued such as PWD , CWD , LIST , etc.
targetname	memo	The name of the file being accessed. This field may be empty if the current command/action is not targeting a file.
filesize	number	The total size of the file for the action. This size may be zero or missing if the statistic is not associated with a file.
response	number	Contains the standard three-digit response code sent from the server to the client. For SFTP actions, this response will be a single digit.
responsestring	text	Contains the text response associated with the response number sent from the server to the client for the current action.



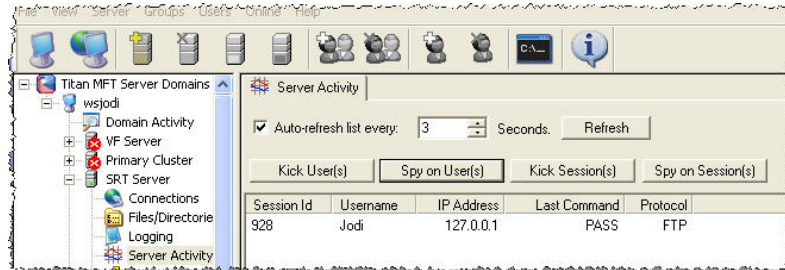
If you would like more information about creating an Event Handler in Titan, please see the Titan Using Events to Thwart Hackers [Quick Start Guide](#).

Server Activity

Server Activity Tab

The **Server Activity** tab shows information about current connections to the server.

To access the **Server Activity** tab, in the tree pane expand the **Server**, click **Server Activity**, and then click the **Server Activity** tab.



- **Auto-refresh list every** - Use the check box to enable this option. When this option is enabled, the list will be refreshed automatically after each interval.
- **Refresh** - Click **Refresh** to refresh the list immediately.
- **Kick User(s)** - This option will kick the selected user from the server. All sessions for this user will be disconnected, but they will not be banned. They will be permitted to log back in to the server. To **Kick and Ban** a user, use the [Events Management](#) features of Titan
- **Spy on User(s)** - Opens up a new [Spy User](#) tab to spy on the selected user(s).
- **Kick Session(s)** - Click **Kick Sessions(s)** to kick the currently selected user session from the system. If the user is connected multiple times and has multiple sessions open, the remaining sessions will remain active.
- **Spy on Session(s)** - Click **Spy on Sessions(s)** to spy on the currently selected user session. Spying on a session allows you to view the logging information specific to the user session. This option opens a [Spy Session](#) tab to spy on the selected session(s).

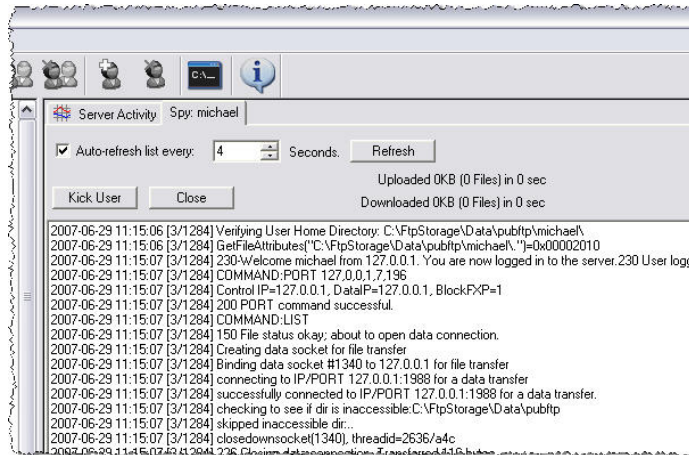


If you would like more information about configuring events in Titan, please see the [Using Events to Thwart Hackers Quick Start Guide](#).

Server Spy User Tab

The **Spy User** tab is useful for tracking the activities of a single user. Keep in mind that multiple sessions can exist for the same user name. This viewer allows you to watch all activity from all sessions for the currently selected user.

To access the **Spy User** tab, from the tree pane expand the **Server** and click **Server Activity**. Select the user that you would like to track and click **Spy on User(s)**. The **Spy:User** tab will activate.



- **Auto-refresh** - When this option is enabled, the list will be refreshed automatically after each interval. If this option is not enabled, you can refresh the viewer manually by clicking **Refresh**.
- **Refresh** - Click **Refresh** to refresh the viewer manually.
- **Kick User** - Click **Kick User** to kick the selected user from the server. All sessions for this user will be disconnected, but they will not be banned. They will be permitted to log back in to the server. To **Kick and Ban** a user, use the [Events Management](#) features of Titan.
- **Close** - Use this button to close the Spy viewer for this user.



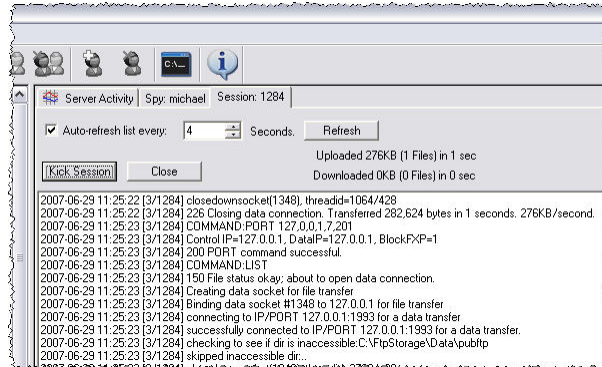
If you would like more information about configuring events in Titan, see the [Using Events to Thwart Hackers Quick Start Guide](#).

Server Spy Session Tab

The **Spy Session** tab is useful for tracking the activities of a single user session.

To access the **Spy Session** tab, from the tree pane expand the **Server**, click **Server Activity**, and then click **Spy Session**.

Keep in mind that multiple sessions can exist for the same user name. This viewer allows you to monitor the current session activity only. To view activity from all sessions, use the **Spy User** feature.



- **Auto-refresh** - When this option is enabled, the list will be refreshed automatically after each interval. If this option is not enabled, you can manually refresh the viewer by clicking **Refresh**.
- **Refresh** - Click **Refresh** to refresh the viewer manually at any time.
- **Kick Session** - Click **Kick Session** to kick the selected user session from the server.
- **Close** - Click **Close** to close the Spy viewer for this session.

Server Events

Server Event Handlers Tab

The **Event Handlers** tab is used to **view**, **create**, **modify**, and **delete** Event Handlers. Event Handlers trigger customized actions based on **events** and **conditions**. Read [Introduction to Event Handling](#) for an overview.

To access the **Event Handlers** tab, in the tree pane expand the **Server**, click **Events**, and then click the **Event Handlers** tab.

- **Add** - Click **Add** to add a new Event Handler for this server.
- **Edit** - click **Edit** to edit an existing Event Handler.
- **Enable/Disable** - Select the **Event** and click **Enable** or **Disable** to enable or disable one or more Event Handlers. Enabled Event Handlers will appear in black text, while disabled Event Handlers will appear in gray text.
- **Remove** - Select the **Event** and click **Remove** to remove one or more Event Handlers.

To view **Events**, **Conditions**, or **Actions** for an Event, select and expand the Event.

- **Events** - [Details on Events](#)
- **Conditions** - [Details on Conditions](#)
- **Actions** - [Details on Actions](#)

Performance Tips

The Event Handler system is designed for efficient performance, but it possible to design an Event Handler that will slow down the server. Therefore, care must be taken to ensure that system performance is not adversely affected.

- **Logging** -When you create a custom log, ensure that the file is periodically rotated or renamed to prevent large file sizes. Appending to large files (1MB+) will cause a noticeable delay, especially if the log is being updated frequently.
- **Flag for admin review** - The **Flag for admin review** action is not designed to be triggered frequently. Overuse of this action will cause the Flagged Events list to become bloated. As this list grows, especially beyond a few hundred entries, performance will deteriorate.
- **Send email** -Depending on your e-mail system, sending e-mail frequently could slow performance. In addition, many e-mail servers have spam countermeasures that may block emails or even blacklist the sender.



- For more information about configuring events in Titan, see the **Using Events to Thwart Hackers Quick Start Guide**.
- Event Management features are only available in the Enterprise edition of Titan Server.


Server Flagged Events Tab

The **Flagged Events** tab is used to display a list of events that have been fired on the server.

To access the **Flagged Events** tab, in the tree pane expand the **Server**, click **Events**, and then click the **Flagged Events** tab.

Flagged events are special events that a Titan Server Administrator can set up to activate under certain conditions. When the event is triggered, it is stored in this tab and saved for administrative review.

• **Remove From List** - Once the event has been reviewed by the Administrator, the event can be removed from the list. To remove the event, select the **event** and click **Remove From List**.

	<ul style="list-style-type: none">• For more information about configuring events in Titan, see the Using Events to Thwart Hackers Quick Start Guide.• Event Management features are only available in the Enterprise edition of Titan Server.
---	--

Configuring Groups

Groups Overview

Groups provide the ability to associate multiple users with similar characteristics. By using **Groups**, you can set access permissions that apply to all members of the group.

To access **Groups**, expand the **Server** in the tree pane, and click **Groups**.

By default, Titan FTP Server will generate a global system group called **Everyone**. The **Everyone** Group will always exist and cannot be deleted. All users are members of the **Everyone** group, so care should be taken when adjusting the access permissions for this group.

As discussed in the [User Authentication](#) section, your Titan FTP Server can be configured to use native Titan User Authentication or Windows NT User Authentication. **Note:** Titan Groups are different from Windows NT User Groups.

You can set up an unlimited number of Groups for each server. Each **Group** can have zero or more **Users**. You can also add the same user to multiple groups. **Note:** If you add a user to multiple groups, that user inherits the sum or culmination of the **Directory Access Permissions** for all of the groups.

NOTE: If you would like to **disable** a group after you have created it, you must first change the Group Home Directory to **No group directory**. See the [Group General tab](#) topic for more information.



NT/SAM authentication is available only in the Enterprise Edition of Titan FTP Server.

Creating New Groups

To create a new **Group** using the Titan Server Administrator:

- Expand the **Server** and right-click on the **Groups** tree pane. Select **New Group**. The **New Group Wizard** will launch.

or

- From the menu bar, select **Groups, New Group Wizard**.

or

- Click **Groups** in the tree pane of the Titan **Administrator**. A list of groups is then generated and displayed in the tab pane. Click **New Group**. The **New Group Wizard** will launch.

Deleting Groups

User groups can be deleted from the Titan Administrator.

NOTE: If you would like to disable a group after you have created it, you must first change the Group Home Directory to No group directory.

To disable delete an existing group using the Titan Administrator:

- In the tree pane, right-click the **Group** that you would like to delete and select **Delete Group** from the context menu. You will be prompted to confirm the deletion of the group.

or

- In the tree pane, select the **Group** that you would like to delete. From the Titan Administrator menu bar, select **Groups, Delete Group**. You will be prompted to confirm the deletion of the group.

or

- In the tree pane, click **Groups**. A list of groups is generated and displayed in the tab pane. Select the **Group** that you would like to delete and click **Delete**. You will be prompted to confirm the deletion of the group.

Note: When you delete a group from the system, all users who are members of that group are removed from the group and their **Directory Access Permissions** are updated.

Adding Users to Groups

Users can be added to any number of **Groups**. To add a user to a group, select the **group** in the tree pane of the Titan Administrator and then click the **Users** tab.

Make the appropriate changes in group membership and then click **Apply** to save the changes.

Note: If a user is a member of multiple groups, the user will inherit the sum or culmination of the **Directory Access Permissions** for the various groups.

For example:

User A is a member of **Group 1** and **Group 2**.

Group 1 has **Read** permissions to folder **/F/**.

Group 2 has **Write** permissions to folder **/F/**.

This means that **User A** will have **Read AND Write** permissions to folder **/F/**.

Removing Users from Groups

Users can be removed from all groups except for the **Everyone** group. Removing a user from a specific group also deletes any **Directory Access Permissions** that user may have had due to membership in that group.

To delete a user from a group, in the tree pane select the **group**, and then click on the **Users** Tab. Make the appropriate changes to the membership of the group and then click **Apply** to save the changes.

Note: These changes will be applied to the user during their **next login session**. If the user is currently logged in to the server, the user will need to log out for their changes to be applied.

Group Properties

Groups tab

The **Groups** tab is used to view currently defined groups. You can also launch the New Group wizard from this tab.

To access the **Groups** tab, in the tree pane, click **Groups**.

New Group Wizard

1. Click **New Group**.
2. Type the **name** of the Group. Use the radio button to set **directory options**.
3. Use the arrow buttons to **add members** to the group, and then click **Finish**.

Group General Tab

The **Group General** tab is used to configure general group settings.

To access the **Group General** tab, in the tree pane click the **group**, and then click the **Group General** tab.

● **Group Enabled** - Specifies whether the group is enabled. This value can be set to quickly enable or disable member users who inherit the **Account Enabled** value from the **Group** level. **NOTE:** If you **disable** a group, you must first set the **Group Home Directory** to **No group directory**; otherwise, the users in this group will continue to use the **Group Home Directory** setting below.

● **Group Name** - Displays the **Group Name**. Use this text box to change the group name.

● **Group Home Directory** - This setting specifies what kind of group home directory is to be used. The Group home directory can be used to ensure that group members are organized according to Group settings, rather than having to set these values at the User level. Care must be taken to ensure that users that are members of multiple groups have the correct home directory. **NOTE:** If you **disable** a group, you must first set the **Group Home Directory** to **No group directory**; otherwise, the users in this group will continue to use the **Group Home Directory** setting below:

- **No group directory** - This group does not have a home directory. Home directories will default to the **Server home directory**.
- **User home directories default to group directory** - Select this value to cause any users that are members of this group to use the **Group Home Directory** as their home directory.
- **User home directories default to subdirectory of group directory** - Select this value to cause any users that are members of this group to have their own subdirectory created under the **Group Home Directory**.

● **Group Directory** - If the group is defined to have user home directories be based on a group directory, this is where the base group directory will be set.

● **Expiration Date** - This setting allows expiration date values to be set at the Group level, thus preventing them from having to be set at the User level.

● **Always Allow Login** - This setting allows you to give members of a Group the ability to be always able to connect to the Server, even if the maximum number of logins has been reached.

Group Users Tab

The **Users** tab for each group is used to manage the membership of users to the group.

To access the **Users** tab, in the tree pane select the **group**, and then select the **Users** tab.

- **Members** - Lists the members of the currently selected group. By being a member of a group, a user adopts all of the permissions, rights, and restrictions assigned to the group.
- **Non-Members** - Lists users who are not currently members of the selected group.

Group FTP Tab

The **FTP** tab is used to configure the group level FTP settings used by Titan. These values can be set at the **Server**, **Group**, and **User level**. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access the **Group** level **FTP settings tab**, in the tree pane select the **group**, and then click the **FTP** tab.

- **Use Inherited Setting** - This check box appears on the Group and User levels. If checked, the settings will be inherited. Clear this check box to override the inherited values.
- **Enable FTP access on this server** - Select this check box to allow members of the group to connect to the server using FTP.

Group Connections

Group Connections General tab

The **Connections General** tab is used to configure general connection settings. These values can be set at the **Server**, **Group**, and **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access the **Connections General** tab at the **Server** level, expand the **Server** in the tree pane, click **Connections**, and then click the **Connections General** tab.

To access these settings at the **Group** or **User** level, in the tree pane click the **Group** or **User**, click **Connections**, and then click the **Connections General** tab.

- **Use Inherited Setting** - This check box appears on the **Group** and **User** levels. When selected, the settings will be inherited. Clear this check box to override the inherited values.

- **Max Concurrent Connections** - Specifies the maximum number of concurrent sessions that may be established.

- **Idle Connection Time-out** - Specifies the maximum amount of time, in minutes, to wait before dropping a user due to inactivity.

- **Max Connections/IP** - Specifies the maximum number of concurrent connections a user can establish from any given IP address.

- **Max Upload Speed** - Specifies the maximum KPS (Kilobytes Per Second) upload speed that the server will allow from the user. If the user attempts to exceed this bandwidth allotment, the server will pause their transfer and slow it down to the proper speed.

- **Max Download Speed** - Specifies the maximum KPS (Kilobytes Per Second) download speed that the server will allow data to be sent to the user.

- **Max Uploads/Session** - Specifies the maximum number of files that can be uploaded per session. Once this limit has been reached, the user will not be able to upload/replace any files until they log out and then log back in.

- **Max Downloads/Session** - Specifies the maximum number of files that can be downloaded per session. Once this limit has been reached, the user will not be able to download any files until they log out and then log back in.

- **Max File Upload Size** - Specifies the maximum file size that can be uploaded by the user. Any attempt to upload a larger file will be aborted and the file will be deleted from the system.

- **Max File Download Size** - Specifies the maximum file size that can be downloaded by this user. Any attempts to download files larger than this value will be denied.

Group Connections Advanced tab

The **Connections Advanced** tab is used to configure advanced connection settings. These values can be set at the **Server**, **Group**, and **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access the **Connections Advanced** tab at the **Server** level, expand the **Server** in the tree pane, click **Connections**, and then click the **Connections Advanced** tab.

To access these settings at the **Group** or **User** level, in the tree pane click the **Group** or **User**, click **Connections**, and then click the **Connections Advanced** tab.

- **Use Inherited Setting** - This check box appears on the **Group** and **User** levels. When selected, the values will be inherited. Clear this check box to override the inherited values.

- **Disable account after X invalid password attempts** - When enabled, the user account will be disabled after the specified number of consecutive incorrect password attempts.

- **Kick User after X consecutive bad commands** - When enabled, the user connection will be dropped after the user types the specified number of invalid commands.
 - **Disable user account afterwards** - When selected, the user's account will be disabled after they are dropped from the system.
 - **Ban access from this IP Address once this happens** - When selected, the user's IP address will be added to the **IP Access Restrictions** list and will no longer have access to the server.

- **Allow PASV Mode Connections** - When enabled, allows the server to be placed in **PASV** mode by the client. If this feature is disabled and the client attempts to issue a **PASV** command, they will receive a **502 Not Implemented** response.

- **Allow EPSV Mode Connections** - Similar to the **PASV** command, but used for IP v6 addressing. When enabled, allows the server to be placed in EPSV mode by the client. If this feature is disabled and the client attempts to issue an EPSV command, they will receive a **502 Not Implemented** response.
 - **Limit PASV Port Range** - Allows you to specify a specific range of ports that the server will use when the user issues a **PASV** command. This is useful if the FTP Server is behind a firewall/router and you only want to open a specified range of ports for use by the server.

- **Delete Partially Uploaded Files** - When enabled, the server will delete any files that are not successfully uploaded. For example, if a **STOR** or **STOU** does not complete successfully, the file will be deleted from the server.

- **Block Anti-Timeout Schemes** - When enabled, the server will ignore **NOOP** commands as attempts are made to keep the connection alive.

- **Block FTP Bounce Attacks and FXP** - When enabled, the server will not accept any data connections from IP addresses other than that of the user's primary connection.

- **Allow change of password (SITE PSWD)** - When enabled, users are permitted to change their password using the **SITE PSWD** command.

Group Connections IP Access Tab

The **IP Access** tab is used to configure IP Access restrictions. These values can be set at the **Server**, **Group**, and **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access the **IP Access** tab at the **Server** level, in the tree pane expand the Server, click **Connections**, and then click the **IP Access** tab.

To access these settings at the **Group** or **User** level, in the tree pane select the **Group** or **User**, click **Connections**, and then click the **IP Access** tab.

- **Use Inherited Setting** - This check box appears on the **Group** and **User** levels. If checked, the values will be inherited. Clear this check box to override the inherited values.
- **Enable IP Access Restrictions** - When enabled, IP Access restrictions will be applied at this level whenever a user attempts to connect.
- **Grant/Deny access by default** - Select the default action that will be applied at this level when a connection attempt is made.
- **Except the addresses listed below** - (Exception List) Enter a list of IP addresses that will be the exception to the default rule. For example, you can **Deny Access by default** and then type a single IP address. This will be the only IP address that the user will be permitted to connect from.

Group Connections Upload/Download Ratios Tab

The **Upload/Download Ratios tab** is used to configure Upload/Download ratios. These values may be set at the **Server**, **Group**, or **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access the **Upload/Download Ratios** tab at the **Server** level, in the tree pane expand the **Server** and click **Connections**, and then click the **Upload/Download Ratios** tab.

To access these settings at the **Group** or **User** level, in the tree pane select the **Group** or **User**, click **Connections**, and then click the **Upload/Download Ratios** tab.

● **Use Inherited Setting** - This check box appears on the **Group** and **User** levels. When selected, the values will be inherited. Clear this check box to override the inherited values.

● **Enable UL/DL Ratios** - When enabled, the user will have **Upload/Download ratios** applied to their sessions.

- **Count # of Files Per Session** - Ratios will be applied at the file level on a per-session basis. Each time the user disconnects from the server, the counters are reset. If the user has multiple concurrent sessions open, statistics are applied for each individual session.
- **Count KBytes Per Session** - Ratios will be applied at the file size level on a per-session basis. Each time the user disconnects from the server, the counters will be reset. If the user has multiple concurrent sessions open, statistics are applied for each individual session.
- **Count # of Files across All User Sessions** - Ratios will be applied at the file level across all concurrent sessions for the user.
- **Count KBytes across All User Sessions** - Ratios will be applied at the file size level across all concurrent sessions for the user.

● **Ratios** - Enter the UL to DL ratio for the user. For example, you may want to require that the user upload two files for every one file that they download, so the ratio would be **Uploads 2, Downloads 1**. You may also want to require that the user upload 1MB of data for each 1MB of data that they download, so the ratio would be **Upload 1000, Download 1000**, and select the **Count KBytes** option.

● **Free Files List** - Type a list of files/file types that will be excluded from the ratios.

Group Custom Messages Tab

The **Messages tab** is used to configure custom messages at the **Server**, **Group**, and **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access these settings at the **Server** level, in the tree pane expand the **Server**, click **Connections**, and then click the **Messages** tab.

To access these settings at the **Group** or **User** level, in the tree pane select the **Group** or **User**, click **Connections**, and then click the **Messages** tab.

- **Custom Message Type** - Select the message to customize. The current message will be displayed in the **Message Text** field.
- **Use Default Setting** - At the **Server** level, you can select this check box to use the default message for the selected Message Type. Clear this check box to customize the message.
- **Use Inherited Setting** - At the **Group** and **User** level, you can select this check box to use the inherited message for the selected **Message type**. Clear this check box to customize the message.
- **Custom Message** - Type the message to be displayed when this event occurs. Custom messages are limited to **1024 bytes**.

Custom Message Variables

Group Files/Directories

Group Files/Directories Tab

The **Files/Directories** tab is used to configure general file/directory settings. These values can be set at the **Server**, **Group**, and **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access these settings on the **Server** Level, in the tree pane expand the **server**, click **Files/Directories**, and then click the **Files/Directories** tab.

To access these settings at the **Group** or **User** level, in the tree pane select the **Group** or **User**, click **Files/Directories**, and then click the **Files/Directories** tab.

- **Use Inherited Setting** - These configuration options appear on the **Group** and **User** levels. When selected, the values will be inherited. Clear this check box to override the inherited values.

- **Lock User(s) in Home Directory** - Enable this feature to prevent the user from leaving the user's home directory and venturing "up" the tree. If the user's Home Directory is `c:\usr\test1\` and this feature is enabled, then the user's Home Directory will appear as the root "/" when a **PWD** is performed from the client. The user will not be able to **CWD** or **CDUP** from the user's home directory. If this feature is not enabled, then the user's Home Directory will appear as `/usr/test1/` and the user will be able to **CWD** or **CDUP** to other directories in the system, provided that the user has adequate permissions.

- **Show Hidden Files** - When enabled, Titan will display hidden files in the directory listings that are sent to the client.

- **Hide directories users cannot enter** - When enabled, Titan will not display any folder/directory entries that the user does not have adequate rights to.

- **Allow modification of file dates/times via MDTM command** - When enabled, the user will be permitted to modify file dates/times by issuing the **SITE MDTM** command from the client.

- **Allow modification of file dates/times via MFMT command** - When enabled, the user will be permitted to modify file dates/times by issuing the **SITE MFMT** command from the client.

- **Allow modification of file dates/times via MFCT command** - When enabled, the user will be permitted to modify file dates/times by issuing the **SITE MFCT** command from the client.

- **STOU Prefix** - The **STOU** command requires that all unique filenames have a prefix. Use the text box to customize the prefix.

- **STOU Extension** - Use the text box to customize the STOU suffix (file extension).

- **Ban the following file types** - Select this check box to ban certain file types. Use the text box to specify a list of file types that are prohibited from the server. Users will not be permitted to upload or rename a file that matches this filter. You must **separate multiple entries** with a **semicolon**.

Group Files/Directories Directory Access Tab

The **Directory Access** tab provides you with the ability to grant or deny access to folders on the server. These settings can be configured at the **Server**, **Group**, and **User levels**. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access these settings at the **Server** level, in the tree pane expand the **server**, click **Files/Directories**, and then click the **Directory Access** tab.

To access these settings at the **Group** or **User** level, in the tree pane select the **Group** or **User**, click **Files/Directories**, and then click the **Directory Access** tab.

Directory Access is broken down into two categories, **File Permissions** and **Folder Permissions**:

● File Permissions

- **Read/Download Files** - Allows users to download files from the server. (**RETR**)
- **Write/Upload New Files** - Allows users to upload files to the server (**STOR**, **STOU**). **Note:** This right applies to **new** files only; it does not allow the user to replace/append existing files.
- **Append/Replace Files** - Allows users to upload/replace/append existing files on the server. **Note:** This rule applies only to **existing** files; this rule does not apply to new files.
- **Delete Files** - Allows users to delete existing files from the server (**DELE**).
- **Rename Files** - Allows users to rename existing files on the server (**RNFR**).

● Folder Permissions

- **Create Subdirectories** - Allows users to create subdirectories within the current folder/directory (**MKD**).
- **Remove Subdirectories** - Allows users to remove subdirectories from the current folder/directory (**RMD**). **Note:** The sub-directory must be empty to remove it. This permission usually coincides to the **Delete Files** permission. The combination of these two permissions will permit members to delete directory trees.
- **Can View Directory Listing** - Allows users to generate a directory listing of the contents of the folder (**LIST/NLST**).
- **Apply Rights to Subdirectories** - Enable this check box to have these permissions, both **File** and **Folder**, propagated to all subdirectories of the specified path.

Titan FTP Server will apply permissions as follows:

1. Directory Access rules for the **USER** are loaded first.
2. Directory Access rules for all **GROUPS** in which the user is a member of are loaded next. These **GROUPS** are loaded in the order in which they appear in the **Groups tab** of the **User Configuration dialog**. If a duplicate folder is encountered for which there are already **Dir Access Perms** specified, the **SUM** of the permissions is used.
3. Directory Access rules for the **SERVER** are loaded last. Again, if duplicates are located, they are summed together.

Example:

User A is a member of **Group 1** and **Group 2** for **Server S**.

User A has the following permissions: **/pub/** - **Read** permissions

Group 1 has the following permissions: **/pub/** - **Write** permissions

Group 2 has the following permissions: **/pub/** - **NO ACCESS AT ALL**

Server S has the following permissions: **/pub/** - **LIST** permissions

Outcome: **User A** will have **READ**, **WRITE** and **LIST** permissions to the **/pub/** folder.

Group Files/Directories Virtual Folders Tab

Virtual folders are used to link or map external folders into a user's directory space. For Windows users, think of a virtual folder as a Windows shortcut. The link appears in one location whereas the data lives elsewhere. For UNIX users, virtual folders are very similar to symbolic links. Virtual Folders are commonly used to map network shares or folders from different drive letters into the server directory structure.

Virtual folders can be added at the **Server**, **Group**, or **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access the **Virtual Folders** tab at the **Server** level, in the tree pane expand the **Server**, click **Files/Directories**, and then click the **Virtual Folders** tab.

To access the **Virtual Folders** tab at the **Group** or **User** level, in the tree pane select the **Group** or **User**, click **Files/Directories**, and then click the **Virtual Folders** tab.


Using Virtual Folders

1. Select the **Server/Group/User** where you wish to add the Virtual Folder.
2. Select the **Files/Directories** node.
3. Select the **Virtual Folders** tab.
4. Click on the **Add** button to display the **New Virtual Folder** wizard.
5. To select the fully qualified path, browse to the actual/real **physical** folder. You may select a folder on your local computer, or you may choose a network folder that has been previously shared. If you are mapping a UNC share, make sure that the account under which the Titan FTP Service is running has access to the UNC. Click **Next**.
6. After you have selected a **Physical Path**, the **Virtual Path** will be filled in automatically. Click **Next**.
7. Select the default **Access Rights** for this new Virtual Folder using the check boxes. Click **Next**.
8. The Actual Path of the folder is displayed and the Virtual Path is displayed. You can change the **Folder Name** as it will appear under the virtual path, or you can leave the default name (which is the same as the Actual Path name). Click **Finish** to generate the virtual folder mapping.
9. The Virtual Path and the Actual Path are now displayed in the **Virtual Folders** tab. Click **Apply**.

Note: Virtual folder updates are not real-time. If a user is currently connected to the server, and you make changes to the Virtual Folder list, users will need to log out and then log back in to the system to see the virtual folder changes.

Note: If mapping a UNC share, make sure that the account under which the Titan Service is running has access to the UNC. Otherwise, you will need to add the appropriate username and password under the [UNC Accounts](#) tab.

[More information about using virtual folders](#)



- If you attempt to create a virtual folder to a mapped network drive, Titan will replace the drive mapping with the actual UNC name. This is done because the Titan Service does not have access to mapped drives, only to UNC shares.
- If you would like more information about configuring group level virtual folders in Titan, see the Using Group Level Folders [Quick Start Guide](#).

Group Files/Directories Disk Quotas Tab

The **Disk Quotas tab** is used to configure disk quota limits. These values may be set at the **Server**, **Group**, or **User level**. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access these settings at the **Server** level, in the tree pane expand the **Server**, click **Files/Directories**, and then click the **Disk Quotas** tab.

To access these settings at the **Group** or **User** level, in the tree pane expand the **Server**, expand the **Group** or **User**, click **Files/Directories**, and then click the **Disk Quotas** tab.

- **Use Inherited Setting** - This check box appears on the **Group** and **User** levels. When selected, the values will be inherited. Clear this check box to override the inherited values.

- **Enable Disk Quotas** - When enabled, the server will have a limited amount of storage space.

- **Current Disk Usage** - Shows the amount of storage currently being used by this server.

- **Maximum Disk Usage** - Specifies the maximum number of Kilobytes of data that the server will be permitted to store.

- **Recalculate** - Recalculates the current disk usage.

- **Explore** - Launches Explorer so that you can browse the server's **data directory**. This is useful if you need to purge invalid files or if you want to see where space is being used.

- **Free Files List** - Type a list of files/file types that will not be included in the quota calculations.

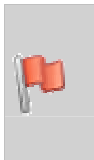
Group Security

Group FTPS/SSL Tab

The Group **FTPS/SSL** tab is used to configure secure SSL/TLS settings for this group. For more information about the SSL features supported by Titan, See [Server FTPS/SSL settings](#) and [SSL Support](#).

To access the Group **FTPS/SSL** tab, in the tree pane, expand **Groups**, click the **Group>Security**, and then click the **FTPS/SSL** tab.

- **Permit SSL/TLS access for this user** - When selected, this option allows the group members to use SSL/TLS protocol with Titan Server.
- **Require all connections from users in this group to be secure** - When selected, this option requires the group members to connect with Titan using secure protocols only.
- **Allow secure site-to-site (FXP) file transfers (CPSV/SSCN)** - When selected, this options allows **FXP site-to-site** transfers. FXP indicates a direct server-to-server file transfer.
- **Encrypt data channel by default (Used if client does not issue PROT command)** - Protect the data channel by encrypting all communications by default. If a client sends a PROT command, that value with override this setting.


	<ul style="list-style-type: none">• Titan does not support SSL v2 and will reject any client that attempts a connection• FTPS/SSL features are available only in the Professional and Enterprise editions of Titan Server.
--	---

Group SFTP/SSH

The Group **SFTP/SSH** tab is used to configure SFTP/SSH settings. See [SFTP Support](#) for more information.

To access the **SFTP/SSH** tab, expand the **Group**, click **Security**, and then click the **FTPS/SSL** tab.

Permit SFTP (SSH's Secure File Transfer Protocol) access for this group - When selected, allows members of this group to connect securely using SFTP.

	SFTP/SSH features are only available in the Enterprise edition of Titan Server.
---	---


Configuring Users

User Authentication Overview

Titan FTP Server currently supports native Titan Authentication and Windows NT/SAM Authentication.

By establishing a separate user account for each user in the system, you provide a secure isolated area for each user to store their files. You will also be able to fine tune permissions and access rights for the various users in your system.

For more information about User Authentication methods, see [User Authentication Options](#).

	<ul style="list-style-type: none">• Windows NT/SAM Authentication is only available in the Enterprise Edition of Titan FTP Server.• For more information about configuring NT/SAM user authentication, see the Titan NT/SAM quick start guide.
---	---

Creating New Users

If your server is configured for standard Titan Authentication, then new user accounts can be created directly within the Administration program.

To create a new user using the Titan Administrator:

● In the tree pane, right click on **Users** and select **New User**. The **New User Wizard** will launch.

or

● In the tree pane select **Users**, and then on the **Users** tab click **New User**.

or

● In the tree pane, select **Users**. On the menu bar, select **Users, New User Wizard**.

Deleting Users

If your server is configured for standard Titan Authentication, you can delete users using the Titan Administrator.

To delete an existing user using the Administration program:

- In the tree pane, right-click on the user you want to delete and select **Delete User** from the context menu. You will be prompted to confirm the deletion of the user.

or

- In the tree pane, select the **user** you want to delete. From the menu bar, click **Users**, **Delete User**. You will be prompted to confirm the deletion of the user.

or

- In the tree pane, click **Users**. A list of users is generated and displayed in the tab pane. Select the user from the list of available users and click **Delete**. You will be prompted to confirm the deletion of the user.

Note: Deleting a user is **permanent**. If you are unsure about deleting a user account from the system, you may want to **disable** the account. When you delete a user account, Titan will also remove the user from all groups.

User Properties

Users Tab

The **Users** tab is used to view currently defined users. You can also launch the New User Wizard from this tab.

To access the **Users** tab, in the tree pane, click **Users**.

New User Wizard

1. Click **New User**.
2. Type **Users Full Name**.
3. Type the **Username**.
4. Use the drop-down arrow to select the **Password Type**:
 - **Standard** - This is the most common. The user must type the password that was assigned to them.
 - **Anonymous** - Allows the user to type anything for a password; however, they must type something. Null or zero length passwords are not permitted.
 - **OTP S/key with MD4** - When selected, the server requires an MD4 hashed S/Key version of the password. This feature must be supported by the FTP Client. [WebDrive®](#) supports S/Key.
 - **OTP S/key with MD5** - When selected, the server requires an MD5 hashed S/Key version of the password. This feature must be supported by the FTP Client. [WebDrive®](#) supports S/Key.
5. Type the user's **Password**.
6. Confirm the **Password**.
7. Type the **Email Address**. Click **Next**.
8. Use the arrow button to **select the groups** that this user is a member of. Click **Next**.
9. Select the user's **directory settings** and then click **Finish**.

User General Tab

The **User General** tab is used to manage basic configuration options for the user.

To access the **User General** tab, in the tree pane select the **user**, and then click the **User General** tab.

- **Use Inherited Setting** - When selected, the settings will be inherited. Clear this check box to override the inherited values.
- **Account Enabled** - When selected, the user account is enabled and available for use. When this check box is not selected, the account is disabled and the user will not be permitted to log into the server.
- **Username** - The user's unique name that is used to log into the server. User names must be unique per server.
- **Password** - The user's password that is used to log in to the system. The user's password is not displayed in the Titan Administrator for security reasons. **Note:** Passwords are **case sensitive** and must be at least four characters long with no spaces.
- **Confirm Password** - Type the user's password a second time to verify that it is correct. **Note:** Passwords are case sensitive.
- **Password Type** - Various password types are supported:
 - **Standard** - This is the most common. The user must type the password that was assigned to them.
 - **Anonymous** - Allows the user to type anything for a password. However, they must type something; null or zero length passwords are not permitted.
 - **OTP S/key with MD4** - When selected, the server requires an **MD4 hashed S/Key** version of the password. This feature will also need to be supported by the FTP Client. [WebDrive](#) supports **S/Key**.
 - **OTP S/key with MD5** - When selected, the server requires an **MD5 hashed S/Key** version of the password. This feature also needs to be supported by the FTP Client. [WebDrive](#) supports **S/Key**.

- **Users Full Name** - This text box contains the user's full name. Used for informational purposes only.
- **Email address** - This text box contains the user's e-mail address. The user's e-mail address can be used by the Event Handler system to contact this user.
- **Home Directory** - This is the root/home directory for the user. When the user logs into the system, the user starts in this directory by default. The user's home directory must be a **fully qualified local path** or a **UNC** name that refers to a remote share on a network server. **Note:** if you specify a UNC name, the NT User Account being used by the Titan Service must have adequate rights to access that shared folder.
- **Account Expiration Date** - Select the check box to enable this feature. Specify the date the user's account expires. On this date, the account is disabled and the user is no longer permitted to log in to the system.
- **Always Allow Login** - When enabled (and the server is configured to allow a maximum number of connections and that limit has been reached) the user will still be permitted to log in to the server. This feature is beneficial for Administrators and Users who must have access to the server at all times.
- **Notes** - The **Notes** feature allows you to store extra text information about the user. Text is limited to 1024 characters.

User Groups Tab

The **Groups** tab for each user is used to manage group membership.

To access the **Groups** tab, in the tree pane select the **user**, and then click the **Groups** tab.

- **Member of** - Lists the groups that this user is currently a member of.
- **Not a Members of** - Lists the groups that the user is currently not a member of.

User FTP Tab

The **FTP** tab is used to configure FTP settings used by Titan. These values can be set at the **Server**, **Group**, and **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access the **FTP** tab at the **User** or **Group** levels, in the tree pane select the **group** or **user**, and then click the **FTP** tab.

- **Use Inherited Setting** - This check box appears on the **Group** and **User** levels. When selected, the settings will be inherited. Clear this check box to override the inherited values.
- **Enable FTP access on this server** - This option enables/disables the user's ability to connect to the server using FTP.

For information about **Server** FTP settings, see the [Server FTP Tab](#).

User Connections

User Connections General tab

The **Connections General** tab is used to configure general connection settings. These values can be set at the **Server**, **Group**, and **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access the **Connections General** tab at the **Server** level, expand the **Server** in the tree pane, click **Connections**, and then click the **Connections General** tab.

To access these settings at the **Group** or **User** level, in the tree pane click the **Group** or **User**, click **Connections**, and then click the **Connections General** tab.

- **Use Inherited Setting** - This check box appears on the **Group** and **User** levels. When selected, the settings will be inherited. Clear this check box to override the inherited values.
- **Max Concurrent Connections** - Specifies the maximum number of concurrent sessions that may be established.
- **Idle Connection Time-out** - Specifies the maximum amount of time, in minutes, to wait before dropping a user due to inactivity.
- **Max Connections/IP** - Specifies the maximum number of concurrent connections a user can establish from any given IP address.
- **Max Upload Speed** - Specifies the maximum KPS (Kilobytes Per Second) upload speed that the server will allow from the user. If the user attempts to exceed this bandwidth allotment, the server will pause their transfer and slow it down to the proper speed.
- **Max Download Speed** - Specifies the maximum KPS (Kilobytes Per Second) download speed that the server will allow data to be sent to the user.
- **Max Uploads/Session** - Specifies the maximum number of files that can be uploaded per session. Once this limit has been reached, the user will not be able to upload/replace any files until they log out and then log back in.
- **Max Downloads/Session** - Specifies the maximum number of files that can be downloaded per session. Once this limit has been reached, the user will not be able to download any files until they log out and then log back in.
- **Max File Upload Size** - Specifies the maximum file size that can be uploaded by the user. Any attempt to upload a larger file will be aborted and the file will be deleted from the system.
- **Max File Download Size** - Specifies the maximum file size that can be downloaded by this user. Any attempts to download files larger than this value will be denied.

User Connections Advanced tab

The **Connections Advanced** tab is used to configure advanced connection settings. These values can be set at the **Server**, **Group**, and **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access the **Connections Advanced** tab at the **Server** level, expand the **Server** in the tree pane, click **Connections**, and then click the **Connections Advanced** tab.

To access these settings at the **Group** or **User** level, in the tree pane click the **Group** or **User**, click **Connections**, and then click the **Connections Advanced** tab.

- **Use Inherited Setting** - This check box appears on the **Group** and **User** levels. When selected, the values will be inherited. Clear this check box to override the inherited values.

- **Disable account after X invalid password attempts** - When enabled, the user account will be disabled after the specified number of consecutive incorrect password attempts.

- **Kick User after X consecutive bad commands** - When enabled, the user connection will be dropped after the user types the specified number of invalid commands.
 - **Disable user account afterwards** - When selected, the user's account will be disabled after they are dropped from the system.
 - **Ban access from this IP Address once this happens** - When selected, the user's IP address will be added to the **IP Access Restrictions** list and will no longer have access to the server.

- **Allow PASV Mode Connections** - When enabled, allows the server to be placed in **PASV** mode by the client. If this feature is disabled and the client attempts to issue a **PASV** command, they will receive a **502 Not Implemented** response.

- **Allow EPSV Mode Connections** - Similar to the **PASV** command, but used for IP v6 addressing. When enabled, allows the server to be placed in EPSV mode by the client. If this feature is disabled and the client attempts to issue an EPSV command, they will receive a 502 Not Implemented response.
 - **Limit PASV Port Range** - Allows you to specify a specific range of ports that the server will use when the user issues a **PASV** command. This is useful if the FTP Server is behind a firewall/router and you only want to open a specified range of ports for use by the server.

- **Delete Partially Uploaded Files** - When enabled, the server will delete any files that are not successfully uploaded. For example, if a **STOR** or **STOU** does not complete successfully, the file will be deleted from the server.

- **Block Anti-Timeout Schemes** - When enabled, the server will ignore **NOOP** commands as attempts are made to keep the connection alive.

- **Block FTP Bounce Attacks and FXP** - When enabled, the server will not accept any data connections from IP addresses other than that of the user's primary connection.

- **Allow change of password (SITE PSWD)** - When enabled, users are permitted to change their password using the **SITE PSWD** command.

User Connections IP Access Tab

The **IP Access** tab is used to configure IP Access restrictions. These values can be set at the **Server**, **Group**, and **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access the **IP Access** tab at the **Server** level, in the tree pane expand the Server, click **Connections**, and then click the **IP Access** tab.

To access these settings at the **Group** or **User** level, in the tree pane select the **Group** or **User**, click **Connections**, and then click the **IP Access** tab.

- **Use Inherited Setting** - This check box appears on the **Group** and **User** levels. If checked, the values will be inherited. Clear this check box to override the inherited values.
- **Enable IP Access Restrictions** - When enabled, IP Access restrictions will be applied at this level whenever a user attempts to connect.
- **Grant/Deny access by default** - Select the default action that will be applied at this level when a connection attempt is made.
- **Except the addresses listed below** - (Exception List) Enter a list of IP addresses that will be the exception to the default rule. For example, you can **Deny Access by default** and then type a single IP address. This will be the only IP address that the user will be permitted to connect from.

User Connections Upload/Download Ratios Tab

The **Upload/Download Ratios tab** is used to configure Upload/Download ratios. These values may be set at the **Server**, **Group**, or **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access the **Upload/Download Ratios** tab at the **Server** level, in the tree pane expand the **Server** and click **Connections**, and then click the **Upload/Download Ratios** tab.

To access these settings at the **Group** or **User** level, in the tree pane select the **Group** or **User**, click **Connections**, and then click the **Upload/Download Ratios** tab.

● **Use Inherited Setting** - This check box appears on the **Group** and **User** levels. When selected, the values will be inherited. Clear this check box to override the inherited values.

● **Enable UL/DL Ratios** - When enabled, the user will have **Upload/Download ratios** applied to their sessions.

- **Count # of Files Per Session** - Ratios will be applied at the file level on a per-session basis. Each time the user disconnects from the server, the counters are reset. If the user has multiple concurrent sessions open, statistics are applied for each individual session.
- **Count KBytes Per Session** - Ratios will be applied at the file size level on a per-session basis. Each time the user disconnects from the server, the counters will be reset. If the user has multiple concurrent sessions open, statistics are applied for each individual session.
- **Count # of Files across All User Sessions** - Ratios will be applied at the file level across all concurrent sessions for the user.
- **Count KBytes across All User Sessions** - Ratios will be applied at the file size level across all concurrent sessions for the user.

● **Ratios** - Enter the UL to DL ratio for the user. For example, you may want to require that the user upload two files for every one file that they download, so the ratio would be **Uploads 2, Downloads 1**. You may also want to require that the user upload 1MB of data for each 1MB of data that they download, so the ratio would be **Upload 1000, Download 1000**, and select the **Count KBytes** option.

● **Free Files List** - Type a list of files/file types that will be excluded from the ratios.

User Custom Messages Tab

The **Messages tab** is used to configure custom messages at the **Server**, **Group**, and **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access these settings at the **Server** level, in the tree pane expand the **Server**, click **Connections**, and then click the **Messages** tab.

To access these settings at the **Group** or **User** level, in the tree pane select the **Group** or **User**, click **Connections**, and then click the **Messages** tab.

- **Custom Message Type** - Select the message to customize. The current message will be displayed in the **Message Text** field.

- **Use Default Setting** - At the **Server** level, you can select this check box to use the default message for the selected Message Type. Clear this check box to customize the message.

- **Use Inherited Setting** - At the **Group** and **User** level, you can select this check box to use the inherited message for the selected **Message type**. Clear this check box to customize the message.

- **Custom Message** - Type the message to be displayed when this event occurs. Custom messages are limited to **1024 bytes**.

Custom Message Variables

User Files/Directories

Files/Directories Tab

The **Files/Directories** tab is used to configure general file/directory settings. These values can be set at the **Server**, **Group**, and **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access these settings on the **Server** Level, in the tree pane expand the **server**, click **Files/Directories**, and then click the **Files/Directories** tab.

To access these settings at the **Group** or **User** level, in the tree pane select the **Group** or **User**, click **Files/Directories**, and then click the **Files/Directories** tab.

- **Use Inherited Setting** - These configuration options appear on the **Group** and **User** levels. When selected, the values will be inherited. Clear this check box to override the inherited values.
- **Lock User(s) in Home Directory** - Enable this feature to prevent the user from leaving the user's home directory and venturing "up" the tree. If the user's Home Directory is `c:\usr\test1\` and this feature is enabled, then the user's Home Directory will appear as the root "/" when a **PWD** is performed from the client. The user will not be able to **CWD** or **CDUP** from the user's home directory. If this feature is not enabled, then the user's Home Directory will appear as `/usr/test1/` and the user will be able to **CWD** or **CDUP** to other directories in the system, provided that the user has adequate permissions.
- **Show Hidden Files** - When enabled, Titan will display hidden files in the directory listings that are sent to the client.
- **Hide directories users cannot enter** - When enabled, Titan will not display any folder/directory entries that the user does not have adequate rights to.
- **Allow modification of file dates/times via MDTM command** - When enabled, the user will be permitted to modify file dates/times by issuing the **SITE MDTM** command from the client.
- **Allow modification of file dates/times via MFMT command** - When enabled, the user will be permitted to modify file dates/times by issuing the **SITE MFMT** command from the client.
- **Allow modification of file dates/times via MFCT command** - When enabled, the user will be permitted to modify file dates/times by issuing the **SITE MFCT** command from the client.
- **STOU Prefix** - The **STOU** command requires that all unique filenames have a prefix. Use the text box to customize the prefix.

- **STOU Extension**- Use the text box to customize the STOU suffix (file extension).
- **Ban the following file types** - Select this check box to ban certain file types. Use the text box to specify a list of file types that are prohibited from the server. Users will not be permitted to upload or rename a file that matches this filter. You must **separate multiple entries** with a **semicolon**.

User Directory Access Tab

The **Directory Access** tab provides you with the ability to grant or deny access to folders on the server. These settings can be configured at the **Server**, **Group**, and **User levels**. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access these settings at the **Server** level, in the tree pane expand the **server**, click **Files/Directories**, and then click the **Directory Access** tab.

To access these settings at the **Group** or **User** level, in the tree pane select the **Group** or **User**, click **Files/Directories**, and then click the **Directory Access** tab.

Directory Access is broken down into two categories, **File Permissions** and **Folder Permissions**:

● File Permissions

- **Read/Download Files** - Allows users to download files from the server. (**RETR**)
- **Write/Upload New Files** - Allows users to upload files to the server (**STOR**, **STOU**). **Note:** This right applies to **new** files only; it does not allow the user to replace/append existing files.
- **Append/Replace Files** - Allows users to upload/replace/append existing files on the server. **Note:** This rule applies only to **existing** files; this rule does not apply to new files.
- **Delete Files** - Allows users to delete existing files from the server (**DELE**).
- **Rename Files** - Allows users to rename existing files on the server (**RNFR**).

● Folder Permissions

- **Create Subdirectories** - Allows users to create subdirectories within the current folder/directory (**MKD**).
- **Remove Subdirectories** - Allows users to remove subdirectories from the current folder/directory (**RMD**). **Note:** The sub-directory must be empty to remove it. This permission usually coincides to the **Delete Files** permission. The combination of these two permissions will permit members to delete directory trees.
- **Can View Directory Listing** - Allows users to generate a directory listing of the contents of the folder (**LIST/NLST**).
- **Apply Rights to Subdirectories** - Enable this check box to have these permissions, both **File** and **Folder**, propagated to all subdirectories of the specified path.

Titan FTP Server will apply permissions as follows:

1. Directory Access rules for the **USER** are loaded first.
2. Directory Access rules for all **GROUPS** in which the user is a member of are loaded next. These **GROUPS** are loaded in the order in which they appear in the **Groups tab** of the **User Configuration dialog**. If a duplicate folder is encountered for which there are already **Dir Access Perms** specified, the **SUM** of the permissions is used.
3. Directory Access rules for the **SERVER** are loaded last. Again, if duplicates are located, they are summed together.

Example:

User A is a member of **Group 1** and **Group 2** for **Server S**.

User A has the following permissions: **/pub/** - **Read** permissions

Group 1 has the following permissions: **/pub/** - **Write** permissions

Group 2 has the following permissions: **/pub/** - **NO ACCESS AT ALL**

Server S has the following permissions: **/pub/** - **LIST** permissions

Outcome: **User A** will have **READ**, **WRITE** and **LIST** permissions to the **/pub/** folder.

User Virtual Folders Tab

Virtual folders are used to link or map external folders into a user's directory space. For Windows users, think of a virtual folder as a Windows shortcut. The link appears in one location whereas the data lives elsewhere. For UNIX users, virtual folders are very similar to symbolic links. Virtual Folders are commonly used to map network shares or folders from different drive letters into the server directory structure.

Virtual folders can be added at the **Server**, **Group**, or **User** level. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access the **Virtual Folders** tab at the **Server** level, in the tree pane expand the **Server**, click **Files/Directories**, and then click the **Virtual Folders** tab.

To access the **Virtual Folders** tab at the **Group** or **User** level, in the tree pane select the **Group** or **User**, click **Files/Directories**, and then click the **Virtual Folders** tab.


Using Virtual Folders

1. Select the **Server/Group/User** where you wish to add the Virtual Folder.
2. Select the **Files/Directories** node.
3. Select the **Virtual Folders** tab.
4. Click on the **Add** button to display the **New Virtual Folder** wizard.
5. To select the fully qualified path, browse to the actual/real **physical** folder. You may select a folder on your local computer, or you may choose a network folder that has been previously shared. If you are mapping a UNC share, make sure that the account under which the Titan FTP Service is running has access to the UNC. Click **Next**.
6. After you have selected a **Physical Path**, the **Virtual Path** will be filled in automatically. Click **Next**.
7. Select the default **Access Rights** for this new Virtual Folder using the check boxes. Click **Next**.
8. The Actual Path of the folder is displayed and the Virtual Path is displayed. You can change the **Folder Name** as it will appear under the virtual path, or you can leave the default name (which is the same as the Actual Path name). Click **Finish** to generate the virtual folder mapping.
9. The Virtual Path and the Actual Path are now displayed in the **Virtual Folders** tab. Click **Apply**.

Note: Virtual folder updates are not real-time. If a user is currently connected to the server, and you make changes to the Virtual Folder list, users will need to log out and then log back in to the system to see the virtual folder changes.

Note: If mapping a UNC share, make sure that the account under which the Titan Service is running has access to the UNC. Otherwise, you will need to add the appropriate username and password under the [UNC Accounts](#) tab.

[More information about using virtual folders](#)



- If you attempt to create a virtual folder to a mapped network drive, Titan will replace the drive mapping with the actual UNC name. This is done because the Titan Service does not have access to mapped drives, only to UNC shares.
- If you would like more information about configuring group level virtual folders in Titan, see the Using Group Level Folders [Quick Start Guide](#).

Files/Directories Disk Quotas Tab

The **Disk Quotas** tab is used to configure disk quota limits. These values may be set at the **Server**, **Group**, or **User level**. See [Inheritance](#) and [Shared Attributes](#) for more information.

To access these settings at the **Server** level, in the tree pane expand the **Server**, click **Files/Directories**, and then click the **Disk Quotas** tab.

To access these settings at the **Group** or **User** level, in the tree pane expand the **Server**, expand the **Group** or **User**, click **Files/Directories**, and then click the **Disk Quotas** tab.

- **Use Inherited Setting** - This check box appears on the **Group** and **User** levels. When selected, the values will be inherited. Clear this check box to override the inherited values.

- **Enable Disk Quotas** - When enabled, the server will have a limited amount of storage space.

- **Current Disk Usage** - Shows the amount of storage currently being used by this server.

- **Maximum Disk Usage** - Specifies the maximum number of Kilobytes of data that the server will be permitted to store.

- **Recalculate** - Recalculates the current disk usage.

- **Explore** - Launches Explorer so that you can browse the server's **data directory**. This is useful if you need to purge invalid files or if you want to see where space is being used.

- **Free Files List** - Type a list of files/file types that will not be included in the quota calculations.

Security

User SSL Tab


The User **SSL** tab is used to configure secure SSL/TLS settings for this user. For more information about the SSL features supported by Titan, see [Server FTPS/SSL settings](#) and [SSL Support](#).

To access the User **SSL** tab, in the tree pane expand the **user**, click **Security**, and then in the tab pane, select the **SSL** tab.

- **Permit SSL/TLS access for this user** - When selected, this option allows the user to use SSL/TLS protocol with Titan Server.
- **Require all connections from this user to be secure** - When selected, this option allows the user to connect with Titan using secure protocols only.
- **Allow secure site-to-site (FXP) file transfers (CPSV/SSCN)** - When selected, this option allows **FXP site-to-site** transfers. FXP indicates a direct server-to-server file transfer.
- **Encrypt data channel by default (Used if client does not issue PROT command)** - Protect the data channel by encrypting all communications by default. If a client makes a PROT command, that value will override this setting.

User Certificate Settings

- **Trust the following certificate for this user** - Use the drop-down arrow to select a certificate or click **Certificate Management** to create/import a certificate for this user.
- **Certificate Store Folder** - This is the location where Titan will store all certificates for this server. **Note:** Local paths and UNC shares are supported; do not use a mapped drive because mapped network drives are not accessible from the Titan service.



- If you would like more information about configuring Titan Server FTPS/SSL settings, please see the Titan FTPS/SSL & Certificate Management [Quick Start Guide](#).
- Titan does not support SSL v2 and will reject any client that attempts a connection using SSL v2.
- FTPS/SSL features are available only in the Professional and Enterprise Editions of Titan FTP Server.

User SFTP/SSH Tab

The User **SFTP/SSH** tab is used to configure SFTP/SSH settings for this user. See [SFTP Support](#) for more information.

To access the User **SFTP/SSH** tab, in the tree pane expand the user and click **Security**, and then in the tab pane, click the **SFTP/SSH** tab.


● **Permit SFTP (SSH's Secure File Transfer Protocol) access for this user** - When enabled, the user will be permitted to access this server using SFTP.

User Host Key Settings

● **Use the following host key set for this user** - Use the drop-down arrow to select a host key for the user. If public host keys are required for connection, the user-supplied host key must match the host key that has been associated with the user. If the host keys do not match, the user connection is denied and the connection is terminated.

● **Host Key Management** - Click **Host Key Management** to create or import host keys for this user.

● **Host key Folder** - Location where host keys are stored.

A small icon of two orange pushpins on a grey background, indicating a pinned note or important information.

- If you would like more information about configuring Titan FTP Server SFTP/SSH settings, please see the [Titan SFTP/SSH & Host Key Management Quick Start Guide](#).
- SFTP/SSH features are available only in the Enterprise Edition of Titan FTP Server.

Advanced Features

Event Handling

Introduction to Event Handling

Event Handlers are used to trigger customized actions based on specific events and conditions. Events are fired whenever anything of importance occurs on the server, such as a user logging in or a file being uploaded.

Event Handlers Examples/Tutorials

- Notify the administrator every time the server is started.
- Create an event log that logs every event that occurs on the server.
- Create user accounts that expire after a single use.
- Send an e-mail when a user account is near expiration.
- Create custom log files for each user account.
- Move/off-load a file on the server after it has been uploaded.

Additional Examples

- Constrain certain users/groups to a limited command set.
- Create a directory log for each shared folder, so that whenever anything happens in that folder, an entry is added.
- Create command logs for every command entered.
- Kick any lower-class users (members of a less privileged group) if the number of connections to the server reaches a certain threshold.
- Compress or decompress a file after it has been uploaded (requires third party compression software).
- Encrypt or decrypt a file after it has been uploaded (requires third party encryption software).
- Scan a file for viruses after it has been uploaded (requires third party virus scanning software).
- Send an e-mail to the administrator every hour showing the current status of the server.



- For more information about configuring events in Titan, see the **Using Events to Thwart Hackers Quick Start Guide**.
- Event Management features are only available in the Enterprise edition of Titan Server.

Events

Events are organized into a relational hierarchy that provides a mechanism to handle events generically. For example, **All events** can be used as a basis to handle **every** server event. More events may be added in the future to address protocol enhancements.

Note: The handling of **Before Events** differs from all other events.

- **All Events** - Encompasses every server event.
- **Scheduled event** – This is the parent event for scheduled events. Use scheduled events to set up a one-time or repeatable event that will run based on the current time. This event type must have a corresponding **Scheduled time elapsed** condition.
 - **Scheduled standard event** – Use this event type to schedule a standard event that will send an email to one user.
 - **Scheduled broadcast event** – Use this event type to schedule an email broadcast to a group of users.
- **Server events** - This is the parent event for any server-specific events.
 - **Server start succeeded** - The server has started successfully.
 - **Server start failed** - The server failed to start correctly.
 - **Server start failed -- Statistics failed** - The server failed to start correctly because the Statistics subsystem failed to initialize.
 - **Server start failed -- FTP failed** - The server failed to start correctly because the FTP subsystem failed to initialize. This often indicates a port conflict.
 - **Server start failed -- FTPS failed** - The server failed to start correctly because the FTPS subsystem failed to initialize. This often indicates a port conflict.
 - **Server start failed -- SFTP failed** - The server failed to start correctly because the SFTP subsystem failed to initialize. This often indicates a port conflict.
 - **Server start failed -- HTTP failed** - The server failed to start correctly because the HTTP subsystem failed to initialize. This often indicates a port conflict.
 - **Server start failed -- HTTPS failed** - The server failed to start correctly because the HTTPS subsystem failed to initialize. This often indicates a port conflict.
 - **Server stopped** - The server has stopped. This event is triggered if the server is stopped or if the server is restarted (in which case, a **Server start** event would immediately follow).
 - **Server log rotated** - The server log has been rotated. This occurs every time the server starts, any time the log rotation period expires, or any time the administrator manually rotates the log.
 - **Before command processed** - Occurs any time a command is sent from the client to the server. By handling this event, you can block unwanted commands or simply keep track of commands issued, etc. [List of FTP commands](#), [List of SFTP commands](#).
 - **Return code sent to client** - Occurs after the server has processed a client command and is about to return a status code. [List of FTP return codes](#), [List of SFTP return codes](#).
 - **Connection attempt succeeded** - A client connection attempt has succeeded. This event is fired before any username/password verification, so it is still possible that the connection will be closed afterwards.
 - **Connection attempt failed** - A client connection attempt has failed.

- **Connection attempt failed -- Banned IP address** - A client connection attempt has failed because the IP address is banned at the server level.
- **Connection attempt failed -- Server Hammering** - A client connection attempt has failed because the IP address is banned at the server level due to DoS/Hammering.
- **Disconnection** - A client connection has been closed.
 - **Disconnection -- User quit** - A client connection has been closed because the user has quit.
 - **Disconnection -- User kicked** - A client connection has been closed because the user has been kicked out by the server.
 - **Disconnection -- Timeout** - A client connection has been closed because it has exceeded the idle timeout limit.
 - **Disconnection -- Server stopped** - A client connection has been closed because the server is stopping.
- **Security events** - This is the parent event for any security-specific events.
 - **Security events -- FXP blocked** - A FXP attempt has been blocked. This event will only occur if FXP is disabled, otherwise the FXP attempt will proceed as normal.
 - **Security events -- Server hammered** - This event occurs when the server is repeatedly hit with requests (also known as **flooding** or **hammering**). After reaching a certain threshold, the server will not accept these requests and this event will be fired.
- **User events** - This is the parent event for any user-specific events.
 - **User login attempt successful** - A user has successfully logged in.
 - **User login attempt failed** - A user login attempt has failed.
 - **User login attempt failed -- Bad username** - The username specified does not exist.
 - **User login attempt failed -- Bad password** - The password specified does not match the correct password for the supplied username.
 - **User login attempt failed -- Banned IP address** - A user login attempt failed because the IP address is banned for the specified username.
 - **User login attempt failed -- Max connections** - A user login attempt failed because the maximum number of connections has been reached.
 - **User login attempt failed -- Max connections/IP** - A user login attempt failed because the maximum number of connections for that IP address has been reached.
 - **User login attempt failed -- FTPS login failed - A user FTPS login attempt failed.**
 - **User FTPS login attempt failed -- User not using FTPS** - A user FTPS login attempt failed because the user is not using FTPS.

- **User FTPS login attempt failed -- FTPS not permitted for this user** - A user FTPS login attempt failed because the specified username is not permitted FTPS access on the server.
- **User FTPS login attempt failed -- Cert load failed** - A user FTPS login attempt failed because the user's certificate cannot be loaded by the server.
- **User FTPS login attempt failed -- Cert not supplied** - A user FTPS login attempt failed because the user did not supply a certificate.
- **User FTPS login attempt failed -- Cert did not match** - A user FTPS login attempt failed because the user-supplied certificate does not match the certificate on the server.
- **User login attempt failed -- SFTP login failed** - A user SFTP login attempt failed.
 - **User SFTP login attempt failed -- User not using SFTP** - A user SFTP login attempt failed because the user is not using SFTP.
 - **User SFTP login attempt failed -- SFTP not permitted for this user** - A user SFTP login attempt failed because the specified username is not permitted SFTP access.
 - **User SFTP login attempt failed -- Host key load failed** - A user SFTP login attempt failed because the user's host key cannot be loaded by the server.
 - **User SFTP login attempt failed -- Host key not supplied** - A user SFTP login attempt failed because the user did not supply a host key.
 - **User SFTP login attempt failed -- Host key did not match** - A user SFTP login attempt failed because the user-supplied host key does not match the host key on the server.
- **User login attempt failed -- Account disabled** - A user login attempt failed because the specified username is disabled.
 - **User login attempt failed -- Account disabled - Account expired** - A user login attempt failed because the specified username is expired. Expired accounts are always disabled.
- **Bad command issued** - A user has issued a bad command. The command was either unrecognized, the syntax was incorrect, or the command was invalid based on the connection state.
- **User IP address auto-banned** - An IP address has been automatically banned by the server due to excessive bad commands.
- **User account auto-disabled** - A user account has been automatically disabled by the server due to excessive bad commands.
- **User account created** - A user account has been created.
- **User account deleted** - A user account has been deleted.
- **User changed password** - The user has changed their account password.

- **File events** - This is the parent event for any file-specific events.
 - **File download/read** - This is the parent event for any file download/read events.
 - **File download/read -- Before download/read** - Occurs before the downloading of a file begins. By handling this event, you can block unwanted downloads.
 - **File download/read -- Download/read successful** - A file has been successfully downloaded.
 - **File download/read -- Download/read failed** - A file download has failed.
 - **File download/read failed -- Bad filename** - A file download has failed because the specified filename is a reserved name.
 - **File download/read failed -- Insufficient permissions** - A file download has failed because the user has insufficient access rights.
 - **File download/read failed -- File not found** - A file download has failed because the specified file was not found.
 - **File download/read failed -- Insufficient upload/download ratios** - A file download has failed because the user has insufficient upload/download ratios. The user must first upload enough files/bytes to satisfy the ratio.
 - **File download/read failed -- Max session downloads** - A file download has failed because the user has downloaded the maximum number of files allowed in a session.
 - **File download/read failed -- Max download size** - A file download has failed because the file is larger than the maximum download size.
 - **File upload/write** - This is the parent event for any file upload/write events.
 - **File upload/write -- Before upload/write** - Occurs before the uploading of a file begins. By handling this event, you can block unwanted uploads.
 - **File upload/write -- Upload/write successful** - A file has been successfully uploaded.
 - **File upload/write -- Upload/write failed** - A file upload has failed.
 - **File upload/write failed -- Bad filename** - A file upload has failed because the specified filename is a reserved name.
 - **File upload/write failed -- Insufficient permissions** - A file upload has failed because the user has insufficient access rights.
 - **File upload/write failed -- Banned file** - A file upload has failed because the specified filename matches the banned file filter.
 - **File upload/write failed -- Disk quota limit** - A file upload has failed because the disk quota limit has been exceeded.
 - **File upload/write failed -- Max session uploads** - A file upload has failed because the user has uploaded the maximum number of files allowed in a session.

- **File upload/write failed -- Max upload size** - A file upload has failed because the file is larger than the maximum upload size.
- **File append** - This is the parent event for any file append events.
 - **File append -- Before append** - Occurs before appending to a file begins. By handling this event, you can block unwanted appends.
 - **File append -- Append successful** - A file append has been successful.
 - **File append -- Append failed** - A file append has failed.
 - **File append failed -- Bad filename** - A file append has failed because the specified filename is a reserved name.
 - **File append failed -- Insufficient permissions** - A file append has failed because the user has insufficient access rights.
 - **File append failed -- Banned file** - A file append has failed because the specified filename matches the banned file filter.
 - **File append failed -- Disk quota limit** - A file append has failed because the disk quota limit has been exceeded.
 - **File append failed -- Max session uploads** - A file append has failed because the user has uploaded the maximum number of files allowed in a session. This only occurs when an upload is really masquerading as an append.
 - **File append failed -- Max upload size** - A file append has failed because the file is larger than the maximum upload size.
- **File delete** - This is the parent event for any file delete events.
 - **File delete -- Before delete** - Occurs before deleting a file. By handling this event, you can block unwanted deletes, or back up files to another location.
 - **File delete -- Delete successful** - A file has been successfully deleted.
 - **File delete -- Delete failed** - A file delete has failed.
 - **File delete failed -- Bad filename** - A file delete has failed because the specified filename is a reserved name.
 - **File delete failed -- Insufficient permissions** - A file delete has failed because the user has insufficient access rights.
- **File rename** - This is the parent event for any file rename events.
 - **File rename -- Before rename** - Occurs before renaming a file. By handling this event, you can block unwanted renaming.
 - **File rename -- Rename successful** - A file has been successfully renamed.
 - **File rename -- Rename failed** - A file rename has failed.
 - **File rename failed -- Bad filename** - A file rename has failed because the specified filename is a reserved name.
 - **File rename failed -- Insufficient permissions** - A file rename has failed because the user has insufficient access rights.

- **File rename failed -- Banned file** - A file rename has failed because the specified filename matches the banned file filter.
- **File rename failed -- Source file not found** - A file rename has failed because the specified source file was not found.
- **File rename failed -- Dest file already exists** - A file rename has failed because the specified destination file already exists.
- **Partially uploaded file deleted** - A partially uploaded file has been deleted by the server.
- **Directory events** - This is the parent event for any directory-specific events.
 - **Directory created** - This is the parent event for any directory created events.
 - **Directory created -- Before directory create** - Occurs before creating a directory. By handling this event, you can block unwanted directory creation.
 - **Directory created -- Directory create successful** - A directory has been successfully created.
 - **Directory created -- Directory create failed** - A directory create has failed.
 - **Directory create failed -- Bad directory name** - A directory create has failed because the specified directory name is a reserved name.
 - **Directory create failed -- Insufficient permissions** - A directory create has failed because the user has insufficient access rights.
 - **Directory create failed -- Directory already exists** - A directory create has failed because the directory already exists.
 - **Directory removed** - This is the parent event for any directory removed events.
 - **Directory removed -- Before directory remove** - Occurs before removing a directory. By handling this event, you can block unwanted directory removal.
 - **Directory removed -- Directory remove successful** - A directory has been successfully removed.
 - **Directory removed -- Directory remove failed** - A directory remove has failed.
 - **Directory remove failed -- Bad directory name** - A directory remove has failed because the specified directory name is a reserved name.
 - **Directory remove failed -- Insufficient permissions** - A directory remove has failed because the user has insufficient access rights.
 - **Directory remove failed -- Directory not found** - A directory remove has failed because the directory was not found.
 - **Directory contents listed** - This is the parent event for any directory list events. These events occur under the following circumstances: **FTP:** [LIST](#), [NLST](#), [MLST](#), [MLSD](#) **SFTP:** [READDIR](#).

- **Directory contents listed -- Before directory list** - Occurs before a directory listing. By handling this event, you can block unwanted directory listing.
- **Directory contents listed -- Directory list successful** - A directory listing has been successful.
- **Directory contents listed -- Directory list failed** - A directory listing has failed.
 - **Directory list failed -- Bad directory name** - A directory listing has failed because the specified directory name is a reserved name.
 - **Directory list failed -- Insufficient permissions** - A directory listing has failed because the user has insufficient access rights.
- **Limit events** - This is the parent event for any limit-specific events.
 - **Timeout limit hit** - The idle timeout limit has been reached.
 - **Connection limit hit** - The maximum number of connections limit has been reached.
 - **Connections/IP limit hit** - The maximum number of connections per IP address limit has been reached.
 - **Session upload limit hit** - The session upload limit has been reached.
 - **Session download limit hit** - The session download limit has been reached.
 - **Upload size limit hit** - The file upload size limit has been reached.
 - **Download size limit hit** - The file download size limit has been reached.
 - **Max upload speed limit hit** - The maximum file upload speed limit has been reached.
 - **Max download speed limit hit** - The maximum file download speed limit has been reached.
 - **Upload/Download ratio limit hit** - The upload/download ratio limit has been reached. No more downloads will be allowed until more uploads occur.
 - **Bad command limit hit** - The bad command limit has been reached.
 - **Disk quota limit hit** - The disk quota limit has been reached.



- For more information about configuring events in Titan, see the **Using Events to Thwart Hackers Quick Start Guide**.
- Event Management features are only available in the Enterprise edition of Titan Server.

Conditions

Conditions can be created to fine-tune Event Handler definitions to handle specific cases. Different conditions apply to each event type, and it is important to be careful not to create conditions that are never satisfied.

- **Server command in list** - Use this condition to specify one or more server commands. Note that there are both FTP and SFTP commands listed. If *any* of the specified server commands match the current server command, the condition will be satisfied. Wildcards can be used. For example, specifying **A*** in the list would cause the condition to be satisfied if the server command started with an **A**.
- **Server return code in list** - Use this condition to specify one or more server return codes. Note that there are both FTP and SFTP return codes listed. If *any* of the specified server return codes match the current server return code, the condition will be satisfied. Wildcards can be used. For example, specifying **4*** in the list would cause any **400-level** return codes to satisfy the condition.
- **User name** - Use this condition to specify one or more usernames. If *any* of the specified usernames match the username of the account that caused the event to be triggered, the condition will be satisfied. Wildcards can be used. For example, specifying ***z*** in the list would cause the condition to be satisfied for any username that contains a **z**.
- **User group membership** - Use this condition to specify one or more groups. If *any* of the specified groups match the group membership of the account that caused the event to be triggered, the condition will be satisfied. Wildcards can be used. For example, specifying **grp*** in the list would cause the condition to be satisfied for any group names that begin with **grp**.
- **User account expiration date** - Use this condition to specify a time range for account expiration date values. The time range can be either less than or greater than a set number of days/hours/minutes/seconds. If the account that caused the event to be triggered has a valid expiration date that is within the specified range, the condition will be satisfied. For example, specifying **Less than** and **20** days means that the condition will be satisfied if the user account is expiring within the next 20 days. This condition is very useful to remind users and administrators that an account will soon be expiring.
- **User enabled** - Use this condition to specify whether an account is expired. If the account that caused the event to be triggered matches this specified value, the condition is satisfied.
- **Max failed login attempts** - Use this condition to specify whether the user has attempted the maximum allowed login attempts. If the account that caused the event to be triggered matches this specified value, the condition is satisfied. Invalid password attempts are set on the [Connections Advanced](#) tab.
- **File name** - Use this condition to specify one or more file names. If *any* of the specified file names match the current file name being accessed, the condition will be satisfied. Wildcards can be used. For example, specifying ***.txt** in the list would cause the condition to be satisfied for any filename with a **.txt** extension.
- **Directory name** - Use this condition to specify one or more directory names. If *any* of the specified directory names match the directory name being accessed, the condition will be satisfied. Wildcards can be used. For example, specifying ***f*** in the list would cause the condition to be satisfied for any directory name that contains an **f**.

● **IP address** - Use this condition to specify one or more IP addresses. If *any* of the specified IP addresses match the IP address of the account that caused the event to be triggered, the condition will be satisfied. Wildcards can be used. For example, specifying **123.*.*.*** in the list would cause the condition to be satisfied for any IP address that starts with **123**.

● **Connection time** - Use this condition to specify a time range for the connection time. The time range can be either less than or greater than a set number of days/hours/minutes/seconds. If the connection that caused the event to be triggered has a connection time that is within the specified range, the condition will be satisfied. For example, specifying **More than** and **1** days means that the condition will be satisfied if the connection has been alive for more than 1 day.

● **Connection idle time** - Use this condition to specify a time range for the connection idle time (time since the last command was received). The time range can be either less than or greater than a set number of days/hours/minutes/seconds. If the connection that caused the event to be triggered has a connection idle time that is within the specified range, the condition will be satisfied. For example, specifying **More than** and **1** hours means that the condition will be satisfied if the connection has been idle for more than 1 hour.

● **Scheduled time elapsed** - Use this condition in conjunction with the **Scheduled event**. For a one-time scheduled event, simply specify the date/time you wish the event to occur. This one-time condition is satisfied if the current system time is beyond the **First occurrence** time. For a repeatable scheduled event, specify the date/time of the first occurrence, and the repeat interval. The repeatable condition is satisfied if the current system time is beyond the **First occurrence** time and it has been at least **Repeat Interval** units of time since the condition was last satisfied. The repeatable condition will keep track of the last time the condition was satisfied so that it does not repeat more than once in any **Repeat Interval** units of time.



- For more information about configuring events in Titan, see the **Using Events to Thwart Hackers Quick Start Guide**.
- Event Management features are only available in the Enterprise edition of Titan Server.

Actions

Actions are used to implement the server's response to a specific event.

- **Do not process command** - Causes the command that fired the event to be cancelled. This action is only valid with **Before events**. For example, you could use this action to cancel any DELE command sent by a specific user.

- **Send email** - Causes an e-mail to be sent, provided you have a SMTP mail server that will handle the request. SMTP mail server configuration is set at the server level on the **EMail Server** tab. The following fields may be set for the **Send e-mail** action:
 - From** - Specifies the e-mail address from which the e-mail will be sent.
 - To** - Specifies the e-mail address to which the e-mail will be sent. For example, you could specify the **%USEREMAIL%** variable in the case where a user-triggered event has occurred and you wish to notify that user. You can send the e-mail to more than one person by separating addresses with a semicolon. Example: **bob@abc.com; joe@abc.com**
 - Subject** - Specifies the text that will go into the e-mail subject line.
 - Body** - Specifies the text that will go into the body of the e-mail.

- **Flag for admin review** - Causes a **Flagged Event** to be created, which will appear on the **Flagged Events** tab.

- **Run file/script** - Launches a file/script with optional command line parameters. The following fields may be set for the **Run file/script** action:
 - File/script** - Specifies the location of the file/script to be run.
 - Parameters** - Specifies any command line arguments to the file. Add each parameter to the parameter list in the order in which they should be passed to the file/script. Any parameters that could contain a space should always be wrapped in double quotes.
Examples:
"C:\long file name.txt"
"%FILEPATH%"

- **Write to custom logfile** - Writes a **message** to a logfile. The following fields may be set for the **Write to custom logfile** action:
 - Logfile** - Specifies the location of the file that the message will be written to.
 - Log text** - Specifies the message that will be written to the logfile.

- **Ban IP address** - Bans an IP address. The following fields may be set for the **Ban IP address** action:
 - IP address** - Specifies the IP address to ban.

- **Kick user** - Kicks a user from the server. The following fields may be set for the **Kick user** action:
 - Username** - Specifies the username of the account you wish to kick from the server.

- **Disable user account** - Disables a user account. The following fields may be set for the **Disable User account** action:
 - Username** - Specifies the username of the account you wish to disable on the server.

"Before" Events

Events with the word **Before** in them are handled differently than other events. These events wait for further processing to continue to see if it will be cancelled by a **Do not process command** action. For example, you can set up an Event Handler to not allow the downloading of any files that begin with an **x**. Only **Before** events can trigger a **Do not process command** action.

List of "Before" Events:

- Before command processed
- Before download/read
- Before upload/write
- Before append
- Before delete
- Before rename
- Before directory create
- Before directory remove
- Before directory list



- For more information about configuring events in Titan, see the **Using Events to Thwart Hackers Quick Start Guide**.
- Event Management features are only available in the Enterprise edition of Titan Server.

Event Message Variables

Titan Event Management provides you with the ability to create a custom log message that is based on the current state of the server by entering any of the following variables into the **Log Text** field. This functionality is very similar to the custom messages found under the [Connections->Messages](#) tab.

%TIME%

Current system date/time.

%TIMEONLY%

Current system time only.

%DATEONLY%

Current system time only.

%VER%

The current version of the server. Version information will be displayed as **X.XX**.

%DOMAINNAME%

The name of the domain on which the server is running.

%COMMAND%

The last command to be received by the server in the context of this server/connection.

%RETCODE%

The last return code to be sent to the client in the context of this server/connection.

%EVENTTEXT%

Text describing the current event that is being processed.

%FILEPATH%

The complete filename, including path, of the last file to be accessed by the server/connection.

%FILENAME%

The complete file name of the last file to be accessed by the server/connection (no path).

%DIRPATH%

The path in which the last server/connection access occurred.

%OLDFILENAME%

The old file name (from a rename).

%ATTRIBUTESCSV%

File attributes, tagged pair of values in CSV format.

%ATTRIBUTESXML%

File attributes, tagged pair of values in XML format.

%SERVERNAME%

The name of the server.

%SERVERTZ%

The Time Zone configuration value for the server.

%SSUPCNT%

Server stats: total files uploaded.

%SSDNCNT%

Server stats: total files downloaded.

%SSUPKB%

Server stats: total KB uploaded.

%SSDNKB%

Server stats: total KB downloaded.

%SSUPTIME%

Server stats: total time uploading.

%SSDNTIME%

Server stats: total time downloading.

%SSUPKPS%

Server stats: total KB per second uploading.

%SSDNKPS%

Server stats: total KB per second downloading.

%SSTOTCNT%

Server stats: total files transferred.

%SSTOTKB%

Server stats: total KB transferred.

%SSTOTTIME%

Server stats: total time transferring.

%SSTOTKPS%

Server stats: total KB per second (bandwidth).

%SSSTARTTIME%

Server stats: the time that the server started (GMT).

%SSRUNTIME%

Server stats: the total days, hours, minutes, and seconds that the server has been up and running.

%SSTOTCXN%

Server stats: the total number of open connections on the server.

%SSTOTUSRS%

Server stats: the total number of distinct user connections on the server.

%EXPDATE%

User connection stats: the account expiration date for the logged in user.

%CIP%

User connection stats: the IP address for the client/user.

%FILESIZEB%

User connection stats: current file transfer size in Bytes.

%%MAXUPNUM%%

User connection stats: the configuration value for the **Max Uploads Per Session**. Since this is a Server configuration value that can be overridden/customized at the **User** level, this variable will display the setting that is currently in use. This is useful when the user exceeds the **Max Uploads allowed Per session**. You can display the maximum setting to the user and inform the user that the value has been exceeded.

%MAXDNNUM%

User connection stats: the configuration value for the **Max Downloads Per Session**. This is a **Server** configuration value that can be overridden/customized at the **User** level, so this variable will display the setting that is currently in use. This is useful when the user exceeds the **Max Downloads allowed Per Session**. You can display the maximum setting to the user and inform the user that the value has been exceeded.

%MAXUPSI ZKB%

User connection stats: the configuration value for the **Max Uploadable File Size**. This is a **Server** configuration value that can be overridden/customized at the **User** level, so this variable will display the setting that is currently in use. This is useful when the user exceeds the **Max Uploadable Size allowed**. You can display the maximum setting to the user and inform the user that this value has been exceeded.

%MAXDNSI ZKB%

User connection stats: the configuration value for the **Max Downloadable File Size**. Since this is a Server configuration value that can be overridden/customized at the **User** level, this variable will display the setting that is currently in use. This is useful when the user exceeds the **Max Downloadable File Size**. You can display the maximum setting to the user and inform the user that this value has been exceeded.

%USERNAME%

User connection stats: the user name of the logged in user.

%USEREMAIL%

User connection stats: the e-mail address for the logged in user.

%USERFULLNAME%

User connection stats: the full name of the logged in user.

%USERHOMEDIR%

User connection stats: the home directory for the logged in user.

%USTOTTIME%

User connection stats: the count of the total time that was spent transferring data since the user logged in to the server.

%USTOTKKB%

User connection stats: the count of the total kilobytes (KB) transferred (uploaded and downloaded) since the user logged in to the server.

%USTOTCNT%

User connection stats: the count of the total number of files transferred (uploaded and downloaded) since the user logged in to the server.

%USUPKPS%

User connection stats: the total average bandwidth utilization for uploads, in kilobytes-per-second, since the user logged in to the server.

%USDNKPS%

User connection stats: the total average bandwidth utilization for downloads, in kilobytes-per-second, since the user logged in to the server.

%USUPTIME%

User connection stats: the count of the total elapsed time that has been spent uploading files to the server since the user logged in to the server.

%USDNTIME%

User connection stats: the count of the total elapsed time that has been spent downloading files to the server since the user has logged into the server.

%USUPKB%

User connection stats: the count of the total kilobytes (KB) of data that has been successfully uploaded to the server since the user logged in to the server.

%USDNKB%

User connection stats: the count of the total kilobytes (KB) of data that was successfully downloaded from the server since the user logged in to the server.

%USDNCNT%

User connection stats: the count of the total number of files that have been successfully downloaded since the user logged in to the server.

%USUPCNT%

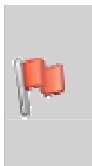
User connection stats: the count of the total number of files that have been successfully uploaded since the user logged in to the server.

%USTOTKPS%

User connection stats: the total kilobytes-per-second (KPS) value since the user logged in to the server. This is the overall bandwidth utilization for the user.

%USTOTCXN%

User connection stats: the total number of current connections for the logged in user.



- For more information about configuring events in Titan, see the **Using Events to Thwart Hackers Quick Start Guide**.
- Event Management features are only available in the Enterprise edition of Titan Server.

Inherited Settings

Titan FTP Server supports the ability to set any [Shared Attributes](#) at the **Server**, **Group**, or **User level**.

If an attribute is set at the **Server** level, every Group can inherit the value.

If an attribute is set at the **Group** level, every user who is a member of that group can inherit the value.

Inheritance at the **Group** and **User** level is controlled by the **Use Inherited Setting** check box. For certain attributes such as directory permissions and virtual folders, the **Server** and **Group** level values are **always** inherited.

Advantages of Using Inherited Settings

Setting attribute values at the **Server** or **Group** level and then enabling **Use Inherited** settings at the **Group** or **User** level provides several advantages, including organization and simplification. Rather than setting values for each **user**, you can set these values at the **Server** or **Group** level and minimize the amount of work required when you add new users or updating attribute values.

Shared Attributes

Shared Attributes are variables that can be set at the **Server**, **Group**, or **User** level. These variables are common to all three levels and can be **Inherited**. **Shared Attributes** are located under the **Connections** and **Files/Directories** nodes for **servers**, **groups**, and **users**. Whenever possible, set these values at the **Server** or **Group** level.

Attribute	Description
AcctDisabledMsg	Message to display if a disabled user attempts to log in to the server. For multi-line messages, separate each line with a ' '.
AcctExpiredMsg	Message to display if an expired user attempts to log in to the server. For multi-line messages, separate each line with a ' '.
AllowMDTM	Allow users to modify file date/time settings via MDTM command. 0 - No, 1 - Yes.
AllowPASV	Allow Passive (PASV) mode connections. 0 - No, 1 - Yes.
BadPass	Disable Users account after BadPassCnt number of invalid password attempts. 0 - Do not disable the account, 1 - Yes disable the account.
BadPassCnt	Number of bad password attempts allowed before disconnecting the user. This value is ignored unless BadPass is enabled.
BadPassVal	Reserved.
BanFileTypes	Controls the use of banned file types. 0 - Disabled, 1 - Enabled.
BannedFileTypeMsg	Message to be displayed when the user attempts to upload a file that is banned. For multi-line messages, separate each line with a ' '.
BannedFileTypes	A semicolon separated list of file types to ban. For example: *.exe;*.txt;
BannedMsg	Message to display to users who are banned from the server. For multi-line messages, separate each line with a ' '.
BannerMsg	Message returned to the FTP client upon initial connection to the server. For multi-line messages, separate each line with a ' '.
BaseLogDir	The fully qualified path to store the FTP Server log files.
BlockAntiTimeout	Block anti-timeout schemes. 0 - Disabled, 1 - Enabled.
BlockFXP	Block site-to-site/FXP file transfer attempts. 0 - Disabled, 1 - Enabled.
BlockFXPMsg	Message to display to the user when a site-to-site/FXP transfer is detected. For multi-line messages, separate each line with a ' '.
CanChangePwd	Allow the user to change their own password. 0 - Disabled, 1 - Enabled.
DelPartFiles	Delete partially uploaded files. 0 - Disabled, 1 - Enabled.

DenyAccessByDefault	When enabled, the default setting is to deny access to all IP addresses except those listed in the IPAccessList attribute. Ignored unless EnableIPAccess is enabled. 0 - Disabled, 1 - Enabled.
DiskQuota	Control disk quota usage. 0 - Disabled, 1 - Enabled.
DiskQuotaCnt	Disk quota, in KB. Ignored unless DiskQuota is enabled.
EnableIPAccess	Controls access checking by IP Address. If enabled, DenyAccessByDefault and IPAccessList will be used to determine who can access the FTP Server. 0 - Disabled, 1 - Enabled.
EnableRatios	Controls upload/download ratios. 0 - Disabled, 1 - Enabled.
ExceededQuotaMsg	Message displayed to the user if the user's disk quota has been exceeded. For multi-line messages, separate each line with a ' '.
IdleTimeout	Controls the monitoring of the IdleTimeoutCnt . 0 - Disabled, 1 - Enabled.
IdleTimeoutCnt	Number of minutes to elapse before kicking an idle user off the system. This value is ignored unless IdleTimeout is enabled.
IPAccessList	List of IP addresses and masks that are used in conjunction with the DenyAccessByDefault and EnableIPAccess attributes to determine who can access the Server. Each entry in the list is composed of a pair of values. The first value is whether to Deny (0) or Permit (1) access from the address. The second value is the address mask. Example1: 1 1.2.3.4 1 12.*.23.34 1 192.168.1.100-255 Example2: 1 127.0.0.1 0 63.*.*.*
KickedMsg	Message displayed to the user when they are kicked off the server. For multi-line messages, separate each line with a ' '.
KickUser	Controls whether or not to kick users off the server after a certain number of consecutive bad commands. 0 - Disabled, 1 - Enabled.
KickUserBanIP	If enabled, the IP will be banned once the user has been kicked. This value is ignored unless KickUser is enabled. 0 - Disabled, 1 - Enabled.
KickUserCnt	The number of consecutive invalid commands to allow before kicking the user off the system.
KickUserDisable	If enabled, the user account will be disabled once the user has been kicked off the server. This value is ignored unless KickUser is enabled. 0 - Disabled, 1 - Enabled.
LimitPASVPORT	Controls the use of the range of ports available for use by the PASV command. 0 - Disabled (use all ports), 1 - Enabled (use only those ports in PasvPortStart and PasvPortEnd).
LockHome	Lock the user in their home directory. The user home directory is treated like the root of a drive, and the user will not be able to traverse up the directory structure.
MaxConnects	Controls the Maximum Number of Connections. 0 - Disabled, 1 - Enabled.
MaxConnectsCnt	If MaxConnects is enabled, this value determines the maximum number of simultaneous connections permitted.
MaxConnectsIP	Controls the Maximum Number of Connections Per IP. 0 - Disabled, 1 - Enabled.

MaxConnectsIPCnt	If MaxConnectsIP is enabled, this value determines the maximum number of simultaneous connections per IP.
MaxConnectsIPMsg	Message to display to a user if the MaxConnectsIPCnt value has been exceeded. For multi-line messages, separate each line with a ' '.
MaxConnectsMsg	Message to display to a User if the MaxConnectsCnt value has been exceeded. For multi-line messages, separate each line with a ' '.
MaxDownloadKPS	Controls the Maximum Download KB Per Second. 0 - Disabled, 1 - Enabled.
MaxDownloadKPSCnt	If MaxDownloadKPS is enabled, this value determines the maximum bandwidth allowed for downloading on the server (in KPS). This value will be divided up evenly among all sessions logged in. If this value is 10KPS and there are 5 active sessions, each session will max out at 2KPS during downloads
MaxDownloadNum	Controls the Maximum Number of Files Downloaded Per Session. 0 - Disabled, 1 - Enabled.
MaxDownloadNumCnt	If MaxDownloadNum is enabled, this value is the maximum number of files that a User is permitted to download during a session.
MaxDownloadNumMsg	Message to display to the user once they have exceeded their MaxDownloadNum for the session. For multi-line messages, separate each line with a ' '.
MaxDownloadSize	Controls the Maximum Download File Size. 0 - Disabled, 1 - Enabled.
MaxDownloadSizeCnt	If MaxDownloadSize is enabled, this value contains the maximum file size in KB that can be downloaded.
MaxDownloadSizeMsg	Message to display to the user when they attempt to download a file larger than MaxDownloadSizeCnt . For multi-line messages, separate each line with a ' '.
MaxUploadKPS	Controls the Maximum Upload KB Per Second. 0 - Disabled, 1 - Enabled.
MaxUploadKPSCnt	If MaxUploadKPS is enabled, this value determines the maximum bandwidth allowed for uploading on the server (in KPS). This value will be divided up evenly among all sessions logged in to the Server. If this value is 10KPS and there are 5 active sessions, each session will max out at 2KPS during uploads.
MaxUploadNum	Controls the Maximum Number of Files Uploaded Per Session. 0 - Disabled, 1 - Enabled.
MaxUploadNumCnt	If MaxUploadNum is enabled, this value is the maximum number of files that a user can upload during any given session.
MaxUploadNumMsg	Message to display to the user once they have exceeded MaxUploadNum for the session. For multi-line messages, separate each line with a ' '.
MaxUploadSize	Controls the Maximum Upload File Size. 0 - Disabled, 1 - Enabled.
MaxUploadSizeCnt	If MaxUploadSize is enabled, this value contains the maximum file size in KB that can be uploaded.
MaxUploadSizeMsg	Message to display to the user when they attempt to upload a file larger than MaxUploadSizeCnt . For multi-line messages, separate each line with a ' '.

Notes	Comments about this Server/Group/User.
PASVPortEnd	If LimitPasvPort is enabled, this value is the ending value for the allowable passive port range.
PASVPortStart	If LimitPasvPort is enabled, this value is the starting value for the allowable passive port range.
QuitMsg	Message to display to the user in response to the QUIT command. For multi-line messages, separate each line with a ' '.
QuotaFreeFileList	If DiskQuota is enabled, this value specifies a list of file types (*.exe;*.txt;*.doc) that are considered free and will not count against the disk quota.
RatioDLCnt	If EnableRatios is enabled, this value represents the number of downloads.
RatioFreeFileList	If EnableRatios is enabled, this value specifies a list of file types that are considered free and will not count against the ratios.
RatioType	If EnableRatios is enabled, this value determines the type of ratios to use. 0 - Count number of files per each individual session. 1 - Count number of KB per each individual session. 2 - Count number of files across all sessions for a user. 3 - Count number of KB across all sessions for a user.
RatioULCnt	If EnableRatios is enabled, this value represents the number of uploads required before a download can occur.
ShowHiddenFiles	Controls the display of hidden files in directory listings. If this feature is enabled, files marked as Hidden in the local file system will show up during a generic directory request from a client. If this feature is not enabled, the client will need to explicitly specify the -H flag during a LIST command. 0 - Disabled, 1 - Enabled.
SSLAllowFXPS	If SSLEnabled is enabled, this flag determines if secure Site-To-Site FXPS transfers are permitted. 0 - No, do not allow FXPS. 1 - Yes, allow FXPS.
SSLBannerMsg	Message to display to the user when they connect to the server over the SSLImplicitPort . For multi-line messages, separate each line with a ' '.
SSLCertName	Name of the SSL certificate to be used at this level.
SSLCertPassword	Password for the SSL certificate. For security reasons, this attribute can only be set, not retrieved.
SSLDisabledMsg	Message to display to the user if they attempt to initiate an SSL connection and SSLEnabled is disabled. For multi-line messages, separate each line with a ' '.
SSLEnabled	Controls if SSL is enabled. 0 - Disabled, 1 - Enabled.
SSLProtData	This flag controls the default protection mode of the data connection in SSL. If the client does not specify a PROT P or PROT C , this flag will be used to determine if the data connection is encrypted.
SSLRequired	1 - Require any FTP connection to use SSL. 0 - Allow FTP connections that are not secured with SSL.

SSLRequiredMsg	Message to display to the user if they attempt to initiate an FTP connection without SSL and SSLRequired is enabled. For multi-line messages, separate each line with a ' '.
STOUPrefix	Prefix to be used during the generation of a unique file name used in the STOU command.
STOUExtension	File extension to be used during the generation of a unique file name used in the STOU command.
WelcomeMsg	Message to display to a user once they have successfully logged in. For multi-line messages, separate each line with a ' '.
SFTPEnabled	1 - Enables SFTP support. 0 - Disables SFTP support.
SFTPHostKeyName	The name of the Host Key file, stored in the Host Key Folder, used by Titan during SFTP handshaking.
SFTPHostKeyPassword	Password for the SFTP Host Key. For security reasons, this attribute can only be set, not retrieved.
SFTPUseCompression	1 - Titan will use Zlib compression if also supported by the client.
SFTPCipherList	A list of encryption ciphers, in order, that Titan will present to the SFTP client during the handshaking/negotiation phase of the connection. The following ciphers are currently supported: 3des-cbc blowfish-cbc aes128-cbc aes192-cbc aes256-cbc aes128-ctr aes192-ctr aes256-ctr cast128-cbc arcfour rijndael-cbc@lysator.liu.se
SFTPMAList	A list of MAC algorithms, in order, that Titan will present to the SFTP client during the handshaking/negotiation phase of the connection. The following MAC algorithms are currently supported: md5 sha1 ripemd160 ripemd160@openssh.com sha1-96 md5-96

Custom Message Variables

Custom message variables are used to dynamically generate strings that reflect the current status of variables.

General Message Variables

%TIME%

Current system date/time.

%TIMEONLY%

Current system time only.

%DATEONLY%

Current system time only.

%VER%

The current version of the server. Version information will be displayed as **X.XX**.

%DOMAINNAME%

The name of the domain on which the server is running. By default, this is the name of the physical machine, but it can be customized by the Administrator to be any text string.

%COMMAND%

The last command to be received by the server in the context of this server/connection.

%RETCODE%

The last return code to be sent to the client in the context of this server/connection.

%EVENTTEXT%

Text describing the current event that is being processed.

%FILEPATH%

The complete filename, including path, of the last file to be accessed by the server/connection.

%FILENAME%

The complete file name of the last file to be accessed by the server/connection (no path).

%DIRPATH%

The path in which the last server/connection access occurred.

%OLDFILENAME%

The old file name (from a rename).

%ATTRIBUTESCSV%

File attributes, tagged pair of values in CSV format.

%ATTRIBUTESXML%

File attributes, tagged pair of values in XML format.

Server Message Variables

%SERVERNAME%

The name of the server.

%SERVERTZ%

The Time Zone configuration value for the server.

%SSUPCNT%

Server stats: total files uploaded.

%SSDNCNT%

Server stats: total files downloaded.

%SSUPKB%

Server stats: total KB uploaded.

%SSDNKB%

Server stats: total KB downloaded.

%SSUPTIME%

Server stats: total time uploading.

%SSDNTIME%

Server stats: total time downloading.

%SSUPKPS%

Server stats: total KB per second uploading.

%SSDNKPS%

Server stats: total KB per second downloading.

%SSTOTCNT%

Server stats: total files transferred.

%SSTOTKB%

Server stats: total KB transferred.

%SSTOTTIME%

Server stats: total time transferring.

%SSTOTKPS%

Server stats: total KB per second (bandwidth).

%SSSTARTTIME%

Server stats: the time that the server started (GMT).

%SSRUNTIME%

Server stats: the total days, hours, minutes, and seconds that the server has been up and running.

%SSTOTCXN%

Server stats: the total number of open connections on the server.

%SSTOTUSRS%

Server stats: the total number of distinct user connections on the server.

User/Connection Message Variables**%EXPDATE%**

User connection stats: the account expiration date for the logged in user.

%CIP%

User connection stats: the IP address for the client/user.

%FILESIZEB%

User connection stats: current file transfer size in Bytes.

%%MAXUPNUM%%

User connection stats: the configuration value for the **Max Uploads Per Session**. Since this is a **Server** configuration value that can be overridden/customized at the **User** level, this variable will display the setting that is currently in use. This is useful when the user exceeds the **Max Uploads allowed Per session**. You can display the maximum setting to the user and inform the user that the value has been exceeded.

%%MAXDNNUM%

User connection stats: the configuration value for the **Max Downloads Per Session**. This is a **Server** configuration value that can be overridden/customized at the **User** level, so this variable will display the setting that is currently in use. This is useful when the user exceeds the **Max Downloads allowed Per Session**. You can display the maximum setting to the user and inform the user that the value has been exceeded.

%%MAXUPSIZEKB%

User connection stats: the configuration value for the **Max Uploadable File Size**. This is a **Server** configuration value that can be overridden/customized at the **User** level, so this variable will display the setting that is currently in use. This is useful when the user exceeds the **Max Uploadable Size allowed**. You can display the maximum setting to the user and inform the user that this value has been exceeded.

%%MAXDNSIZEKB%

User connection stats: the configuration value for the **Max Downloadable File Size**. Since this is a **Server** configuration value that can be overridden/customized at the **User** level, this variable will display the setting that is currently in use. This is useful when the user exceeds the **Max Downloadable File Size**. You can display the maximum setting to the user and inform the user that this value has been exceeded.

%%USERNAME%

User connection stats: the user name of the logged in user.

%%USEREMAIL%

User connection stats: the e-mail address for the logged in user.

%%USERFULLNAME%

User connection stats: the full name of the logged in user.

%%USERHOMEDIR%

User connection stats: the home directory for the logged in user.

%%USTOTTIME%

User connection stats: the count of the total time that was spent transferring data since the user logged in to the server.

%%USTOTKB%

User connection stats: the count of the total kilobytes (KB) transferred (uploaded and downloaded) since the user logged in to the server.

%%USTOTCNT%

User connection stats: the count of the total number of files transferred (uploaded and downloaded) since the user logged in to the server.

%%USUPKPS%

User connection stats: the total average bandwidth utilization for uploads, in kilobytes-per-second, since the user logged in to the server.

%%USDNKPS%

User connection stats: the total average bandwidth utilization for downloads, in kilobytes-per-second, since the user logged in to the server.

%USUPTIME%

User connection stats: the count of the total elapsed time that has been spent uploading files to the server since the user logged in to the server.

%USDNTIME%

User connection stats: the count of the total elapsed time that has been spent downloading files to the server since the user has logged into the server.

%USUPKB%

User connection stats: the count of the total kilobytes (KB) of data that has been successfully uploaded to the server since the user logged in to the server.

%USDNKB%

User connection stats: the count of the total kilobytes (KB) of data that was successfully downloaded from the server since the user logged in to the server.

%USDNCNT%

User connection stats: the count of the total number of files that have been successfully downloaded since the user logged in to the server.

%USUPCNT%

User connection stats: the count of the total number of files that have been successfully uploaded since the user logged in to the server.

%USTOTKPS%

User connection stats: the total kilobytes-per-second (KPS) value since the user logged in to the server. This is the overall bandwidth utilization for the user.

%USTOTCXN%

User connection stats: the total number of current connections for the logged in user.

CRC File Integrity Checking

Titan FTP Server supports the ability for the FTP client to verify that the file has been successfully transferred to the server without corruption. This is accomplished by requesting that the server perform a **Cyclic Redundancy Check**, or **CRC-32**, on the file once it has been uploaded.

What is CRC?

A CRC performs a mathematical calculation on a block of data and returns a number that represents the content and organization of that data. The CRC returns a number that uniquely identifies the data. CRC is the operation that generates a "fingerprint" for a block of data. The actual number, or fingerprint, that is used to identify the data is called a **checksum**.

How to use it?

Once the file has been uploaded, the client issues the **XCRC <filename>** command to the server. The server will perform the **CRC-32** on the file and return the 4-byte "fingerprint" for the specified file. The CRC fingerprint will be returned in hexadecimal.

The client application can compare this fingerprint to the **CRC-32** fingerprint that it generates locally to determine if the file has been modified, corrupted, or altered during its transfer to the server.


For more information on the syntax and return values of the **XCRC** command, refer to the [XCRC command reference](#).

User Authentication Options

Titan FTP Server supports native **Titan** Authentication and **Windows NT/SAM** authentication.

- **Titan Authentication** - When using Titan authentication, the server administrator creates, manages, and deletes user accounts from within the Titan **Administrator**. The user accounts created in the Titan Administrator are used to access the Titan FTP server for which they are defined. These user accounts will not permit users to access other areas of your network.

- **Windows NT/SAM Authentication** - When using Windows NT/SAM authentication, the server administrator creates and deletes user accounts using the **Windows NT User Manager**, found in the Windows **Control Panel**. The Titan Administrator can then be configured to include one or more **NT Groups** from the **Windows SAM database**. All NT user accounts from the selected NT group or groups will then appear in the valid user list for Titan. This has the benefit of providing your NT users with a single username/password that they can use to access both the NT domain and the Titan FTP Server. When using Windows NT/SAM Authentication, Titan can be configured to access a local Windows workstation or a Windows Domain Controller.

 An icon consisting of two orange pillars on a silver base, used to denote a note or important information.	<ul style="list-style-type: none">• Windows NT/SAM support is available only in the Enterprise Edition of Titan FTP Server.• If you would like more information about configuring NT/SAM user authentication, see the Titan NT/SAM user authentication Quick Start Guide.
---	--

Remote Administration

Your Titan FTP Server Service can be configured to allow for **Remote Administration**. If **Remote Administration** is enabled, you can launch the Administrator program and connect to a Titan FTP Server Service over the Internet. To administer a Titan FTP Server remotely, launch the Titan **Administrator** and select **Administer Remote Domain** from the **File** menu. When prompted, you will need to type the **IP address** and **PORT** number that the remote Titan FTP Server is listening on.

Remote Administration provides most of the same functionality available during Local Administration with the exception of some directory traversal functions and certain security functions. If you attempt to administer a server remotely and the configuration options are disabled, it is most likely because you are not logged on locally.



Remote Administration Support is included in the Enterprise Edition of Titan FTP Server.

SSL Support

Titan FTP Server provides support for the industry standard **Secure Sockets Layer (SSL)**. When using **SSL** and an **SSL enabled** FTP client (such as [WebDrive](#)), data can be securely transferred over the Internet.

Titan FTP Server supports two methods of SSL enabling, **Implicit** and **Explicit** SSL.

Explicit SSL - Allows the client to initiate an SSL connection explicitly using the **AUTH SSL** command. See [RFC 2228](#) for more information.

Implicit SSL - Using **Implicit SSL**, Titan FTP Server will open a specific port that will only be used for SSL connections. By default, this is **port 990**; however, any port can be used.

Notes on SSL

Titan FTP Server does not support the **PBSZ** FTP command as defined in [RFC 2228](#). When an FTP client issues the **PBSZ** command, Titan FTP will return a **200 OK, PBSZ=0** message. See [PBSZ](#) for more information.

PROT P, **PROT S**, and **PROT E** are all treated as equal. Issuing any of these commands will result in the data channel being encrypted. To disable encryption of the data channel, issue the **PROT C** command. See [PROT](#) for more information.



- If you would like information about configuring Titan Server FTPS (FTPS/SSL) settings and public key certificate authentication, please see the Titan FTPS & Public Key Certificate Authentication [Quick Start Guide](#).
- FTPS/SSL features are only available in the Enterprise edition of Titan Server.

SFTP Support


Titan FTP Server provides support for **SFTP** (SSH's Secure File Transfer Protocol). When using **SFTP** and an **SFTP enabled** FTP client (such as [WebDrive](#)), data can be securely transferred over the Internet.

Titan FTP Server currently supports SFTP version 3 through SFTP version 6. Version 5 and 6 support is still in a preliminary stage since many clients do not yet support these versions. If you experience any problems with SFTP, try setting the version back to 3 or 4.

Host Key Support

Titan supports **SSH Host key authentication** for secure connections. **SSH Host keys** are similar to SSL Certificates in that there is a **public** key and a **private** key. The client keeps the **private** key secure on the client's local computer and distributes the **public** key so that it may be imported into the Titan system. Titan will associate this public key portion with a **user's** account and will use it to verify that the client is using the correct key pair.

Titan currently supports two host key algorithms, **RSA** and **DSA**. Predefined key lengths of **512**, **1024**, **2048**, **4096** are supported as well as **custom** key lengths. Smaller key lengths are slightly faster, but less secure. Larger/longer key lengths are stronger, but slower.

A small icon of a pencil with an orange eraser and a silver tip, positioned to the left of the text.

- If you would like more information about configuring Titan Server SFTP/SSH settings, please see the Titan SFTP/SSH & Host Key Management [Quick Start Guide](#).
- SFTP/SSH features are only available in the Enterprise edition of Titan Server.

Virtual Folders

Virtual Folders are folders that can be mapped into a server's **data directory** and are used to link or map external folders into a user's directory space. In a virtual folder it appears as if the data resides within folder structure; however, the data is actually stored somewhere else.

If you are a Windows user, you can think of a virtual folder as a Windows Shortcut. The link appears in one location and the data lives in another location. For UNIX users, virtual folders are very similar to Symbolic Links.

Virtual folders can be added at the **Server, Group, or User** level.

Group Level virtual folders allow data to be shared with all users of a given group. In a **Group Level** virtual folder, all users can share the same data and have **Directory Access Rights** to that data. Virtual Folders added at the **Group** level can be made accessible to all users in the group, depending on the **Directory Access Permissions** that are set for that group.

Virtual folders added at the **User Level** are limited to that specific user. When you add a virtual folder to a Titan server configuration, the default **Directory Access Permissions** will be set to **Read Only**. **Read Only** permissions means that users are allowed to browse the folder, and download information, but cannot modify the contents or upload files. You can modify the standard **Directory Access Permissions** after the virtual folder has been added to the configuration.

UNC Support

One of the benefits of Virtual Folders is that you can **access network shares** from the Titan server using of Virtual Folders. Titan supports the ability to add a **UNC** (Universal Naming Convention) path into the name space.


For example, if you have a share on your network called `\\MyServer\My Music\` you can use Virtual Folder support to map that into your Server Data Directory as `/pub/My Music/` or `/usr/joe/My Music/`

If you attempt to create a virtual folder for a mapped network drive, Titan will replace the drive mapping with the actual UNC name. This is because the Titan Service does not have access to mapped drives, only to UNC shares. Titan Server runs as a Windows Service that, by default, does not have access to shared network resources because shared network resources are based on the authorized Windows user. If you are mapping a UNC share, you must make sure that the account under which the Titan Service is running has access to the UNC. Otherwise, you will need to enter the appropriate username and password under the **UNC Accounts** tab.

Note: Titan FTP Server runs as a Windows Service that, by default, does not have access to shared network resources since these are based on the currently authorized Windows user. If you plan to use Virtual Folders, or if you plan to set your Server Data Directory to be a network resource, you should specify one or more user accounts to authenticate with on the server's Accounts tab.

Default Permissions for Virtual Folders

When you add a virtual folder to a Titan FTP Server configuration, the default **Directory Access Permissions** will be set to **READ ONLY**. Users will be able to browse the folder, and download information, but will not be able to modify the contents or upload files. You can modify the standard **Directory Access Permissions** once the Virtual Folder has been added to the configuration.

A red pushpin icon is located on the left side of the callout box, indicating a note or important information.

- If you attempt to create a Virtual Folder to a mapped network drive, Titan will replace the drive mapping with the actual UNC name. This is done because the Titan Service does not have access to mapped drives, only to UNC shares.
- If you would like more information about configuring group level virtual folders, please see the Titan Using Group Level Virtual Folders [Quick Start Guide](#).

srxCfg Command Line Utility

The **srxCfg** utility is a program that allows administrators to configure the Titan FTP server from a command prompt. The basic syntax is as follows:

```
srxCfg.exe /ADMINUSER=<adminusername> /ADMINPASS=<adminpass>  
/ADMINHOST=<machinename_or_ip> /ADMINPORT=<adminport>  
/CMD=<command> /SERVER=<ftpservername> [/OUTFILE=<filename.ext>]  
[/CMDFILE=<filename.ext>] [/attr=<value>]
```

Required Parameters

The following parameters are required by the **srxCfg** utility:

/ADMINUSER - Specifies the **Administrator User Name** used to connect to the Titan Service. This is the same user name that you supply when you log in through the Titan Administrator.

/ADMINPASS - Specifies the **Administrator Password** used to connect to the Titan Service. This is the same password that you supply when you log in through the Titan Administrator.

/ADMINHOST - Specifies the **computer name** or **IP address** of the computer on which the Titan Service is running. In most cases, Titan will be running on the local computer so you can use **localhost** or **127.0.0.1** for this parameter.

/ADMINPORT - Specifies the **Port** that the Titan Service is listening for administration commands. By default, Titan will listen on **port 31000** for local administration and **port 31001** for remote administration.

/CMD - Specifies the command to be executed against the Titan Service. The following list of commands is currently valid:

Server Based Commands

ADDSERVER - Creates a new server.

DELSERVER - Deletes an entire FTP server configuration from the system.

GETSATTR - Retrieves one or more FTP server attributes.

SETSATTR - Sets one or more FTP server attributes.

LISTSERVERS - Generates a list of FTP servers that are currently defined.

STARTSERVER - Starts an FTP server.

STOPSERVER - Stops an FTP server.

RESTARTSERVER - Restarts an FTP server.

Group Based Commands

ADDGROUP - Adds/Creates a new group for the specified server.

DELGROUP - Deletes an existing group from the specified server.

GETGATTR - Retrieves one or more attributes for the specified group.

SETGATTR - Sets one or more attributes for the specified Group.

LISTGROUPS - Returns a list of existing groups for the specified server.

User Based Commands

- ADDUSER** - Adds/Creates a new user for the specified Server.
- DELUSER** - Deletes an existing user from the specified Server.
- GETUATTR** - Retrieves one or more attributes for the specified user.
- SETUATTR** - Sets one or more attributes for the specified user.
- LISTUSERS** - Returns a list of existing Users for the specified server.
- KICKUSER** - Disconnects the user from the system. All sessions for this user are kicked.
- KICKSESS** - Disconnects the specified user session from the system.
- BANUSER** - Disables the users account and kicks the user from the system.

/SERVER - Specifies the name of the FTP Server to execute the command against. If the FTP server name is a long name or contains spaces, use double-quotes around the name. For Example: **/SERVER="My Ftp Server"**, or **/server=server1**, or **/SERVER="server1"**.

Conditional Parameters

The following parameters are **conditional**. These parameters must either be specified on the **command line**, or they must be supplied in a **Command File (/CMDFILE=)**.

/CMDFILE - Allows you to specify a file that contains the various attributes to **GET/SET** for the operation. If the attributes are not specified in a command file, they must be specified on the command line. If a command file is specified, command line attributes are ignored.

/ATTR - specifies the attribute for the command. For example, to retrieve server description (**ServerDesc**) attribute from the server named **Server1**:

```
... /CMD=GETSATTR /SERVER="Server1" /ServerDesc
```

This will return the **ServerDesc** attribute. To set the **ServerDesc**:

```
... /CMD=SETSATTR /SERVER="Server1" /ServerDesc="My New
Server Description"
```

Optional Parameters

The following parameters are optional:

/OUTFILE - Allows you to specify a file for output. This is useful for **GET*ATTR** calls so that a command file can be generated from the output.

Server Configuration Attributes

The following is a list of **configuration attributes** for Servers. These values can be retrieved and set using the **srxCfg Utility** or **srxCOM Interface**.

A general note about **message attributes**: These are messages that will be returned to the user/ftp client. Messages can be one or more lines and should be approximately 60 characters per line. If you want to have a multi-line message, separate each message with a '|'.

For example:

AcctDisabledMsg="Hello, Your Account Has Been Disabled.|Please contact your system administrator to have it re-opened.|GoodBye!".

List of Shared Attributes -- these attributes are shared at the **Server, Group, and User** level.

List of Server-Specific Attributes

Attribute	Description
AllowAnonymous	Allow anonymous access flag. 0 - No, 1 - Yes.
AuthDomainName	Name of the Windows Domain to use during NT Authentication. Ignored unless AuthType is set to NT User Authentication.
AuthPCName	Name of the computer that contains the NT User Database used during NT Authentication. This value is ignored unless AuthType is set to NT User Authentication
AuthType	Type of User Authentication database to use. 0 - Titan User Authentication, 1 - NT User Authentication.
AutoAssignDirs	Auto assign home directories for new users. If this value is enabled, each new user will have a directory under /usr/ . For example, /usr/user1/ , /usr/user2/ . 0 - Disabled, 1 - Enabled
BannerMsg	Message returned to the FTP client upon initial connection to the server. For multi-line messages, separate each line with a ' '.
BaseDataDir	The fully qualified path that refers to the base data directory where all of the server data files will be stored.
BaseLogDir	The fully qualified path to store the FTP Server log files.
CheckAnonymousPWD	Perform basic syntax checking on anonymous passwords to see if they are a valid email address. 0 - Disabled, 1 - Enabled.
DenyAccessByDefault	If enabled, the default setting is to deny access to all IP addresses except those listed in the IPAccessList attribute. Ignored unless EnableIPAccess is enabled. 0 - Disabled, 1 - Enabled.

FileCase	<p>Determines how to modify the file name during uploads, appends, and renames.</p> <p>0 - Preserve Case 1 - Covert to Lowercase 2 - Convert to Uppercase.</p>
FolderCase	<p>Determines how to modify the folder name when creating or renaming a directory.</p> <p>0 - Preserve Case 1 - Covert to Lowercase 2 - Convert to Uppercase.</p>
FTPEnabled	<p>Whether or not FTP access is permitted on the server.</p> <p>0 - FTP disabled, 1 - FTP enabled.</p>
GoingOfflineMsg	<p>Message displayed to the user when the server is going off-line. For multi-line messages, separate each line with a ' '.</p>
Host	<p>IP address that the FTP server is listening on. This will be either an actual IP address or 0.0.0.0, which means Listen on all available addresses.</p>
LogFormat	<p>Defines the format of the log file.</p> <p>0 - text format 1 - W3C format</p>
LogFieldsText	<p>Defines the fields written to the log file if LogFormat is text format. Separate each field with a ' '.</p> <p>Date Time ServerId Socket# Message</p>
LogFieldsW3C	<p>Defines the fields written to the log file if LogFormat is W3C format. Separate each field with a ' '.</p> <p>date time s-ip s-port x-socket c-ip c-port x-csocket cs-username cs-bytes sc-bytes cs-uri-stem</p>
LogLevel	<p>Controls the detail level of log information written out to both the screen and the log file.</p> <p>0 - HIGH, 1 -MEDIUM, 2 - LOW, 3 - DEBUG.</p>
LogToFile	<p>Controls the logging of data to the log file.</p> <p>0 - Disabled, 1 - Enabled.</p>

LogRotation	Defines the rotation schedule for the log files. 0 - No rotation 1 - Rotate Daily 2 - Rotate Weekly 3 - Rotate Monthly
LogToScreen	Controls the logging of data to the Administrators Activity Window. 0 - Disabled, 1 - Enabled.
LogWrap	Controls whether or not log entries are wrapped if they exceed LogWrapLength size. 0 - Disabled, 1 - Enabled.
LogWrapLength	The length at which to wrap log entries. A wrapped entry will appear as a second line in the logfile .
ModeZ	1 - Enable MODE Z support for this server. 0 - Disable MODE Z support for this server.
ModeZLevel	Sets the compression level to be used. The range is 0 through 9 , inclusive, with 0 being minimal or no compression and 9 being maximum compression . A higher compression level could produce slower performance due to the time it takes to compress the data.
NoAnonMsg	Message to display to a user who attempts to connect anonymously if anonymous access is not allowed. For multi-line messages, separate each line with a ' '.
OfflineMsg	Message to display to users who attempt to connect to the server when it is off-line. For multi-line messages, separate each line with a ' '.
Port	Specifies the port number that the FTP Server is listening on. The default value is port 21 .
RouterHost	If UseRouterHost is enabled, this value is the IP address of the router/cable-modem/firewall that is sitting in front of the FTP Server. In response to a PASV command, the server will return this IP address as the local IP address instead of the IP address of the server itself. This will allow the FTP client to open a data connection back to the router (which, in turn, will forward that request back to the server).
RunAtStartup	Controls the startup state of the server. 0 - No, do not start this FTP Server when the Titan Service starts. 1 - Yes, start this FTP Server when the Titan Service starts.
ServerDesc	Text description for the Server.
ServerName	Text name for the server.
SFTPEnabled	1 - Enables SFTP support on this server. 0 - Disables SFTP support on this server.
SFTPPort	Used to identify the port used for SFTP. The default port is port 22 .
SFTPVer	Identifies the maximum SFTP Protocol version supported by Titan. Titan currently supports SFTP v3, v4, v5, and v6.
SFTPHostKeyFolder	The fully qualified path location where Titan will store the user and server host keys.

SFTPRequireHK	<p>1 - Host keys are required from clients who attempt to connect using SFTP. When this feature is enabled, the Titan Administrator will need to import each user's public host key into the Titan Host Key Folder. If a user connects and supplies a host key that is not stored in Titan, the user will not be able to gain access.</p> <p>0 - Host keys are not required from clients who attempt to connect using SFTP.</p>
SFTPHostKeyName	The name of the host key file, stored in the Host Key Folder, used by Titan during SFTP handshaking.
SFTPUseCompression	1 - Titan will use Zlib compression if also supported by the client.
SFTPCipherList	<p>A list of encryption ciphers, in order, that Titan will present to the SFTP client during the handshaking/negotiation phase of the connection. The following ciphers are currently supported:</p> <p>3des-cbc blowfish-cbc aes128-cbc aes192-cbc aes256-cbc aes128-ctr aes192-ctr aes256-ctr cast128-cbc arcfour rijndael-cbc@lysator.liu.se</p>
SFTPMACList	<p>A list of MAC algorithms, in order, that Titan will present to the SFTP client during the handshaking/negotiation phase of the connection. The following MAC algorithms are currently supported:</p> <p>md5 sha1 ripemd160 ripemd160@openssh.com sha1-96 md5-96</p>
SSLBannerMsg	Message to display to the user when the user connects to the server over the SSLImplicitPort . For multi-line messages, separate each line with a ' '.
SSLCertStoreFolder	Text location of the SSL certificate store.
SSLCanCCC	<p>1 - Allow support for CCC (Clear Control Channel). This command will allow a secured SSL session to return to a non-secured state.</p> <p>0 - Disallow this command.</p>
SSLDisabledMsg	Message to display to the user if they attempt to initiate an SSL connection and SSLEnabled is disabled. For multi-line messages, separate each line with a ' '.
SSLExplicit	<p>Controls if AUTH SSL is enabled on the Server. This value is ignored if SSLEnabled is disabled.</p> <p>0 - AUTH SSL is not permitted, 1 - AUTH SSL is permitted.</p>
SSLImplicit	<p>Controls if Implicit SSL is enabled on the Server. This value is ignored if SSLEnabled is disabled.</p> <p>0 - Implicit SSL is not allowed, 1 - Implicit SSL is allowed.</p>

SSLImplicitPort	If SSLEnabled and SSLImplicit are enabled, this value represents the port number to be used for Implicit SSL connections. Implicit SSL connections connect to the server and immediately perform the SSL handshaking instead of issuing an AUTH SSL command.
SSLProtoVersion	The SSL protocol version number. 0 - SSL 3.0, 1 - SSL 3.1 (TLS 1.0)
SSLRequireCerts	1 - Require certificates from users who attempt to connect using SSL. 0 - Do not require certificates from users who connect using SSL.
StatMsg	Message to be displayed to the user in response to the STAT command. For multi-line messages, separate each line with a ' '.
StatsArchive	Controls the Archive stats before pruning value. 0 - Do not archive prior to prune, 1 - Archive before prune.
StatsArchiveDate	Contains date of the last archive for the stats database.
StatsDSN	The ODBC data source name identifying the statistics database
StatsEnabled	Controls the setting for Statistics. 0 - Disable statistics logging, 1 - Enable Statistics logging.
StatsMask	A mask defining the information to be recorded. The string is a series of 0 or 1 indicating which statistic is recorded. The following positions, left to right, are defined: <ul style="list-style-type: none"> 0 - File Uploads 1 - File Downloads 2 - User Login Attempts 3 - User Logout Attempts 4 - Client Connection Attempts 5 - Client Disconnects 6 - Directory Listings 7 - Delete Files 8 - Delete Folders 9 - Create Folders <p>For example, if the StatsMask were set to 0100001000, Titan would record statistics for File Downloads and Directory Listings.</p>
StatsRotation	Defines the archive/pruning schedule for the statistics. <ul style="list-style-type: none"> 0 - No pruning (not recommended) 1 - Prune Daily 2 - Prune Weekly 3 - Prune Monthly 4 - Prune Annually
STOUPrefix	Prefix to be used during the generation of a unique file name used in the STOU command.
TimeZoneAdjust	Controls the setting for whether or not to make adjustments for time zone when returning the directory information to the client.
TimeZoneMinutes	If TimeZoneAdjust is enabled, this value indicates the number of minutes to add/subtract from the GMT time for file times.
UNCPassword	Password used for UNC access. For security reasons, this attribute can only be set, not retrieved.

UNCUser	User name used for UNC access.
UseRouterHost	Controls the setting for whether or not this server is sitting behind a router/cable modem/firewall. This setting is used in conjunction with the RouterHost value and the PASV command to return a valid local IP address to the remote FTP client. 0 - Disabled, 1 - Enabled.
UseStdUnixDirs	Controls the creation of standard Unix FTP style directories. If enabled, the Server will create /incoming/ , /usr/ , /bin/ , /pub/ folders under the BaseDataDir .
UseNTHomeDir	Use the NT home directory for a user. 0 - Disabled, 1 - Enabled.
UseNTImpersonation	Use NT Impersonation. 0 - Disabled, 1 - Enabled.
UTF8Support	Controls the UTF8 feature for the server. 0 - UTF8 Disabled, 1 - UTF8 Enabled.
WelcomeMsg	Message to display to a user once they have successfully logged in to the server. For multi-line messages, separate each line with a ' '.

ADDSERVER

ADDSERVER creates a new FTP Server. **NOTE:** It is advised that you specify the **HOST IP and PORT Attributes** for the new server. If the **RunAtStartup** attribute is enabled, the server will be started as soon as it is created.

Syntax:

```
srxCfg /CMD=ADDSERVER /SERVER=<ftpservername>  
[[/CMDFILE=<filename.ext>] or [/attr=<value>]]
```

/SERVER - Specifies the name of the FTP server to execute the command against. If the FTP server name is a long name and it contains spaces, use double-quotes around the name. For example: **/server="My Ftp Server"** or **/server=server1** or **/server="server1"**.

/CMDFILE - Allows you to specify a file that contains the various **server attributes** for the new FTP Server. If the server attributes are not specified in a command file, the attributes must be specified on the command line. If a command file is specified, command line attributes are ignored.

/ATTR - specifies one or more server attributes to modify.

Examples:

```
srxCfg /CMD=addserver /SERVER="Server1" /ServerDesc="My Server  
Description" /Host=1.2.3.4 /Port=21 /RunAtStartup=0
```

Creates a new server on **IP 1.2.3.4, Port 21**. Does **not** start the server

```
srxCfg /CMD=addserver /server="Server1" /cmdfile=server_in.ini  
/outfile=server_out.ini  
; ***** server_in.ini file contents *****  
[SERVER:server1]  
Host=1.2.3.4  
Port=21  
RunAtStartup=1  
; ***** end of server_in.ini file
```

Creates a new server on **IP 1.2.3.4, Port 21**. Starts the server.

DELSERVER

DELSERVER deletes an FTP server configuration from the system. All group and user information for this FTP Server is also deleted. **Caution:** This command is permanent and cannot be undone.

Syntax:

```
srxCfg /CMD=DELSERVER /SERVER=<ftpservername>
```

/SERVER - Specifies the name of the FTP Server to execute the command against. If the FTP server name is a long name and it contains spaces, use double-quotes around the name. For Example: `/server="My Ftp Server"` or `/server=server1` or `/server="server1"`.

Examples:

```
srxCfg /cmd=delservice /server="Server1"
```

Completely removes **Server1** and **all Groups/Users under it**.



CAUTION: DELSERVER is permanent and cannot be undone.

GETSATTR

GETSATTR retrieves one or more **server attributes** from the specified Server.

Syntax:

```
srxCfg /CMD=GETSATTR /SERVER=<ftpservername>  
[[/CMDFILE=<filename.ext>] or [/attr=<value>]]
```

/SERVER - Specifies the **name** of the FTP Server to execute the command against. If the FTP Server Name is a long name and it contains spaces, use double-quotes around the name. For Example: **/server="My Ftp Server"** or **/server=server1** or **/server="server1"**.

/CMDFILE - Allows you to specify a **file** that contains the various **server attributes** to retrieve. If the server attributes are not specified in a command file, they must be specified on the command line. If a command file is specified, command line attributes are ignored.

/attr - specifies the **server attribute** to retrieve. If no attribute is supplied, all of the server attributes will be retrieved.

Examples:

```
srxCfg /cmd=getsattr /server="Server1" /ServerDesc
```

Retrieves the **Server Description**

```
srxCfg /cmd=getsattr /server="Server1" /ServerName /Host /Port /ServerDesc
```

Retrieves the **Servers Name, Description, Host IP and Port**. The information will be dumped out to the **console/stdout**.

```
srxCfg /cmd=getsattr /server="Server1" /ServerName /outfile=serverinfo.ini
```

Retrieves the **server name** and outputs the information to **serverinfo.ini**

```
srxCfg /cmd=getsattr /server="Server1" /cmdfile=server_in.ini  
/outfile=server_out.ini  
; ***** server_in.ini file contents *****  
[SERVER:server1]  
ServerName=  
ServerDesc=  
; ***** end of server_in.ini file
```

Retrieves the **server name** and **server description**. Writes the output to **server_out.ini**

SETSATTR

SETSATTR sets one or more **server attributes** for the specified Server.

Syntax:

```
srxCfg /CMD=SETSATTR /SERVER=<ftpservername>  
[[/CMDFILE=<filename.ext>] or [/attr=<value>]]
```

/SERVER - Specifies the name of the FTP Server to execute the command against. If the FTP server name is a long name and it contains spaces, use double-quotes around the name. For Example: **/server="My Ftp Server"** or **/server=server1** or **/server="server1"**.

/CMDFILE - Allows you to specify a file that contains the various **server attributes** to SET. If the server attributes are not specified in a command file, they must be specified on the command line. If a command file is specified, command line Attributes are ignored.

/attr - specifies one or more **server attributes** to modify.

Examples:

```
srxCfg /cmd=setsattr /server="Server1" /ServerDesc="My Server Description"
```

Modifies the Server Description to be My Server Description

```
srxCfg /cmd=setsattr /server="Server1" /Host=1.2.3.4 /Port=44
```

Modifies the **IP/Port** for the server to be **1.2.3.4, Port 44**. You will need to **restart** the FTP server to have it use the new IP/Port.

```
srxCfg /cmd=setsattr /server="Server1" /cmdfile=server_in.ini  
/outfile=server_out.ini  
; ***** server_in.ini file contents *****  
[SERVER:server1]  
ServerDesc=My New Server Description  
; ***** end of server_in.ini file
```

Modifies the **server description**. Writes the output to **server_out.ini**

LISTSERVERS

LISTSERVERS returns a list of FTP servers currently defined by the system. The list will contain the internal **ServerID** and the **server name**.

Syntax:

```
srxCfg /CMD=LISTSERVERS
```

Examples:

```
srxCfg /cmd=listservers
```

```
200-Server list
2:Server1
4:My Other Server
5:FTP Test Server
200 Command Complete
```

STARTSERVER

STARTSERVER starts an FTP Server.

Syntax:

```
srxCfg /CMD=STARTSERVER /SERVER=<ftpservername>
```

/SERVER - Specifies the **name** of the FTP Server to execute the command against. If the FTP server name is a long name and it contains spaces, use double-quotes around the name. For Example: **/server="My Ftp Server"** or **/server=server1** or **/server="server1"**.

Examples:

```
srxCfg /cmd=startserver /server="Server1"
```

Issues the Start command to Server1.

STOPSERVER

STOPSERVER stops an FTP Server. All users are immediately kicked off the server and the server is shutdown.

Syntax:

```
srxCfg /CMD=STOPSERVER /SERVER=<ftpservername>
```

/SERVER - Specifies the name of the FTP Server to execute the command against. If the FTP server name is a long name and it contains spaces, use double-quotes around the name. For Example: **/server="My Ftp Server"** or **/server=server1** or **/server="server1"**.

Examples:

```
srxCfg /cmd=stopserver /server="Server1"
```

Issues the **Stop** command to **Server1**.

RESTARTSERVER

RESTARTSERVER will **stop**, then **restart** an FTP server. All users are immediately kicked off the server and the server is shutdown before being restarted.

Syntax:

```
srxCfg /CMD=RESTARTSERVER /SERVER=<ftpservername>
```

/SERVER - Specifies the name of the FTP server to execute the command against. If the FTP Server Name is a long name and it contains spaces, use double-quotes around the name. For Example: **/server="My Ftp Server"** or **/server=server1** or **/server="server1"**.

Examples:

```
srxCfg /cmd=restartserver /server="Server1"
```

Issues the **Restart** command to **Server1**.

Group Configuration Attributes

The following is a list of **configuration attributes** for **groups**. These values can be retrieved and set using the **srxCfg Utility** or **srxCOM Interface**.

A general note about Message attributes. These messages are returned to the user/ftp client. Messages can be one or more lines and should be kept to about 60 characters per line. If you want to have a multi-line message, separate each message with a '|'. For example:
AcctDisabledMsg="Hello, Your Account Has Been Disabled.|Please contact your system administrator to have it re-opened.|GoodBye!".

List of Shared Attributes -- these attributes are shared at the **Server, Group, and User level**.

List of Group-Specific Attributes

Attribute	Description
AlwaysAllowLogin	Always allow members of this group to login, even if the maximum number of connections for the server has been reached. 0 - Disabled, 1 - Enabled.
Enabled	Enable or disable this group. If a group is disabled, any users that inherit the Enabled value from the group will also be disabled. 0 - Disabled, 1 - Enabled.
ExpirationDateOn	Enable or disable a group expiration date. 0 - Disabled, 1 - Enabled.
ExpirationDate	The expiration date for the group.
GroupId	Internal ID for the group. Use this ID in the Users Membership attribute.
GroupName	Text name for the group.
HomeDir	Home directory for the group. If HomeDirEnabled is set, users that are a member of this group will have a home directory based on the group home directory.
HomeDirEnabled	Enable or disable the concept of a Group home directory. 0 - Disabled, no group home directory 1 - Enabled, user home directory is set to Group home directory. 2 - Enabled, user home directory is set to Group home directory plus username. Example: if username=bob and group home dir="C:\group\" , then user home dir="C:\group\bob\"
Members	A ' ' separated list of user names of the users who are members of this group. Example: user1 user22 user44

ADDGROUP

ADDGROUP Creates a new group for the specified FTP server.

Syntax:

```
srxCfg /CMD=ADDGROUP /SERVER=<ftpservername> /GROUP=<groupname>
[[/CMDFILE=<filename.ext>] or [/attr=<value>]]
```

/SERVER - Specifies the name of the FTP server to execute the command against. If the FTP server name is a long name and it contains spaces, use double-quotes around the name. For Example: **/server="My Ftp Server"** or **/server=server1** or **/server="server1"**.

/GROUP - Specifies the new groups name.

/CMDFILE - Allows you to specify a file that contains the various [group attributes](#) for the new group. If the group attributes are not specified in a command file, they must be specified on the command line. If a command file is specified, command line attributes are ignored.

/attr - specifies one or more [group attributes](#) to set for the new group.

Examples:

```
srxCfg /cmd=addgroup /server="Server1" /group=Engineering
/Members=|1|2|3|
```

Creates a **new group** named **Engineering** and **adds users 1, 2 and 3** to the group.

```
srxCfg /cmd=addgroup /server="Server1" /cmdfile=group_in.ini
/outfile=group_out.ini
; ***** group_in.ini file contents *****
[GROUP:Engineering]
groupname=Engineering
Members=|1|2|3|
; ***** end of group_in.ini file
```

Creates a **new group** named **Engineering** and **adds users |1|2|3|** to the group.

DELGROUP

DELGROUP deletes a group from the server. **Note:** you cannot delete the **Everyone** group.

Syntax:

srxCfg /CMD=DELGROUP /SERVER=<ftpservername> /GROUP=Engineering

/SERVER - Specifies the name of the FTP Server to execute the command against. If the FTP Server Name is a long name and it contains spaces, use double-quotes around the name. For Example: **/server="My Ftp Server"** or **/server=server1** or **/server="server1"**.

/GROUP - Specifies the group to delete.

Example:

srxCfg /cmd=delgroup /server="Server1" /group=Engineering

Deletes the group named **Engineering**.

GETGATTR

GETGATTR retrieves one or more **group attributes** from the specified FTP Server and Group.

Syntax:

```
srxCfg /CMD=GETGATTR /SERVER=<ftpservername> /GROUP=<groupname>
[[/CMDFILE=<filename.ext>] or [/attr=<value>]]
```

/SERVER - Specifies the name of the FTP Server to execute the command against. If the FTP Server Name is a long name and it contains spaces, use double-quotes around the name. For Example: **/server="My Ftp Server"** or **/server=server1** or **/server="server1"**.

/GROUP - Specifies the group name on the FTP server.

/CMDFILE - Allows you to specify a file that contains the various **group attributes** to retrieve. If the Attributes are not specified in a command file, they must be specified on the command line. If a command file is specified, command line Attributes are ignored.

/attr - specifies the **group attribute** to retrieve. If no attribute is supplied, all of the groups attributes will be retrieved.

Examples:

```
srxCfg /cmd=getgattr /server="Server1" /group=Everyone /Members
```

Retrieves a list of **members** of the group

```
srxCfg /cmd=getgattr /server="Server1" /group=Everyone /cmdfile=group_in.ini
/outfile=group_out.ini
```

```
; ***** group_in.ini file contents *****
```

```
[GROUP:Everyone]
```

```
Members=
```

```
VirtFolders=
```

```
; ***** end of group_in.ini file
```

Retrieves a list of **members** of the group and a list of **Virtual Folders** defined for the group. Writes the output to **group_out.ini**

SETGATTR

SETGATTR sets one or more **group attributes** for the specified Server/User.

Syntax:

```
srxCfg /CMD=SETGATTR /SERVER=<ftpservername> /GROUP=<groupname>
[[/CMDFILE=<filename.ext>] or [/attr=<value>]]
```

/SERVER - Specifies the name of the FTP Server to execute the command against. If the FTP Server Name is a long name, i.e. it contains spaces, use double-quotes around the name. For Example: /server="My Ftp Server" or /server=server1 or /server="server1".

/GROUP - Specifies the group's name.

/CMDFILE - Allows you to specify a file that contains the various **Group Attributes** to SET. If the **Group Attributes** are not specified in a command file, they must be specified on the command line. If a command file is specified, command line Attributes are ignored.

/attr - specifies one or more **Group Attributes** to modify.

Examples:

```
srxCfg /cmd=setgattr /server="Server1" /Group=Everyone /Members=|1|4|5|
```

Adds users 1, 4 and 5 as members of the group

```
srxCfg /cmd=setgattr /server="Server1" /Group=Everyone /cmdfile=group_in.ini
/outfile=group_out.ini
```

```
; ***** group_in.ini file contents *****
```

```
[GROUP:Everyone]
```

```
Members=|1|4|5|
```

```
; ***** end of group_in.ini file
```

Adds users 1, 4 and 5 as members of the group. Writes the output to group_out.ini

LISTGROUPS

LISTGROUPS returns a list of groups currently defined by the supplied FTP Server. The list will contain the internal **GroupId** and the Group account name. The **GroupId** is useful for setting the User's **Membership** attribute.

Syntax:

```
srxCfg /CMD=LISTGROUPS /SERVER=<servername>
```

Examples:

```
srxCfg /cmd=listgroups /server="My Server"
```

200-Group list

1:Everyone

2:Sales

3:Engineering

4:QA

200 Command Complete

User Configuration Attributes

The following is a list of configuration attributes for Users. These values can be retrieved and set using the [srxCfg Utility](#) or [srxCOM Interface](#).

A general note about message attributes. These messages are returned to the user/ftp client. Messages can be one or more lines and should be kept to about 60 characters per line. If you want to have a multi-line message, separate each message with a '|'. For example:
AcctDisabledMsg="Hello, Your Account Has Been Disabled.|Please contact your system administrator to have it re-opened.|GoodBye!".

List of Shared Attributes -- these attributes are shared at the **Server, Group, and User level**.

List of User-Specific Attributes

Attribute	Description
AlwaysAllowLogin	Always allow this user to log in, even if the maximum number of connections for the server has been reached. 0 - Disabled, 1 - Enabled.
EmailAddress	Email address for the user.
Enabled	Enable or disable this user. 0 - Disabled, 1 - Enabled.
ExpirationDateOn	Enable or disable a group expiration date. 0 - Disabled, 1 - Enabled.
ExpirationDate	The expiration date for the group.
FullName	User's full name.
HomeDirInherit	Enable or disable to determine if the Home Directory value should be taken from the group or user attribute. 0 - Disabled, user's home directory will be user's own home directory settings; user will not inherit home directory from the group 1 - Enabled, user will inherit home directory from the group level
Membership	A ' ' separated list of group names of the groups to which this user belongs. Example: group1 group22 group44
Password	Password for this user account. For security reasons, this attribute can only be set, not retrieved.
PwdType	Password Encryption Type. 0 - Standard Password 1 - Anything 2 - OTP S/Key MD4 3 - OTP S/Key MD5
UserId	Internal ID for the user.
Username	Username for the user's account.

ADDUSER

ADDUSER creates a new user Account for the specified FTP Server. Upon successful creation, the account is immediately available for use, provided that the **Enabled** attribute is set.

Syntax:

```
srxCfg /CMD=ADDUSER /SERVER=<ftpservername> /user=<username>  
[[/CMDFILE=<filename.ext>] or [/attr=<value>]]
```

/SERVER - Specifies the name of the FTP Server to execute the command against. If the FTP Server Name is a long name and it contains spaces, use double-quotes around the name. For Example: **/server="My Ftp Server"** or **/server=server1** or **/server="server1"**.

/USER - Specifies the new user's user name.

/CMDFILE - Allows you to specify a file that contains the various [user attributes](#) for the new user. If the user attributes are not specified in a command file, they must be specified on the command line. If a command file is specified, command line attributes are ignored.

/attr - specifies one or more [user attributes](#) to set for the new user.

Examples:

```
srxCfg /cmd=adduser /server="Server1" /user=user1 /password=user1password
```

Creates a new user on Server1 with a username of **user1** and a password of **user1password**. The account is initially enabled.

```
srxCfg /cmd=adduser /server="Server1" /cmdfile=user_in.ini  
/outfile=user_out.ini  
; ***** user_in.ini file contents *****  
[USER:user1]  
username=user1  
password=user1password  
enabled=0  
; ***** end of user_in.ini file
```

Creates a new user named **user1** with a password of **user1password**. The account is initially disabled.

DELUSER

DELUSER deletes a user account from the server. If the user is currently logged in to the server, they are disconnected. **Note:** you cannot delete the **Anonymous** account.

Syntax:

```
srxCfg /CMD=DELUSER /SERVER=<ftpservername> /USER=<username>
```

/SERVER - Specifies the name of the FTP Server to execute the command against. If the FTP server name is a long name and it contains spaces, use double-quotes around the name. For Example: **/server="My Ftp Server"** or **/server=server1** or **/server="server1"**.

/USER - Specifies the user account to delete.

Example:

```
srxCfg /cmd=deluser /server="Server1" /user=fred
```

Deletes the user account for **fred**.

GETUATTR

GETUATTR retrieves one or more [user attributes](#) from the specified FTP server and user.

Syntax:

```
srxCfg /CMD=GETUATTR /SERVER=<ftpservername> /USER=<username>
[[/CMDFILE=<filename.ext>] or [/attr=<value>]]
```

/SERVER - Specifies the name of the FTP Server to execute the command against. If the FTP Server Name is a long name and it contains spaces, use double-quotes around the name. For Example: **/server="My Ftp Server"** or **/server=server1** or **/server="server1"**.

/USER - Specifies the username of the user account on the FTP server.

/CMDFILE - Allows you to specify a file that contains the various [user attributes](#) to retrieve. If the attributes are not specified in a command file, they must be specified on the command line. If a command file is specified, command line attributes are ignored.

/attr - specifies the [user attribute](#) to retrieve. If no attribute is supplied, all of the attributes will be retrieved.

Examples:

```
srxCfg /cmd=getuattr /server="Server1" /user=Anonymous /FullName
```

Retrieves the user's full name.

```
srxCfg /cmd=getuattr /server="Server1" /user=Anonymous /UserName
/outfile=userinfo.ini
```

Retrieves the user's user name and outputs the information to **userinfo.ini**.

```
srxCfg /cmd=getuattr /server="Server1" /user=Anonymous /cmdfile=user_in.ini
/outfile=user_out.ini
```

```
; ***** user_in.ini file contents *****
[USER:Anonymous]
UserName=
FullName=
; ***** end of user_in.ini file
```

Retrieves the user's user name and the user's full name. Writes the output to **user_out.ini**

SETUATTR

SETUATTR sets one or more [user attributes](#) for the specified server/user.

Syntax:

```
srxCfg /CMD=SETUATTR /SERVER=<ftpservername> /USER=<username>  
[[/CMDFILE=<filename.ext>] or [/attr=<value>]]
```

/SERVER - Specifies the name of the FTP server to execute the command against. If the FTP server name is a long name and it contains spaces, use double-quotes around the name. For Example: **/server="My Ftp Server"** or **/server=server1** or **/server="server1"**.

/USER - Specifies the user's user name.

/CMDFILE - Allows you to specify a file that contains the various [user attributes](#) to SET. If the user attributes are not specified in a command file, they must be specified on the command line. If a command file is specified, command line attributes are ignored.

/attr - specifies one or more [user attributes](#) to modify.

Examples:

```
srxCfg /cmd=setuattr /server="Server1" /user=Anonymous /enabled=1
```

Enables the **Anonymous** account

```
srxCfg /cmd=setuattr /server="Server1" /user=Anonymous /AllowMDTM=0
```

Modifies the **Anonymous** account so that anonymous users cannot change file dates/times using the **MDTM** command.

```
srxCfg /cmd=setuattr /server="Server1" /User=Anonymous /cmdfile=user_in.ini  
/outfile=user_out.ini
```

```
; ***** user_in.ini file contents *****
```

```
[USER:Anonymous]
```

```
Enabled=0
```

```
; ***** end of user_in.ini file
```

Disables the **Anonymous** account. Writes the output to **user_out.ini**

LISTUSERS

LISTUSERS returns a list of user accounts currently defined by the supplied FTP server. The list will contain the internal **UserID** and the user account name (**username**).

Syntax:

```
srxCfg /CMD=LISTUSERS /SERVER=<servername>
```

Examples:

```
srxCfg /cmd=listusers /server="My Server"
```

```
200-User list
1:anonymous
2:fred
3:tom
4:admin
143:user12
200 Command Complete
```

BANUSER

BANUSER kicks all sessions for a particular user from the system; then bans the user by disabling their account. The user is not warned about the pending session termination; however, a message is sent out to the FTP client notifying them that the session has been terminated.

Syntax:

```
srxCfg /CMD=BANUSER /SERVER=<ftpservername> /USER=<username>
```

/SERVER - Specifies the name of the FTP Server to execute the command against. If the FTP Server Name is a long name and it contains spaces, use double-quotes around the name. For Example: **/server="My Ftp Server"** or **/server=server1** or **/server="server1"**.

/USER - Specifies the use name to terminate. All sessions for this user will be terminated and the user is banned from accessing the server. If there are no active user sessions, the user is still banned from logging on in the future.

Examples:

```
srxCfg /cmd=banuser /server="Server1" /user=fred
```

Terminates all instances connected to the server of user **fred**. Fred is then banned from accessing the server

```
srxCfg /cmd=banuser /server="Server1" /user=anonymous
```

Kicks all **Anonymous** users from the system, then disables **Anonymous** access.

KICKSESS

KICKSESS kick an individual user session from the system. Supply the users **SessionID**.

Syntax:

```
srxCfg /CMD=KICKSESS /SERVER=<ftpservername> /SESSION=<sessionid>
```

/SERVER - Specifies the name of the FTP Server to execute the command against. If the FTP Server Name is a long name and it contains spaces, use double-quotes around the name. For Example: **/server="My Ftp Server"** or **/server=server1** or **/server="server1"**.

/SESSION - Specifies the session ID to terminate. The Session ID can be obtained using the **Activity** window in the **Titan Server Administrator**.

Example:

```
srxCfg /cmd=kicksess /server="Server1" /session=1234
```

Terminates session **1234** from the server.

KICKUSER

KICKUSER kicks all sessions for a particular user from the system. The user is not warned about the pending session termination; however, a message is sent out to the FTP client notifying them that their session has been terminated.

Syntax:

```
srxCfg /CMD=KICKUSER /SERVER=<ftpservername> /USER=<username>
```

/SERVER - Specifies the name of the FTP Server to execute the command against. If the FTP Server Name is a long name and it contains spaces, use double-quotes around the name. For Example: **/server="My Ftp Server"** or **/server=server1** or **/server="server1"**.

/USER - Specifies the user name to terminate. All sessions for this user will be terminated.

Examples:

```
srxCfg /cmd=kickuser /server="Server1" /user=fred
```

Terminates all instances connected to the server of user **fred**.

```
srxCfg /cmd=kickuser /server="Server1" /user=anonymous
```

Kicks all **Anonymous** users from the system.

srxCOM Interface

srxCOM is a COM/Scripting interface that can be used to configure **Servers, Groups, and Users**. The scripting engine is installed with Titan FTP Server.

Interface Name: **srxCom.SRXTitan**

Implementation **DLL**: **srxCom.dll**

Methods:

SRX_Connect() - Opens a connection to the Titan Service.

SRX_Disconnect() - Closes an existing connection.

SRX_GetErrStr() - Retrieves an error string for the supplied error code.

SVR_Create() - Creates a new FTP Server instance.

SVR_Delete() - Deletes an existing FTP Server instance.

SVR_Enum() - Generates a list of FTP Servers.

SVR_Start() - Starts an FTP Server.

SVR_Stop() - Stops an FTP Server.

SVR_Restart() - Restarts an FTP Server.

SVR_GetAttr() - Retrieves a configuration attribute for an FTP Server.

SVR_SetAttr() - Sets/Modifies a configuration attribute for an FTP Server.

SVR_GetSessions - Generates a list of sessions for an FTP Server.

GRP_Create() - Creates a new group for the specified FTP Server.

GRP_Delete() - Deletes an existing group from the specified FTP Server.

GRP_Enum() - Generates a list of groups defined for the specified FTP Server.

GRP_GetMembers() - Generates a list of members/users for the specified FTP Server/Group.

GRP_SetMembers() - Changes the members list for the specified FTP Server/Group.

GRP_GetAttr() - Retrieves an attribute for the specified FTP Server/Group.

GRP_SetAttr() - Changes an attribute for the specified FTP Server/Group.

USR_Create() - Creates a new user for the specified FTP Server.

USR_Delete() - Deletes an existing user from the specified FTP Server.

USR_Enum() - Generates a list of users defined for the specified FTP Server.

USR_GetAttr() - Retrieves an attribute for the specified FTP Server/User.

USR_SetAttr() - Changes an attribute for the specified FTP Server/User.

VB 6 Example:

```
Dim srxcom
Set srxcom = CreateObject("srxCom.SRXTitan") ' instantiate the object
srxcom.SRX_Connect("localhost",31000,"Administrator","MyPassword")
srxcom.SVR_Create("fred", "1.2.3.4", 12, "C:\fred", 0)
srxcom.SRX_Disconnect
```

SRX_Connect

Opens a connection to the local Titan FTP server.

Syntax:

```
SRX_Connect( LPCTSTR szMachineIP, USHORT usPort, LPCTSTR szUsername,
LPCTSTR szPassword)
```

```
SRX_Connect2( LPCTSTR szMachineIP, USHORT usPort, LPCTSTR szUsername,
LPCTSTR szPassword, LONG* IError)
```

Parameters:

szMachineIP	Specifies the computer name or IP address of the PC on which the Titan Service is running. In most cases, Titan will be running on the local computer so you can use localhost or 127.0.0.1 for this parameter.
usPort	Specifies the Port that the Titan Service is listening for administration commands. By default, Titan will listen on port 31000 for local administration and port 31001 for remote administration.
szUsername	Specifies the administrator user name used to connect to the Titan Service. This is the same user name that is supplied when logging in through the Titan Server Administrator .
szPassword	Specifies the administrator password used to connect to the Titan Service. This is the same password that is supplied when logging in through the Titan Server Administrator .
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

SRX_Disconnect

Closes an existing connection to the Titan Service.

Syntax:

```
SRX_Disconnect( void )
```

```
SRX_Disconnect2( LONG* IError )
```

Parameters:

IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.
---------------	---

Return Values:

This method does not return a value.

SRX_GetErrStr

Retrieves an error string for the supplied error code.

Syntax:

```
void SRX_GetErrStr ( long IRetCode, BSTR* pszErrStr );
```

Parameters:

IRetCode	The error code (returned from one of the other methods) for which the error string will be retrieved.
pszErrStr	Pointer to retrieve the error string information

Return Values:

This method does not return a value.

Server Configuration Attributes

The following is a list of **configuration attributes** for Servers. These values can be retrieved and set using the **srxCfg Utility** or **srxCOM Interface**.

A general note about **message attributes**: These are messages that will be returned to the user/ftp client. Messages can be one or more lines and should be approximately 60 characters per line. If you want to have a multi-line message, separate each message with a '|':

For example:

AcctDisabledMsg="Hello, Your Account Has Been Disabled.|Please contact your system administrator to have it re-opened.|GoodBye!".

List of Shared Attributes -- these attributes are shared at the **Server, Group, and User** level.

List of Server-Specific Attributes

Attribute	Description
AllowAnonymous	Allow anonymous access flag. 0 - No, 1 - Yes.
AuthDomainName	Name of the Windows NT Domain to use during NT Authentication. Ignored unless AuthType is set to NT User Authentication.
AuthPCName	Name of the computer that contains the NT User Database used during NT Authentication. This value is ignored unless AuthType is set to NT User Authentication
AuthType	Type of User Authentication database to use. 0 - Titan User Authentication, 1 - NT User Authentication.
AutoAssignDirs	Auto assign home directories for new users. If this value is enabled, each new user will have a directory under /usr/ . For example, /usr/user1/ , /usr/user2/ . 0 - Disabled, 1 - Enabled
BannerMsg	Message returned to the FTP client upon initial connection to the server. For multi-line messages, separate each line with a ' ':
BaseDataDir	The fully qualified path that refers to the base data directory where all of the server data files will be stored.
BaseLogDir	The fully qualified path to store the FTP Server log files.
CheckAnonymousPWD	Perform basic syntax checking on anonymous passwords to see if they are a valid email address. 0 - Disabled, 1 - Enabled.
DenyAccessByDefault	If enabled, the default setting is to deny access to all IP addresses except those listed in the IPAccessList attribute. Ignored unless EnableIPAccess is enabled. 0 - Disabled, 1 - Enabled.

FileCase	<p>Determines how to modify the file name during uploads, appends, and renames.</p> <p>0 - Preserve Case 1 - Covert to Lowercase 2 - Convert to Uppercase.</p>
FolderCase	<p>Determines how to modify the folder name when creating or renaming a directory.</p> <p>0 - Preserve Case 1 - Covert to Lowercase 2 - Convert to Uppercase.</p>
FTPEnabled	<p>Whether or not FTP access is permitted on the server.</p> <p>0 - FTP disabled, 1 - FTP enabled.</p>
GoingOfflineMsg	<p>Message displayed to the user when the server is going off-line. For multi-line messages, separate each line with a ' '.</p>
Host	<p>IP address that the FTP server is listening on. This will be either an actual IP address or 0.0.0.0, which means Listen on all available addresses.</p>
LogFormat	<p>Defines the format of the log file.</p> <p>0 - text format 1 - W3C format</p>
LogFieldsText	<p>Defines the fields written to the log file if LogFormat is text format. Separate each field with a ' '.</p> <p>Date Time ServerId Socket# Message</p>
LogFieldsW3C	<p>Defines the fields written to the log file if LogFormat is W3C format. Separate each field with a ' '.</p> <p>date time s-ip s-port x-socket c-ip c-port x-csocket cs-username cs-bytes sc-bytes cs-uri-stem</p>
LogLevel	<p>Controls the detail level of log information written out to both the screen and the log file.</p> <p>0 - HIGH, 1 -MEDIUM, 2 - LOW, 3 - DEBUG.</p>
LogToFile	<p>Controls the logging of data to the log file.</p> <p>0 - Disabled, 1 - Enabled.</p>

LogRotation	Defines the rotation schedule for the log files. 0 - No rotation 1 - Rotate Daily 2 - Rotate Weekly 3 - Rotate Monthly
LogToScreen	Controls the logging of data to the Administrators Activity Window. 0 - Disabled, 1 - Enabled.
LogWrap	Controls whether or not log entries are wrapped if they exceed LogWrapLength size. 0 - Disabled, 1 - Enabled.
LogWrapLength	The length at which to wrap log entries. A wrapped entry will appear as a second line in the logfile .
ModeZ	1 - Enable MODE Z support for this server. 0 - Disable MODE Z support for this server.
ModeZLevel	Sets the compression level to be used. The range is 0 through 9 , inclusive, with 0 being minimal or no compression and 9 being maximum compression . A higher compression level could produce slower performance due to the time it takes to compress the data.
NoAnonMsg	Message to display to a user who attempts to connect anonymously if anonymous access is not allowed. For multi-line messages, separate each line with a ' '.
OfflineMsg	Message to display to users who attempt to connect to the server when it is off-line. For multi-line messages, separate each line with a ' '.
Port	Specifies the port number that the FTP Server is listening on. The default value is port 21 .
RouterHost	If UseRouterHost is enabled, this value is the IP address of the router/cable-modem/firewall that is sitting in front of the FTP Server. In response to a PASV command, the server will return this IP address as the local IP address instead of the IP address of the server itself. This will allow the FTP client to open a data connection back to the router (which, in turn, will forward that request back to the server).
RunAtStartup	Controls the startup state of the server. 0 - No, do not start this FTP Server when the Titan Service starts. 1 - Yes, start this FTP Server when the Titan Service starts.
ServerDesc	Text description for the Server.
ServerName	Text name for the server.
SFTPEnabled	1 - Enables SFTP support on this server. 0 - Disables SFTP support on this server.
SFTPPort	Used to identify the port used for SFTP. The default port is port 22 .
SFTPVer	Identifies the maximum SFTP Protocol version supported by Titan. Titan currently supports SFTP v3, v4, v5, and v6.
SFTPHostKeyFolder	The fully qualified path location where Titan will store the user and server host keys.

SFTPRequireHK	<p>1 - Host keys are required from clients who attempt to connect using SFTP. When this feature is enabled, the Titan Administrator will need to import each user's public host key into the Titan Host Key Folder. If a user connects and supplies a host key that is not stored in Titan, the user will not be able to gain access.</p> <p>0 - Host keys are not required from clients who attempt to connect using SFTP.</p>
SFTPHostKeyName	The name of the host key file, stored in the Host Key Folder, used by Titan during SFTP handshaking.
SFTPUseCompression	1 - Titan will use zlib compression if also supported by the client.
SFTPCipherList	<p>A list of encryption ciphers, in order, that Titan will present to the SFTP client during the handshaking/negotiation phase of the connection. The following ciphers are currently supported:</p> <p>3des-cbc blowfish-cbc aes128-cbc aes192-cbc aes256-cbc aes128-ctr aes192-ctr aes256-ctr cast128-cbc arcfour rijndael-cbc@lysator.liu.se</p>
SFTPMAList	<p>A list of MAC algorithms, in order, that Titan will present to the SFTP client during the handshaking/negotiation phase of the connection. The following MAC algorithms are currently supported:</p> <p>md5 sha1 ripemd160 ripemd160@openssh.com sha1-96 md5-96</p>
SSLBannerMsg	Message to display to the user when the user connects to the server over the SSLImplicitPort . For multi-line messages, separate each line with a ' '.
SSLCertStoreFolder	Text location of the SSL certificate store.
SSLCanCCC	<p>1 - Allow support for CCC (Clear Control Channel). This command will allow a secured SSL session to return to a non-secured state.</p> <p>0 - Disallow this command.</p>
SSLDisabledMsg	Message to display to the user if they attempt to initiate an SSL connection and SSLEnabled is disabled. For multi-line messages, separate each line with a ' '.
SSLExplicit	<p>Controls if AUTH SSL is enabled on the Server. This value is ignored if SSLEnabled is disabled.</p> <p>0 - AUTH SSL is not permitted, 1 - AUTH SSL is permitted.</p>
SSLImplicit	<p>Controls if Implicit SSL is enabled on the Server. This value is ignored if SSLEnabled is disabled.</p> <p>0 - Implicit SSL is not allowed, 1 - Implicit SSL is allowed.</p>

SSLImplicitPort	If SSLEnabled and SSLImplicit are enabled, this value represents the port number to be used for Implicit SSL connections. Implicit SSL connections connect to the server and immediately perform the SSL handshaking instead of issuing an AUTH SSL command.
SSLProtoVersion	The SSL protocol version number. 0 - SSL 3.0, 1 - SSL 3.1 (TLS 1.0)
SSLRequireCerts	1 - Require certificates from users who attempt to connect using SSL. 0 - Do not require certificates from users who connect using SSL.
StatMsg	Message to be displayed to the user in response to the STAT command. For multi-line messages, separate each line with a ' '.
StatsArchive	Controls the Archive stats before pruning value. 0 - Do not archive prior to prune, 1 - Archive before prune.
StatsArchiveDate	Contains date of the last archive for the stats database.
StatsDSN	The ODBC data source name identifying the statistics database
StatsEnabled	Controls the setting for Statistics. 0 - Disable statistics logging, 1 - Enable Statistics logging.
StatsMask	A mask defining the information to be recorded. The string is a series of 0 or 1 indicating which statistic is recorded. The following positions, left to right, are defined: <ul style="list-style-type: none"> 0 - File Uploads 1 - File Downloads 2 - User Login Attempts 3 - User Logout Attempts 4 - Client Connection Attempts 5 - Client Disconnects 6 - Directory Listings 7 - Delete Files 8 - Delete Folders 9 - Create Folders <p>For example, if the StatsMask were set to 0100001000, Titan would record statistics for File Downloads and Directory Listings.</p>
StatsRotation	Defines the archive/pruning schedule for the statistics. <ul style="list-style-type: none"> 0 - No pruning (not recommended) 1 - Prune Daily 2 - Prune Weekly 3 - Prune Monthly 4 - Prune Annually
STOUPrefix	Prefix to be used during the generation of a unique file name used in the STOU command.
TimeZoneAdjust	Controls the setting for whether or not to make adjustments for time zone when returning the directory information to the client.
TimeZoneMinutes	If TimeZoneAdjust is enabled, this value indicates the number of minutes to add/subtract from the GMT time for file times.
UNCPassword	Password used for UNC access. For security reasons, this attribute can only be set, not retrieved.

UNCUser	User name used for UNC access.
UseRouterHost	Controls the setting for whether or not this server is sitting behind a router/cable modem/firewall. This setting is used in conjunction with the RouterHost value and the PASV command to return a valid local IP address to the remote FTP client. 0 - Disabled, 1 - Enabled.
UseStdUnixDirs	Controls the creation of standard Unix FTP style directories. If enabled, the Server will create /incoming/ , /usr/ , /bin/ , /pub/ folders under the BaseDataDir .
UseNTHomeDir	Use the NT home directory for a user. 0 - Disabled, 1 - Enabled.
UseNTImpersonation	Use NT Impersonation. 0 - Disabled, 1 - Enabled.
UTF8Support	Controls the UTF8 feature for the server. 0 - UTF8 Disabled, 1 - UTF8 Enabled.
WelcomeMsg	Message to display to a user once they have successfully logged in to the server. For multi-line messages, separate each line with a ' '.

SVR_Create

Creates a new FTP server.

Syntax:

```
SVR_Create(LPCTSTR szServerName, LPCTSTR szServerIP, unsigned short usServerPort, LPCTSTR szBaseDataDir, unsigned short usAuthType );
```

```
SVR_Create2(LPCTSTR szServerName, LPCTSTR szServerIP, unsigned short usServerPort, LPCTSTR szBaseDataDir, unsigned short usAuthType, LONG* IError)
```

Parameters:

szServerName	Name of the FTP server.
szServerIP	IP address that the FTP server is listening on. This will be either an actual IP address or 0.0.0.0 , which means Listen on all available addresses .
usServerPort	Specifies the port number that the server is listening on. The default value is port 21 .
szBaseDataDir	The fully qualified path that refers to the base data directory where all of the server data files will be stored.
usAuthType	Type of User Authentication database to use. 0 - Titan User Authentication 1 - NT User Authentication.
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

SVR_Delete

Deletes an existing FTP server. If the server is currently online and running, it is stopped before being deleted. All user and group information will also be deleted (unless NT Authentication has been specified as the authentication method during the creation).

Syntax:

```
SVR_Delete(LPCTSTR szServerName);
```

```
SVR_Delete2(LPCTSTR szServerName, LONG* IError);
```

Parameters:

szServerName	Name of the FTP server.
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

SVR_Enum

Generates a list of FTP servers that are currently configured on the machine.

Syntax:

```
SVR_Enum ( BSTR* pszServerList );
```

```
SVR_Enum2 ( BSTR* pszServerList, LONG* IError );
```

Parameters:

pszServerList	A pointer that will receive the list of servers defined on the system. The list will be a ' ' delimited list of server names. Example: server1 server2 server3
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

SVR_Start

Instructs the Titan Service to start the FTP server if it is not already running.

Syntax:

```
SVR_Start(LPCTSTR szServerName);
```

```
SVR_Start2(LPCTSTR szServerName, LONG* IError);
```

Parameters:

szServerName	Name of the FTP server.
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

SVR_Stop

Instructs the Titan Service to stop the specified FTP server if it is currently running.

Syntax:

```
SVR_Stop(LPCTSTR szServerName);
```

```
SVR_Stop2(LPCTSTR szServerName, LONG* IError);
```

Parameters:

szServerName	Name of the FTP server.
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

SVR_Restart

Instructs the Titan Service to restart the specified FTP server. If the FTP Server is currently running, it is stopped, then restarted. If it is not running, it is started.

Syntax:

```
SVR_Restart(LPCTSTR szServerName);
SVR_Restart2(LPCTSTR szServerName, LONG* IError);
```

Parameters:

szServerName	Name of the FTP server.
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

SVR_GetAttr

Retrieves a [server configuration attribute](#) for the specified FTP server.

Syntax:

```
SVR_GetAttr(LPCTSTR szServerName, LPCTSTR szAttrName, BSTR* pszAttrValue);
SVR_GetAttr2(LPCTSTR szServerName, LPCTSTR szAttrName, BSTR* pszAttrValue,
LONG* IError);
```

Parameters:

szServerName	Name of the FTP server.
szAttrName	Name of the server configuration attribute to retrieve.
pszAttrValue	Pointer to retrieve the string containing the attribute data.
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

SVR_SetAttr

Sets/Modifies a [server configuration attribute](#) for the specified FTP Server.

Syntax:

```
SVR_SetAttr(LPCTSTR szServerName, LPCTSTR szAttrName, LPCTSTR szAttrValue);
```

```
SVR_SetAttr2(LPCTSTR szServerName, LPCTSTR szAttrName, LPCTSTR szAttrValue,  
LONG* IError);
```

Parameters:

szServerName	Name of the FTP server.
szAttrName	Name of the server configuration attribute to retrieve.
szAttrValue	String containing the attribute data to be applied.
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

Group Configuration Attributes

The following is a list of **configuration attributes** for **groups**. These values can be retrieved and set using the **srxCfg Utility** or **srxCOM Interface**.

A general note about Message attributes. These messages are returned to the user/ftp client. Messages can be one or more lines and should be kept to about 60 characters per line. If you want to have a multi-line message, separate each message with a '|'. For example:
AcctDisabledMsg="Hello, Your Account Has Been Disabled.|Please contact your system administrator to have it re-opened.|GoodBye!".

List of Shared Attributes -- these attributes are shared at the **Server, Group, and User** level.

List of Group-Specific Attributes

Attribute	Description
AlwaysAllowLogin	Always allow members of this group to login, even if the maximum number of connections for the server has been reached. 0 - Disabled, 1 - Enabled.
Enabled	Enable or disable this group. If a group is disabled, any users that inherit the Enabled value from the group will also be disabled. 0 - Disabled, 1 - Enabled.
ExpirationDateOn	Enable or disable a group expiration date. 0 - Disabled, 1 - Enabled.
ExpirationDate	The expiration date for the group.
GroupId	Internal ID for the group. Use this ID in the Users Membership attribute.
GroupName	Text name for the group.
HomeDir	Home directory for the group. If HomeDirEnabled is set, users that are a member of this group will have a home directory based on the group home directory.
HomeDirEnabled	Enable or disable the concept of a Group home directory. 0 - Disabled, no group home directory 1 - Enabled, user home directory is set to Group home directory. 2 - Enabled, user home directory is set to Group home directory plus username. Example: if username=bob and group home dir="C:\group\" , then user home dir="C:\group\bob\"
Members	A ' ' separated list of user names of the users who are members of this group. Example: user1 user22 user44

GRP_Create

Creates a new group in the specified FTP server. The group has no members when you first create it.

Syntax:

```
GRP_Create(LPCTSTR szServerName, LPCTSTR szGroupName);
```

```
GRP_Create2(LPCTSTR szServerName, LPCTSTR szGroupName, LONG* IError);
```

Parameters:

szServerName	Name of the FTP Server.
szGroupName	The name of the new Group.
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

GRP_Delete

Deletes an existing group from the FTP server.

Syntax:

```
GRP_Delete(LPCTSTR szServerName, LPCTSTR szGroupName);
```

```
GRP_Delete2(LPCTSTR szServerName, LPCTSTR szGroupName, LONG* IError);
```

Parameters:

szServerName	Name of the FTP server.
szGroupName	Name of the group to delete.
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

GRP_Enum

Generates a list of currently defined groups for the specified FTP server.

Syntax:

```
GRP_Enum ( LPCTSTR szServerName, BSTR* pszGroupList );
```

```
GRP_Enum2 ( LPCTSTR szServerName, BSTR* pszGroupList, LONG* IError );
```

Parameters:

szServerName	Name of the FTP server.
pszGroupList	A pointer that will receive the list of groups defined on the specified FTP server. The list will be a ' ' delimited list of group names. Example: group1 group3
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

GRP_GetMembers

Retrieves a list of members (users) for the specified FTP server and group.

Syntax:

```
GRP_GetMembers(LPCTSTR szServerName, LPCTSTR szGroupName, BSTR* pszMembers );
```

```
GRP_GetMembers2(LPCTSTR szServerName, LPCTSTR szGroupName, BSTR* pszMembers, LONG* IError );
```

Parameters:

szServerName	Name of the FTP server
szGroupName	Name of the group.
pszMembers	A pointer that will receive the list of members defined on the specified FTP Server. The list will be a ' ' delimited list of member names. Example: group1 group3
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

GRP_SetMembers

Sets/Assigns the list of members (users) to the group.

Syntax:

```
GRP_SetMembers(LPCTSTR szServerName, LPCTSTR szGroupName, LPCTSTR szMembers);
```

```
GRP_SetMembers2(LPCTSTR szServerName, LPCTSTR szGroupName, LPCTSTR szMembers, LONG* IError);
```

Parameters:

szServerName	Name of the FTP server.
szGroupName	Name of the group.
szMembers	' ' separated list of users who will be members of this group. Example: user1 user3 user5 .
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

GRP_GetAttr

Retrieves a **group configuration attribute** for the specified FTP Server and group.

Syntax:

```
GRP_GetAttr(LPCTSTR szServerName, LPCTSTR szGroupName, LPCTSTR  
szAttrName, BSTR* pszAttrValue);
```

```
GRP_GetAttr2(LPCTSTR szServerName, LPCTSTR szGroupName, LPCTSTR  
szAttrName, BSTR* pszAttrValue, LONG* IError);
```

Parameters:

szServerName	Name of the FTP server.
szGroupName	Name of the group.
szAttrName	Name of the group configuration attribute to retrieve.
pszAttrValue	Pointer to retrieve the string containing the attribute data.
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

GRP_SetAttr

Sets/Modifies a **group configuration attribute** for the specified FTP Server/group.

Syntax:

```
GRP_SetAttr(LPCTSTR szServerName, LPCTSTR szGroupName, LPCTSTR  
szAttrName, LPCTSTR szAttrValue);
```

```
GRP_SetAttr2(LPCTSTR szServerName, LPCTSTR szGroupName, LPCTSTR  
szAttrName, LPCTSTR szAttrValue, LONG* IError);
```

Parameters:

szServerName	Name of the FTP server.
szGroupName	Name of the group.
szAttrName	Name of the group configuration attribute to retrieve.
szAttrValue	String containing the attribute data to be applied.
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

User Configuration Attributes

The following is a list of configuration attributes for Users. These values can be retrieved and set using the **srxCfg Utility** or **srxCOM Interface**.

A general note about message attributes. These messages are returned to the user/ftp client. Messages can be one or more lines and should be kept to about 60 characters per line. If you want to have a multi-line message, separate each message with a '|'. For example:
AcctDisabledMsg="Hello, Your Account Has Been Disabled.|Please contact your system administrator to have it re-opened.|GoodBye!".

List of Shared Attributes -- these attributes are shared at the **Server, Group, and User level**.

List of User-Specific Attributes

Attribute	Description
AlwaysAllowLogin	Always allow this user to log in, even if the maximum number of connections for the server has been reached. 0 - Disabled, 1 - Enabled.
EmailAddress	Email address for the user.
Enabled	Enable or disable this user. 0 - Disabled, 1 - Enabled.
ExpirationDateOn	Enable or disable a group expiration date. 0 - Disabled, 1 - Enabled.
ExpirationDate	The expiration date for the group.
FullName	User's full name.
HomeDirInherit	Enable or disable to determine if the Home Directory value should be taken from the group or user attribute. 0 - Disabled, user's home directory will be user's own home directory settings; user will not inherit home directory from the group 1 - Enabled, user will inherit home directory from the group level
Membership	A ' ' separated list of group names of the groups to which this user belongs. Example: group1 group22 group44
Password	Password for this user account. For security reasons, this attribute can only be set, not retrieved.
PwdType	Password Encryption Type. 0 - Standard Password 1 - Anything 2 - OTP S/Key MD4 3 - OTP S/Key MD5
UserId	Internal ID for the user.
Username	Username for the user's account.

USR_Create

Creates a new user for the specified FTP Server.

Syntax:

```
USR_Create(LPCTSTR szServerName, LPCTSTR szUserName, LPCTSTR szPassword);
```

```
USR_Create2(LPCTSTR szServerName, LPCTSTR szUserName, LPCTSTR szPassword, LONG* IError);
```

Parameters:

szServerName	Name of the FTP server.
szUserName	The name of the new user.
szPassword	The new user's password.
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

USR_Delete

Deletes an existing user from the FTP Server.

Syntax:

```
USR_Delete(LPCTSTR szServerName, LPCTSTR szUserName );
```

```
USR_Delete2(LPCTSTR szServerName, LPCTSTR szUserName, LONG* IError );
```

Parameters:

szServerName	Name of the FTP server.
szUserName	Name of the user to be deleted.
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

USR_Enum

Generates a list of currently defined users for the specified FTP server.

Syntax:

```
USR_Enum (LPCTSTR szServerName, BSTR* pszUserList);
```

```
USR_Enum2 (LPCTSTR szServerName, BSTR* pszUserList, LONG* IError);
```

Parameters:

szServerName	Name of the FTP server.
pszUserList	A pointer that will receive the list of users defined on the specified FTP Server. The list will be a ' ' delimited list of names. Example: user1 user2
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

USR_GetAttr

Retrieves a [user configuration attribute](#) for the specified FTP Server and user.

Syntax:

```
USR_GetAttr(LPCTSTR szServerName, LPCTSTR szUserName, LPCTSTR szAttrName,
BSTR* pszAttrValue);
```

```
USR_GetAttr2(LPCTSTR szServerName, LPCTSTR szUserName, LPCTSTR
szAttrName, BSTR* pszAttrValue, LONG* IError);
```

Parameters:

szServerName	Name of the FTP server.
szUserName	Name of the user.
szAttrName	Name of the user configuration attribute to retrieve.
pszAttrValue	Pointer to retrieve the string containing the attribute data.
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

USR_SetAttr

Sets/Modifies a [user configuration attribute](#) for the specified FTP server/user.

Syntax:

```
USR_SetAttr(LPCTSTR szServerName, LPCTSTR szUserName, LPCTSTR szAttrName,
LPCTSTR szAttrValue);
```

```
USR_SetAttr2(LPCTSTR szServerName, LPCTSTR szUserName, LPCTSTR
szAttrName, LPCTSTR szAttrValue, LONG* IError);
```

Parameters:

szServerName	Name of the FTP server.
szUserName	Name of the user.
szAttrName	Name of the user configuration attribute to retrieve.
szAttrValue	String containing the attribute data to be applied.
IError	Optional parameter to retrieve the error code. If an error occurs, a non-zero value will be returned. The return value can be supplied into the SRX_GetErrStr() method to return the actual error code text.

Return Values:

This method does not return a value.

Server Commands and Return Codes

This section outlines the list of FTP commands currently implemented by Titan FTP Server.

Standard Commands

- ABOR** - Instructs the server to abort the current command.
- APPE** - Appends data to an existing file.
- CDUP** - Changes the current working directory to the parent directory.
- CWD** - Changes the current working directory to the relative or absolute path specified.
- DELE** - Deletes the specified file object.
- HELP** - Displays a list of implemented commands.
- LIST** - Generates a list of files for the specified (or current) directory. The file list is returned on a data connection.
- NLST** - Generates a list of filenames for the specified (or current) directory. The filename list is returned on a data connection.
- MKD** - Creates a folder.
- MODE** - Specifies the data transfer mode. Stream and Zlib are supported.
- NOOP** - Pings the server to keep the control connection alive.
- PASS** - Sends the user's password to the server.
- PASV** - Instructs the server to go into passive mode and return an IP/Port combination to be used for a data connection.
- PORT** - Instructs the server to use the supplied IP/Port combination during the establishment of the next data connection.
- PWD** - Displays the current working directory.
- QUIT** - Terminates the user's session and closes the control connection.
- REIN** - Reinitializes the control connection. The currently authenticated user is cleared out and reset for a new USER.
- REST** - Specifies an offset for restarting a data transfer.
- RETR** - Used to retrieve a file from the server.
- RMD** - Removes/Deletes a directory folder from the server. The folder must be empty.
- RNFR** - Renames a file/folder. Used in conjunction with **RNTO**.
- RNTO** - Renames a file/folder. Used in conjunction with **RNFR**.
- SITE** - Special command used to issue site-specific instructions to Titan FTP Server.
- STAT** - Displays status information.
- STOR** - Instructs the server to begin storing a file that will be sent over the data connection.
- STOU** - Instructs the server to generate a unique filename used to store data being sent over the data connection.
- STRU** - Specifies the structure of data on the server. File format is currently supported.
- SYST** - Displays the system type for the server.
- TYPE** - Sets the data representation on the server.
- USER** - Sends the username to the server.

Advanced Commands

- AUTH** - Used to initiate an SSL encrypted session.
- COMB** - COMBines file segments into a single file on the server.
- DQTA** - Returns disk quota information.
- FEAT** - Displays a list of extensions supported by the server.
- MDTM** - Displays/sets date/time information for files.
- MLST** - Displays file information over the control connection.
- MLSD** - Generates/displays directory information over the control connection.
- OPTS** - Allows for the configuration/enabling of special options supported by the server.
- PBSZ** - Sets the Protected Buffer Size for an SSL data connection.
- PROT** - Sets the Protection Level for an SSL data connection.
- SIZE** - Returns the size of a supplied file.
- XCRC** - Performs a CRC-32 checksum of the user supplied filename.
- CCSN** - Used during secure FXP to set the handshaking role of the server.
- CPSV** - Used during secure FXP to set the handshaking role of the server. This is an alternate form of **CCSN**.
- EPRT** - Similar to the **PORT** command, but for IP v6 addressing.
- EPSV** - Similar to the **PASV** command, but used for IP v6 addressing.

SITE Commands

- SITE PSWD** - Used to modify the current user's password on the server.
- SITE ZONE** - Used to display the current time zone setting for the server.

FTP Return Codes

FTP return codes are used to communicate how the server responded to a given command. Each command that is issued is followed by a subsequent return code.

500 Series

The command was not accepted and the requested action did not take place.

500

Syntax error, command unrecognized. This may include errors such as command line too long.

501

Syntax error in parameters or arguments.

502

Command not implemented.

503

Bad sequence of commands.

504

Command not implemented for that parameter.

530

Not logged in.

532

Need account for storing files.

550

Requested action not taken. File unavailable (For example, **file not found, no access**).

551

Requested action aborted. Page type unknown.

552

Requested file action aborted. Exceeded storage allocation (for current directory or dataset).

553

Requested action not taken. File name not allowed.

400 Series

The command was not accepted and the requested action did not take place, but the error condition is temporary and the action may be requested again.

421

Service not available, closing control connection. This may be a reply to any command if the service knows it must shut down.

425

Cannot open data connection.

426

Connection closed; transfer aborted.

450

Requested file action not taken.

451

Requested action aborted. Local error in processing.

452

Requested action not taken. Insufficient storage space in system. File unavailable (For example, file busy).

300 Series

The command has been accepted, but the requested action is dormant, pending receipt of further information.

331

User name okay, need password.

332

Need account for login.

350

Requested file action pending further information.

200 Series

The requested action has been successfully completed.

200

Command okay.

202

Command not implemented, superfluous at this site.

211

System status, or system help reply.

212

Directory status.

213

File status.

214

Help message. On how to use the server or the meaning of a particular non-standard command. This reply is useful only to the human user.

215

NAME system type. Where NAME is an official system name from the list in the Assigned Numbers document.

220

Service ready for new user.

221

Service closing control connection.

225

Data connection open; no transfer in progress.

226

Closing data connection. Requested file action successful (for example: **file transfer**, **file abort**).

227

Entering Passive Mode (h1,h2,h3,h4,p1,p2).

230

User logged in, proceed. Logged out if appropriate.

250

Requested file action okay, completed.

257

PATHNAME created.

100 Series

The requested action is being initiated, expect another reply before proceeding with a new command.

110

Restart marker reply. In this case, the text is exact and not left to the particular implementation; it must read: **MARK yyyy = mmmm** (yyyy is user-process data stream marker, and mmmm server's equivalent marker. Note the spaces between **markers** and =).

120

Service ready in nnn minutes.

125

Data connection already open; transfer starting.

150

File status okay; about to open data connection.

SFTP Commands Overview

This section outlines the list of **SFTP commands** currently implemented by Titan FTP Server. Note that some of these commands are only valid in later protocol versions. If your SFTP client does not support the latest version of the protocol, you will be unable to use the new commands.

INIT
OPEN
CLOSE
READ
WRITE
LSTATS
FSTAT
SETSTAT
FSETSTAT
OPENDIR
READDIR
REMOVE
MKDIR
RMDIR
REALPATH
STAT
RENAME
READLINK
SYMLINK
LINK
BLOCK
UNBLOCK

SFTP Return Codes Overview

This section outlines the list of **SFTP return codes** currently implemented by Titan FTP Server.

Ok - Indicates successful completion of the operation.

End of file - An attempt to read past the end-of-file was made, or there are no more directory entries to return.

No such file - A reference was made to a file that does not exist.

Permission denied - The user does not have sufficient permissions to perform the operation.

Failure - An error occurred, but no specific error code exists to describe the failure.

Bad message - A badly formatted packet or other SFTP protocol incompatibility was detected.

No connection - There is no connection to the server.

Connection lost - The connection to the server was lost.

Operation unsupported - An attempted operation could not be completed by the server because the server does not support the operation.

Invalid handle - The handle value was invalid.

No such path - The file path does not exist or is invalid.

File already exists - The file already exists.

Write protect - The file is on read-only media, or the media is write protected.

No media - The requested operation cannot be completed because there is no media available in the drive.

No space on filesystem - The requested operation cannot be completed because there is no free space on the filesystem.

Quota exceeded - The operation cannot be completed because it would exceed the user's storage quota.

Unknown principal - A principal referenced by the request was unknown (either the **owner**, **group**, or **who** field of an **ACL**).

Lock conflict - The file could not be opened because it is locked by another process.

Directory not empty - The directory is not empty.

Not a directory - The specified file is not a directory.

Invalid filename - The filename is not valid.

Link loop - Too many symbolic links encountered.

Cannot delete - The file cannot be deleted. One possible reason is that the advisory **READONLY** attribute-bit is set.

Invalid parameters - One of the parameters was out of range, or the parameters specified cannot be used together.

File is a directory - The specified file was a directory in a context where a directory cannot be used.

Byte range lock conflict - A read or write operation failed because the mandatory byte-range lock of another process overlaps with the request.

Byte range lock refused - A request for a byte range lock was refused.

Delete pending - An operation was attempted on a file for which a delete operation is pending.

File corrupt - The file is corrupt; a filesystem integrity check should be run.

Standard FTP Commands

ABOR

Overview

The **ABOR** command tells the server to abort the previous FTP service command and any associated transfer of data.

The abort command may require "special action" to force recognition by the server. No action is to be taken if the previous command has been completed (including data transfer). The data connection, if open, will be instructed to close.

There are two cases for the server upon receipt of this command:

- (1) the FTP service command was already completed, or
- (2) the FTP service command is still in progress.

In the first case, the server closes the data connection (if it is open) and responds with a **226** reply, indicating that the abort command was successfully processed. In the second case, the server aborts the FTP service in progress and closes the data connection, returning a **426** reply to indicate that the service request terminated abnormally. The server then sends a **226** reply, indicating that the abort command was successfully processed.

Format

ABOR<crLf>

Returns

226 - The command was successful.

421 - The server is off-line or is going off-line.

426 - The command was successful, data connection was closed, and the data transfer aborted.

530 - No user is currently authenticated on the command channel. A client must be authenticated prior to executing this command.

APPE

Overview

The **APPE** command causes the server to accept the data transferred via the data connection and to store the data in a file at the server site. If the file specified in the **<name>** exists at the server site, then the data shall be appended to that file; otherwise, the file specified in the **<name>** shall be created at the server site.

Format

APPE <name><CRLF>

Returns

150 - File status okay; about to open data connection. Client may begin transferring data over the data connection.

421 - The server is off-line or is going off-line.

451 - Local error in processing. This is usually a result of an internal server error, such as a thread not starting or memory allocation error.

530 - No user is currently authenticated on the command channel. A client must be authenticated prior to executing this command.

501 - Syntax error in parameters or arguments. This usually results from an invalid or missing **<name>**.

550 - Requested action not taken, file not found or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action, or if **<name>** was not found.

552 - Requested file action aborted, exceeded user's storage allocation. This error usually occurs if there is not enough storage on the server to perform the action. It may also occur if the user has exceeded their allotted disk quota.

553 - File name not allowed, or file type banned. This error usually occurs if the user is attempting to store a file that has been banned by the server administrator.

CDUP

Overview

The **CDUP** command is a special case of **CWD**, and is included to simplify the implementation of programs for transferring directory trees between operating systems having different formats for naming the parent directory. The **CDUP** command changes the current working directory to be the parent directory.

Format

CDUP<crlf>

Returns

250 - The command was successful.

421 - The server is off-line or is going off-line.

501 - Syntax error in parameters or arguments. This usually results from adding parameters to the end of the command. This command does not take any parameters.

530 - No user is currently authenticated on the command channel.

550 - Requested action not taken, file not found, or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action.

CWD

Overview

The **CWD** command allows the user to work with a different directory for file storage or retrieval without altering his credentials or accounting information. The argument is a **<name>** specifying a relative or absolute directory path.

Format

CWD <name><crlf>

Returns

250 - The command was successful.

421 - The server is off-line or is going off-line.

501 - Syntax error in parameters or arguments. This usually results from an invalid or missing **<name>**.

530 - No user is currently authenticated on the command channel.

550 - Requested action not taken, file not found or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action, or if **<name>** was not found.

DELE

Overview

The **DELE** command causes the file specified in the **<name>** to be deleted from the server. **<name>** can be an absolute fully qualified path and file name to be deleted, or it can be relative from the current working directory.

Format

DELE <name><crlf>

Returns

250 - The command was successful.

421 - The server is off-line or is going off-line.

501 - Syntax error in parameters or arguments. This usually results from an invalid or missing **<name>**.

530 - No user is currently authenticated on the command channel.

550 - Requested action not taken, file not found, or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action, or if **<name>** was not found.

HELP

Overview

The **HELP** command shall cause the server to send helpful information regarding its implementation status over the control connection to the user. The Titan FTP Server implementation of this command will return a list of commands that are currently implemented by the server.

Format

HELP <crlf>

Returns

214 - The following commands are recognized (* => not implemented).

421 - The server is off-line or is going off-line.

LIST

Overview

The **LIST** command causes a list of filenames to be sent from the server to the client. If the **<name>** specifies a directory or other group of files, the server will transfer a list of filenames in the specified directory. If the **<name>** specifies a file then the server will send current filename information on the specified file. A null/missing **<name>** argument implies the user's current working directory or default directory. The data transfer is over the data connection in type **ASCII** so the user must ensure that the **TYPE** is appropriately set to **ASCII**. Currently supported **<args>** are:

- h** displays hidden files
- a** does not include the '.' and '..' directories in the listing
- F** adds file characterizations to the listing. Directories are terminated with a '/' and executable files are terminated with a '*'.
- A** displays All files.
- T** when used with -l, displays the full month, day, year, hour, minute, and second for the file date/time.

Format

LIST [-<args>] [<name>] <crlf>

Returns

150 - File status okay; about to open data connection. Client may begin receiving data over the data connection.

421 - The server is off-line or is going off-line.

451 - Local error in processing. This is usually a result of an internal server error such as a thread not starting or memory allocation error.

530 - No user is currently authenticated on the command channel. A client must be authenticated prior to executing this command.

550 - Requested action not taken, file not found, or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action, or if **<name>** was not found.

MKD

Overview

The **MKD** command causes the server to create the directory specified in the **<name>**. **<name>** can be an absolute fully qualified path or it can be a relative directory from the current working directory.

Format

MKD <name><crlf>

Returns

250 - The command was successful.

421 - The server is off-line or is going off-line.

501 - Syntax error in parameters or arguments. This usually results from an invalid **<name>**.

530 - No user is currently authenticated on the command channel.

550 - Requested action not taken, file not found or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action.

MODE

Overview

The **MODE** command allows the client to specify the data transfer mode for the server. Titan FTP Server currently only supports ZLIB compression mode.

Format

MODE Z<crLf> - Enables Zlib compression.

MODE S<crLf> - Disables Zlib compression.

Returns

200 - The command was successful.

421 - The server is off-line or is going off-line.

504 - Command not implemented for that parameter.

530 - No user is currently authenticated on the command channel.

NOOP

Overview

The **NOOP** command is simply used as a **ping** command for the server. This command is mainly used to keep the control channel alive during idle periods.

Format

NOOP<crLf>

Returns

200 - OK.

421 - The server is off-line or is going off-line.

PASS

Overview

The **PASS** command is used to send the user's password over to the server. This command must be immediately preceded by the **USER** command and completes the user's identification for access control.

Format

PASS <password><crlf>

Returns

230 - The command was successful, user authenticated.

421 - The server is off-line or is going off-line.

503 - Bad sequence of commands. The **PASS** command must be immediately preceded by the **USER** command.

530 - Access denied. This can happen as a result of many things: the username was invalid, the password was invalid, the user account is disabled, or the user is being denied access for some reason. Possible reasons could be IP address restrictions, user account expired, or too many concurrent logons.

PASV

Overview

The **PASV** command requests the server to listen on a data port (which is not the default data port) and to wait for a connection rather than initiate one upon receipt of a transfer command. The response to this command (if positive) includes the host and port address on which this server is listening.

Note: Some SMC routers intercept **PASV** commands sent over from the FTP client and will re-map those commands as **P@SW** in an attempt to prevent the back-end FTP server from going in to passive mode. Titan FTP Server will process **P@SW** commands as if it were a **PASV** command to overcome this SMC Barricade bug.

Format

PASV<crLf>

Returns

227 - Entering Passive Mode (x,x,x,x,x,x).

421 - The server is off-line or is going off-line.

502 - Command not implemented. This response will be sent if the server is configured with **PASV** mode disabled.

530 - No user is currently authenticated on the command channel.

PORT

Overview

The **PORT** command is used to relay the host and port information for a data connection to the server. If this command is used, the argument is the concatenation of a 32-bit Internet host address and a 16-bit TCP port address. This address information is broken into 8-bit fields and the value of each field is transmitted as a decimal number (in character string representation). The fields are separated by commas.

A typical PORT command would be:

PORT h1,h2,h3,h4,p1,p2

(h1 is the high order 8 bits of the Internet host address.)

Format

PORT h1,h2,h3,h4,p1,p2<crLf>

Returns

200 - The command was successful.

421 - The server is off-line or is going off-line.

501 - Syntax error in parameters or arguments. This usually results from invalid port information. It can also be sent back if the IP address is banned/disallowed from connecting to the server.

530 - No user is currently authenticated on the command channel.

PWD

Overview

The PWD command causes the name of the current working directory to be returned in the reply command.

Format

PWD<crLf>

Returns

257 - <current-directory> is current directory.

421 - The server is off-line or is going off-line.

530 - No user is currently authenticated on the command channel.

550 - Requested action not taken, file not found or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action.

QUIT

Overview

The **QUIT** command terminates a user session and if a file transfer is not in progress, closes the control connection. If a file transfer is in progress, the connection will remain open for the result response and the server will then close the control connection. If the user wants to terminate the session, but not close the control connection, **REIN** should be used instead of the **QUIT** command.

Format

QUIT<crLf>

Returns

221- The command was successful; closing the control connection.

421 - The server is off-line or is going off-line.

530 - No user is currently authenticated on the command channel.

REIN

Overview

The **REIN** command terminates a user session, flushing all I/O and account information, except to allow any transfer in progress to be completed. All parameters are reset to the default settings and the control connection is left open. This is identical to the state in which a user finds himself immediately after the control connection is opened. A **USER** command may be expected to follow this command.

Format

REIN<crLf>

Returns

- 220** - The command was successful.
- 421** - The server is off-line or is going off-line.

REST

Overview

The **REST** command is used to set a marker that will later be used to determine the starting point for a file transfer. The argument field represents the server marker at which a file transfer is to be restarted. This command does not cause the file transfer to be initiated rather instructs the server to skip over the file to the specified data checkpoint. This command shall be immediately followed by the appropriate FTP service command which shall the file transfer to resume, usually **STOR** or **RETR**.

Format

REST <restart-position><crLf>

Returns

- 350** - Restarting at <restart-position> - Send **STOR**e or **RETR**ieve to initiate transfer.
- 421** - The server is off-line or is going off-line.
- 530** - No user is currently authenticated on the command channel.

RETR

Overview

The **RETR** command causes the server to initiate the transfer of the specified file over a newly opened data connection.

Format

RETR <name><crlf>

Returns

150 - The command was successful, data transfer starting.

421 - The server is off-line or is going off-line.

451 - Local error in processing. This is usually a result of an internal server error such as a thread not starting or memory allocation error. This can also occur if a bad **REST** marker value has been set.

501 - Syntax error in parameters or arguments. This usually results from an invalid or missing <name>.

530 - No user is currently authenticated on the command channel.

550 - Requested action not taken, file not found or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action, or if <name> was not found.

553 - File name not allowed, or file type banned. This error usually occurs if the user is attempting to retrieve a file that has been banned by the server administrator, the user UL/DL ratios have not been met, or the user's Download Quotas have been exceeded.

RMD

Overview

The **RMD** command causes the server to remove/delete the directory specified in **<name>**. In order to complete successfully, the directory must be empty. **<name>** can be an absolute fully qualified path or it can be a relative directory from the current working directory.

Format

RMD <name><crlf>

Returns

250 - The command was successful.

421 - The server is off-line or is going off-line.

501 - Syntax error in parameters or arguments. This usually results from an invalid or missing **<name>**.

530 - No user is currently authenticated on the command channel.

550 - Requested action not taken, file not found or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action, or if **<name>** was not found.

RNFR

Overview

The **RNFR/RNTO** commands are used to rename a file/folder on the server. The **RNFR** command specifies the original name of an existing file/folder that is to be renamed. This can be either a relative name or an absolute path and file name. This command must be immediately followed by a **RNTO** command specifying the new file/path name.

Format

RNFR <name><crlf>

Returns

350 - The file exists, continue with **RNTO**.

421 - The server is off-line or is going off-line.

501 - Syntax error in parameters or arguments. This usually results from an invalid or missing file name.

530 - No user is currently authenticated on the command channel.

550 - Requested action not taken, file not found, or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action, or if the file was not found.

RNTO

Overview

The **RNFR/RNTO** commands are used to rename a file/folder on the server. The **RNTO** command specifies the new name of the file/folder that is to be renamed. This can be either a relative name or an absolute path name. This command must be immediately preceded by a **RNFR** command specifying the old file/path name.

Format

RNTO <name><crLf>

Returns

250 - The command was successful.

421 - The server is off-line or is going off-line.

501 - Syntax error in parameters or arguments. This usually results from an invalid or missing file name.

503 - Bad sequence of commands. The **RNTO** command must be immediately preceded by the **RNFR** command.

530 - No user is currently authenticated on the command channel.

550 - Requested action not taken, file not found or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action, or if the file was not found.

553 - File name not allowed, or file type banned. This error usually occurs if the user is attempting to rename to a file that has been banned by the server administrator.

SITE

Overview

The **SITE** command is used by the server to provide services specific to this system that are essential to file transfer but not sufficiently universal to be included as commands in the protocol.

Titan FTP Server currently supports the following **SITE** commands:

Format

SITE PSWD <oldpswd> <newpswd>crLf>

Returns

200 - The command completed successfully.

202 - Command not implemented. This is the generic response if the feature is disabled on the server.

421 - The server is off-line or is going off-line.

501 - Syntax error in parameters or arguments.

530 - No user is currently authenticated on the command channel.

STAT

Overview

The **STAT** command shall cause a status response to be sent over the control connection in the form of a reply.

The command may be sent during a file transfer (along with the Telnet IP and Sync signals) in which case the server will respond with the status of the operation in progress, or it may be sent between file transfers. In the latter case, the command may have an argument field.

If the argument is a path name, the command is analogous to the **LIST** command except that data shall be transferred over the control connection. If a partial path name is given, the server may respond with a list of file names or attributes associated with that specification. If no argument is given, the server will return general status information about the session. This information will use [Custom Message Variables](#) to display the status.

Format

```
STAT <name> <crLf>  
STAT <crLf>
```

Returns

211 - <Custom STAT Message>.

212 - Directory Status.

213 - File Status.

421 - The server is off-line or is going off-line.

501 - Syntax error in parameters or arguments. This usually results from an invalid or missing file name.

530 - No user is currently authenticated on the command channel.

550 - Requested action not taken, file not found or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action, or if the file was not found.

STOR

Overview

The **STOR** command is used to initiate a data transfer to the server. This command causes the server to accept the data transferred via the data connection and to store the data as a file at the server site under the supplied **<name>**. If the file specified in **<name>** exists on the server, then its contents shall be replaced by the data being transferred. A new file is created on the server if the file specified in **<name>** does not already exist. This command can be used in conjunction with a **REST** command to set the starting position in the file.

Format

STOR <name> <crlf>

Returns

150 - File status okay; about to open data connection. Client may begin transferring data over the data connection.

421 - The server is off-line or is going off-line.

451 - Local error in processing. This is usually a result of an internal server error such as a thread not starting or memory allocation error.

501 - Syntax error in parameters or arguments. This usually results from an invalid or missing file name.

530 - No user is currently authenticated on the command channel. A client must be authenticated prior to executing this command.

550 - Requested action not taken, file not found or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action.

552 - Requested file action aborted; exceeded user's storage allocation. This error usually occurs if there is not enough storage on the server to perform the action. It may also occur if the user has exceeded their allotted disk quota.

553 - File name not allowed, or file type banned. This error usually occurs if the user is attempting to store a file that has been banned by the server administrator.

STOU

Overview

The **STOU** command is used to initiate a data transfer to the server. This command causes the server to accept the data transferred via the data connection and to store the data as a file at the server site under a unique name. The file is stored in the current working directory. Upon successful execution of the command, **STOU** will return **150** with the new unique file name used. Each unique file name will be prefixed with a string that is configurable at the **Server Level**, see **STOU Prefix** for more information.

Format

STOU<crLf>

Returns

150 - FILE: <unique-filename>.

421 - The server is off-line or is going off-line.

451 - Local error in processing. This is usually a result of an internal server error such as a thread not starting or memory allocation error.

501 - Syntax error in parameters or arguments. This usually results from an invalid or missing file name.

530 - No user is currently authenticated on the command channel. A client must be authenticated prior to executing this command.

550 - Requested action not taken, file not found, or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action.

552 - Requested file action aborted, exceeded user's storage allocation. This error usually occurs if there is not enough storage on the server to perform the action. It may also occur if the user has exceeded their allotted disk quota.

553 - File name not allowed, or file type banned. This error usually occurs if the user is attempting to store a file that has been banned by the server administrator.

SYST

Overview

The **SYST** command is used to determine the type of operating system at the server. The first word of the reply shall be one of the system names list in the current version of the [IANA Assigned Numbers](#) document.

The Titan FTP Server implementation currently returns **UNIX Type: L8**.

Format

SYST<crLf>

Returns

215 - UNIX Type: **L8**

421 - The server is off-line or is going off-line.

530 - No user is currently authenticated on the command channel.

TYPE

Overview

The **TYPE** command is used to set the data representation on the server.

A - **ASCII** data type

E - **EBCDIC** data type (*) Not currently supported by Titan FTP Server

I - Image data type

The default representation is **ASCII** non-print.

Format

TYPE <data type><crLf>

Returns

200 - Command successful.

421 - The server is off-line or is going off-line.

504 - Command not implemented for that parameter.

530 - No user is currently authenticated on the command channel.

USER

Overview

The **USER** command is used to start a new user session and to send the username to the server. The argument field is a **telnet string** identifying the user. The user identification is required by the server for access to the system. This command will normally be the first command transmitted by the user after the control connection is made. Additional identification information in the form of a **PASSword** may be required. The **USER** command may be entered at any point in order to change the access control and/or accounting information. This has the effect of flushing any **user**, **password**, and **accounting** information already supplied and beginning the authentication sequence again. All transfer parameters are unchanged and any file transfer in progress is completed under the old access control parameters.

Format

USER <username><crLf>

Returns

230 - User logged in.

331 - Command successful. Send Password.

421 - The server is off-line or is going off-line.

501 - Syntax error in parameters or arguments.

530 - User access denied. This usually results from an IP access restriction at the Server level.

Advanced FTP Commands

AUTH

Overview

The **AUTH** command is used as part of the **SSL/FTP Security Extensions (RFC 2228)**. The **AUTH** command is issued by the client to the server to specify the authentication mechanism name to be used for securing the FTP session. The **AUTH** command will be used in conjunction with the **PBSZ** and **PROT** commands for SSL support. Titan FTP Server currently supports the following security mechanism names:

SSL - Instructs Titan FTP Server that SSL v3 encryption will be used.

TLS - Instructs Titan FTP Server that SSL v3.1 (i.e. TLS v1.0) or SSL v3 encryption will be used.

For each of the supported authentication types, an optional **Data Channel Protection Level** can be supplied:

AUTH ???-P - Instructs Titan FTP Server to encrypt/protect the data channel by default. This is equivalent to issuing the **PROT P** command.

AUTH ???-C - Instructs Titan FTP Server to not encrypt the data channel by default. This is equivalent to issuing the **PROT C** command.

Format

AUTH <mechanism-name>[-P|C]<crlf>

Returns

234 - Security data exchange complete.

421 - The server is off-line or is going off-line.

502 - SSL/TLS not enabled on this server.

504 - Command not implemented for that parameter.

530 - No user is currently authenticated on the command channel.

CCSN

Overview

The **CCSN** command is used during secure FXP (site-to-site/server-to-server file transfers). The role of the CCSN command is to allow the FTP client to instruct either the source or destination server to initiate the handshaking for a secure transfer. During a SSL/TLS negotiation, one of the two entities initiates the secure handshake needed to establish the secure connection. Since an FXP occurs between two servers instead of a client and server, one of the two servers needs to assume the role of client in the handshake process.

Issuing the CCSN command with no arguments will return the current handshake state of the server.

The following modes are supported:

Server Mode - Indicates that the Titan server is in **Server Mode** and will not initiate the SSL/TLS handshake during an FXP negotiation. This is the default setting.

Client Mode - Indicates that the Titan server is in **Client Mode** and will initiate the SSL/TLS handshake during an FXP negotiation.

Format

CCSN [ON|OFF]<crlf>

ON - Instructs the Titan FTP Server to enter in to Client Mode.

OFF - Instructs the Titan FTP Server to leave Client Mode and enter in to Server Mode.

Returns

200 - CCSN: Client Mode | Server Mode.

421 - The server is off-line or is going off-line.

502 - SSL/TLS/CCSN not enabled on this server.

504 - Syntax error in arguments.

COMB

Overview

The **COMB** command is used to combine file segments on the server. Client processes can upload file segments in parallel and then issue the COMBine command to assemble them on the server. The first argument, **<target>**, is the name of the target/destination file and the remaining arguments are the individual file segments (in order of assembly).

Format

```
COMB <target> <file-segment> [<file-segment>]<crLf>
```

Returns

250 - Command successful.

421 - The server is off-line or is going off-line.

450 - Requested action not taken, file unavailable. This occurs if a file segment is missing or cannot be accessed.

451 - Local error in processing. This is usually a result of an internal server error such as a thread not starting or memory allocation error.

501 - Syntax error in parameters or arguments. This usually results from an invalid or missing file segment.

530 - No user is currently authenticated on the command channel.

550 - Requested action not taken, file not found or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action.

552 - Requested file action aborted, exceeded user's storage allocation. This error usually occurs if there is not enough storage on the server to perform the action. It may also occur if the user has exceeded their allotted quota.

553 - File name not allowed, or file type banned.

CPSV

Overview

The **CPSV** command is used during secure FXP (site-to-site/server-to-server file transfers). The role of the CPSV command is to allow the FTP client to instruct either the source or destination server to initiate the handshaking for a secure transfer. During a SSL/TLS negotiation, one of the two entities initiates the secure handshake needed to establish the secure connection. Since an FXP occurs between two servers instead of a client and server, one of the two servers needs to assume the role of *client* in the handshaking process.

Issuing the CPSV command with no arguments will instruct the server to assume **PASV** mode for the data connection and also initiate the SSL/TLS handshake.

Format

CPSV<crlf>

Returns

- 227** - Entering Passive Mode (**x,x,x,x,x,x**).
- 421** - The server is off-line or is going off-line.
- 502** - SSL/TLS or PASV not enabled on this server.
- 530** - No user is currently authenticated on the command channel.

DQTA

Overview

The **DQTA** command provides disk quota information for the logged in user. Disk quotas values are measured in bytes.

Format

DQTA<crlf>

Returns

- 200** - Command successful. This is followed by the disk quota information.
For example:
200 Disk Quota:1402194 Bytes Used:1026
- 421** - The server is off-line or is going off-line.
- 305** - No user is currently authenticated on the command channel.
- 550** - Requested action not taken, file not found, or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action or if the file was not found.

EPRT

Overview

The **EPRT** command is used as part of the [IP v6 Extensions, RFC-2428](#). The EPRT command is an extension to the **PORT** command that provides the ability to specify an IP v6 address instead to the Titan FTP Server. Although Titan FTP Server does not currently support IP v6 addressing, it does support the ability to supply an IP v4 address using the EPRT command.

Format

EPRT | <net-protocol> | <net-address> | <tcp-port> |

net-protocol - The protocol version to use. **1=IP v4, 2=IP v6** (not currently supported).

net-address - A standard dotted IP address of the client to use for the connection.

tcp-port - TCP port of the client to use for the connection.

Returns

200 - The command was successful.

421 - The server is off-line or is going off-line.

501 - Syntax error in parameters or arguments. This usually results from invalid port information. It can also be sent back if the IP address is banned/disallowed from connecting to the server.

522 - Invalid protocol specified. Use **1**.

530 - No user is currently authenticated on the command channel.

EPSV

Overview

The **EPSV** command is used as part of the [IP v6 Extensions, RFC-2428](#). The EPSV command is an extension to the **PASV** command that provides the ability to have the server specify an IP v6 address back to the client. Although Titan FTP Server does not currently support IP v6 addressing, it does support the ability to retrieve an IP v4 address in IP v6 format using the EPSV command.

Format

EPSV<crLf>

Titan FTP Server will return the port number that the client can connect to for the data socket.

229 Entering Extended Passive Mode (|||<port>|)

Returns

229 Entering Extended Passive Mode (|||<port>|).

421 - The server is off-line or is going off-line.

421 - The server is off-line or is going off-line.

502 - Command not implemented. This response will be sent if the server is configured with PASV mode disabled.

522 - Invalid protocol specified. Use (1).

530 - No user is currently authenticated on the command channel.

FEAT

Overview

The **FEAT** command is used to dump a list of FTP extensions supported by Titan FTP Server. See [RFC 2389](#) for more information. A list of currently supported features are:

COMB
MLST/MLSD
UTF8
SIZE
MDTM
XCRC
REST
AUTH
CCSN/CPSV
EPRT/EPST
DQTA

Format

FEAT<crLf>

Returns

211 - Extensions Supported.

421 - The server is off-line or is going off-line.

MDTM

Overview

The **MDTM** command provides the ability to retrieve and set date/time (GMT) information for a file on the server. If a single argument is specified, it must be a file name for which date/time information is being requested. If two arguments are supplied, the first argument is the date/time to apply to the file, and the remaining argument is the file to apply that time to. When specifying the date/time information, it must be in the following format:

YYYYMMDD[HHXXSS][+-XXX]

- Y** - four digit year
- M** - two digit month
- D** - two digit day of the month
- H** - two digit hour (0..23)
- X** - two digit minute (0..59)
- S** - two digit second (0..59)

Format

MDTM <name><crLf> (returns the date/time for the file)

MDTM YYYYMMDD <name><crLf> (sets date for the file)

MDTM YYYYMMDDHHXXSS <name><crLf> (sets date/time for the file)

Returns

- 213** - Command successful.
- 421** - The server is off-line or is going off-line.
- 501** - Syntax error in parameters or arguments. This usually results from an invalid or missing file name.
- 502** - Command not implemented. This will be returned if MDTM support is not enabled on the server.
- 530** - No user is currently authenticated on the command channel.
- 553** - File name not allowed, or file type banned.

MFCT

Overview

The **MFCT** (Modify Fact: Creation Time) command provides the ability to modify the creation time (GMT) for a file on the server. When specifying the date/time information, it must be in the following format:

YYYYMMDDHHXXSS[+-XXX]

- Y** - four digit year
- M** - two digit month
- D** - two digit day of the month
- H** - two digit hour (0..23)
- X** - two digit minute (0..59)
- S** - two digit second (0..59)

Format

MFCT YYYYMMDDHHXXSS <name><CrLf> (sets the date/time for the file)

Returns

- 213** - Command successful.
- 421** - The server is off-line or is going off-line.
- 501** - Syntax error in parameters or arguments. This usually results from an invalid or missing file name.
- 502** - Command not implemented. This will be returned if MDTM support is not enabled on the server.
- 530** - No user is currently authenticated on the command channel.
- 553** - File name not allowed, or file type banned.

MFMT

Overview

The **MFMT** (Modify Fact:Modification Time) command provides the ability to modify the last modification time (GMT) for a file on the server. When specifying the date/time information, it must be in the following format:

YYYYMMDDHHXXSS

- Y** - four digit year
- M** - two digit month
- D** - two digit day of the month
- H** - two digit hour (0..23)
- X** - two digit minute (0..59)
- S** - two digit second (0..59)

Format

MFMT YYYYMMDDHHXXSS <name><CrLf> (sets the date/time for the file)

Returns

- 213** - Command successful.
- 421** - The server is off-line or is going off-line.
- 501** - Syntax error in parameters or arguments. This usually results from an invalid or missing file name.
- 502** - Command not implemented. This will be returned if MDTM support is not enabled on the server.
- 530** - No user is currently authenticated on the command channel.
- 553** - File name not allowed, or file type banned.

MLSD

Overview

The **MLSD** command provides data about the contents of the directory named as its argument. If no object is named, the current directory name is assumed. MLSD will send a multi-lined response, on the control connection, describing the contents of the directory. For more information on MLSD, see [IETF Draft - Extensions to FTP](#).

Format

MLSD [<name>]<crLf>

Returns

250 - Command successful. <directory contents follow>.

421 - The server is off-line or is going off-line.

501 - Syntax error in parameters or arguments. This usually results from an invalid or missing directory name.

530 - No user is currently authenticated on the command channel.

550 - Requested action not taken, directory not found, or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action or if the directory was not found.

MLST

Overview

The **MLST** command provides data about the object named as its argument. If no object is named, the current directory name is assumed. MLST will send a one line response, on the control connection that describes the object. For more information on MLST, see [IETF Draft - Extensions to FTP](#).

Format

MLST [<name>]<crLf>

Returns

250 - Command successful.

421 - The server is off-line or is going off-line.

501 - Syntax error in parameters or arguments. This usually results from an invalid or missing file name.

530 - No user is currently authenticated on the command channel.

550 - Requested action not taken, file not found, or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action or if the file was not found.

NLST

Overview

The **NLST** command causes a list of file names to be sent from the server to the client. If the **<name>** specifies a directory or other group of files, the server will transfer a list of file names in the specified directory. If the **<name>** specifies a file, then the server will send current file name information on the specified file. A null/missing **<name>** argument implies the user's current working directory or default directory. The data transfer is over the data connection in type ASCII so the user must ensure that the **TYPE** is appropriately set to ASCII. Currently supported **<args>** are:

- h** displays hidden files.
- a** does not include the '.' and '..' directories in the listing.
- F** adds file characterizations to the listing. Directories are terminated with a '/' and executable files are terminated with a '*'.

Format

NLST [-<args>] [<name>] <crLf>

Returns

150 - File status okay; about to open data connection. Client may begin receiving data over the data connection.

421 - The server is off-line or is going off-line.

451 - Local error in processing. This is usually a result of an internal server error such as a thread not starting or memory allocation error.

530 - No user is currently authenticated on the command channel. A client must be authenticated prior to executing this command.

550 - Requested action not taken, file not found, or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action, or if **<name>** was not found.

OPTS

Overview

The **OPTS** command provides the ability to set options on the server. The following OPTS are currently supported by Titan FTP Server:

UTF8 - Enables or Disables UTF8 encoding of file names.

Format

OPTS UTF8 <ON|OFF><crlf>

Returns

200 - Command successful.

421 - The server is off-line or is going off-line.

501 - Syntax error in parameters or arguments. This usually results from an invalid or missing file name.

502 - Command not implemented on this server.

PBSZ

Overview

The **PBSZ** command is an SSL/FTP Security Extension ([RFC 2228](#)) command used to specify the size of the protected buffer on an SSL connection.

Format

PBSZ <size><crLf>

Returns

200 - Command successful.

421 - The server is off-line or is going off-line.

501 - Syntax error in parameters or arguments. This usually results from an invalid or missing file name.

502 - SSL not enabled on this server.

503 - Bad sequence of commands. Must be used in conjunction with a secure SSL channel established during an SSL session.

530 - No user is currently authenticated on the command channel.

PROT

Overview

The **PROT** command is an SSL/FTP Security Extension ([RFC 2228](#)) command used to specify the protection level for an SSL data connection. The following protection levels are supported:

- C** - Clear
- S** - Safe
- E** - Confidential
- P** - Private

Note: Titan FTP Server maps **Safe**, **Confidential**, and **Private** into the same value. These values indicate that the data connection will be encrypted.

Format

PROT <protection-level><crlf>

Returns

- 200** - Command successful.
- 421** - The server is off-line or is going off-line.
- 501** - Syntax error in parameters or arguments. This usually results from an invalid or missing file name.
- 502** - SSL not enabled on this server.
- 503** - Bad sequence of commands. Must be used in conjunction with a secure SSL channel established during an SSL session.
- 530** - No user is currently authenticated on the command channel.

PSWD

Overview

The **PSWD** command, when used with the **SITE** command, allows the client user to modify the **user's password**. This feature is enabled on the server at the [user level](#).

Format

SITE PSWD <old-password> <new-password><crLf>

Returns

200 - Command successful; password changed.

421 - The server is off-line or is going off-line.

501 - Syntax error in parameters or arguments. This usually results from an invalid or missing file name.

530 - No user is currently authenticated on the command channel.

550 - Requested action not taken; invalid old password or invalid new password.

SITE ZONE

Overview

The **ZONE** command, when used with the **SITE** command, will display the time zone setting for the server. The time zone setting will be returned as plus/minus [+ -] number of minutes from UTC.

Format

SITE ZONE<crlf>

Returns

- 210** - UTC[+-]<minutes>
- 421** - The server is off-line or is going off-line.
- 530** - Not logged in.

SIZE

Overview

The **SIZE** command is used to return the size, in bytes, of the specified <name>. The <name> can either be a file name in the current working directory, a relative path and file name, or an absolute path and file name.

Format

SIZE <name><crlf>

Returns

- 213** - Command Successful.
- 421** - The server is off-line or is going off-line.
- 501** - Syntax error in parameters or arguments. This usually results from an invalid or missing file name.
- 530** - No user is currently authenticated on the command channel.
- 550** - Requested action not taken, file not found, or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action.

STRU

Overview

The **STRU** command is used to specify the structure of data on the server. Titan FTP Server implements this for compatibility purposes only and does not modify any data based on the current structure. Valid structure values are:

- F** - File Structure
- R** - Record Structure
- P** - Page Structure

The default structure is **File**.

Format

STRU <structure><crlf>

Returns

- 200** - The command completed successfully.
- 421** - The server is off-line or is going off-line.
- 504** - Command not implemented for that parameter.
- 530** - No user is currently authenticated on the command channel.

XCRC

Overview

The **XCRC** command causes the server to perform a **CRC-32 checksum** of the user supplied **<name>**. The **<name>** can either be a file name in the current working directory, a relative path and file name, or an absolute path and file name.

Format

XCRC <name><CrLf>

Returns

- 250** - XCRC XXXXXXXXX
Where **XXXXXXXX** is the 32-bit hex checksum for the file.
- 421** - The server is off-line or is going off-line.
- 501** - Syntax error in parameters or arguments. This usually results from an invalid or missing file name.
- 530** - No user is currently authenticated on the command channel.
- 550** - Requested action not taken, file not found, or no access. This error usually results if the client user process does not have appropriate access permissions to perform the action.

Tutorials

Creating a New Server

This tutorial is designed to give you step-by-step instructions for creating a new FTP server.

Choosing an IP address and Port number

Before you create a new **FTP server**, you must decide which **IP address** and **port number** your FTP server will use.

Most computers have a single IP address that can be accessed by other users. If you do not know the IP address of your computer, open a **command prompt** (DOS box) and type the command **IPCONFIG**. This command displays the IP address of your computer. This IP address can be used for your FTP server. You should also make sure that you have a **static** IP address for your computer.

NOTE: If you access the Internet through a dial-up account, you most likely have a **dynamic** IP address. If you have a **dynamic** IP address: during the setup process when you are prompted for the IP address, select **Any Available IP Address**.

For each IP address, there are many **ports** that can be used to access the computer. If you think of the IP address as your house, a port is similar to a door that can be used to gain access to your home. **TCP/IP** defines standard port numbers for various protocols. For example, when you connect to a Web site using your browser, you usually connect over **port 80**, which is the port reserved for **HTTP** access. For **FTP** access, the default port is **port 21**. If you are setting up an **FTP server**, you will usually use **port 21**.

Before you set up the **FTP server**, check if any other program is currently using **port 21** on your computer. To check which ports are being used on your computer, open a **command prompt** and enter the command:

```
netstat -a -n -p TCP
```

This command will dump out a list of IP addresses and ports that are currently in use on your computer. If **ipconfig** revealed that your IP address was **192.168.1.100**, then the **netstat** command may print information such as:

Proto	Local Address/Port	Foreign Address	State
TCP	192.168.1.100:80	0.0.0.0:0	Listening
TCP	192.168.1.100:990	0.0.0.0:0	Listening

Under the **Local Address/Port** column is a list of IP/ports that are currently in use.

If you see an entry that has **:21** after the IP address in the Local Address/Port column, then your default FTP port is currently being used by another application. If you do not see **:21**, then the FTP port is available for use. If **port 21** is currently in use, it may be that another FTP server is active on your computer. You can choose another port, such as **port 2100**, or you can make **port 21** available by closing the application that is using **port 21**.

There are over 32000 ports per IP address on your computer, and you can use any port that is available for your FTP Server. TCP/IP usually reserves **ports 1 through 1024** for special uses (such as **21** for **FTP** and **80** for **HTTP**), so if you do not use **port 21**, you should use a port number above **1024** for your FTP server.

Choose a location for your data

Your FTP server will **serve** files to users who connect using an **FTP client**. When users connect to your FTP server, they will usually want to download existing files or upload new files. The files that your FTP server serve are stored either on your local disk drive or on a network UNC that has been shared for you to use.

Typically, you will not want to provide users with the ability to access **all** of your files. You should choose a location that will house the files that you want users to be able to access (for example, an individual subdirectory). This directory, along with all subdirectories and files within those subdirectories, is known as the **namespace** for the FTP server. By default, Titan will create a base directory on your computer named **C:\srtFtpData**, which is the primary **namespace** where Titan will store all of the data for all FTP servers that you configure. If you create an FTP server name **MyFirstServer**, then Titan will create a directory named **C:\srtFtpData\MyFirstServer** that will be used as the **primary namespace** for all files accessible to users connecting to **MyFirstServer**.

Note: During the process of creating the new FTP server, you will have an opportunity to customize the directory name for the FTP server.

Once you have chosen an **IP address**, a **Port number**, and a **Data Directory** location for your new FTP server, you are ready to create your new server. You can use the Titan Server Administrator **New Server Wizard** to create your new server.

Launching the Server Wizard

Launch the **Titan Server Administrator** to create a new server. You can access the Administrator program by double-clicking on the **Administrator** icon in the Titan FTP Server **Program Group**.

Once the Administrator program is running, select **SERVER** then **NEW SERVER WIZARD** from the main menu bar to launch the Titan **New Server Wizard**.

Basic Server Information

- **Server Name** - A short name that uniquely identifies the Titan server on your system. This name will be displayed to the FTP user when they connect to the server. By default, this name will also be used as the name of the directory on your local computer where the FTP server data files will reside. For example, **BetaServer1** could be a name given to an FTP server that will be used to house beta versions of software.
- **Server Description** - A longer text string that can be used to enter descriptive information about the server. For example, **My Internal Beta Server used for test software** could be a description for the BetaServer1 server.
- **IP Address** - This drop-down list box will contain a list of all IP addresses that are currently registered on your computer. You can choose any of the IP addresses that are listed, or you can select **Any Available IP Address**, which instructs Titan to dynamically determine the IP address at runtime. You may want to select this option if you are using Titan in an environment (such as a dial-up account) where the IP address of your computer changes each time you restart Windows.

- **WAN Address** – Type the external domain name or IP address, for example, **myserver.com**.
- **Data Directory** - Specify the fully qualified path to the directory that will be used to store all of the files and folders accessible by users who connect to your server.
- **Log Directory** - Specify the fully qualified path to the directory where all of the server log files will be stored. This location should **not** be the same as the Data Directory because you do not usually want users to be able to gain access to your log files.
- **Start Server When FTP Service Starts** - Enable/select this option if you want this server to automatically start when the Titan FTP Service starts. The Titan Service usually starts when Windows boots, so enabling this feature ensures that this FTP server automatically starts and is available when Windows starts up. This is beneficial because if for some reason the computer loses power, or Windows needs to restart, your FTP server will automatically come back online.
- **Create Standard Unix Directories** - Enable this option if you would like Titan to automatically create the default Unix style directories of **/bin**, **/incoming**, **/pub**, and **/usr**. If this feature is enabled, each individual user account will have its own separate home directory under the **/usr** path, such as **/usr/user1/**, **/usr/user2/**.

Select Services

Use the check box to select the services/protocols that you will be using. Some options are only available in the Enterprise edition of Titan FTP Server. See the [Feature Comparison](#) for more information.

NOTE: You must enable FTP access if you are using FTPS with explicit SSL (also known as AUTH SSL).

User Authentication Options

Select the desired authentication method from the drop-down list of authentication methods that are available with your particular edition of Titan. For more information, see [User Authentication Options](#).

- **User Authentication Database** - Select the appropriate user authentication method for your server.
- **Authentication Server Setup** - If you are not using native Titan authentication, select **Authentication Server Setup** to launch the **User Authentication Wizard**. The authentication wizard will help you to configure Titan to work with your back end authentication server.
- **Auto Assign Home Directories** - Enable this feature if you would like Titan to automatically generate the user's home directory. If this feature is enabled, users will have their home directory created under the **/usr/** folder in the **Server Data Directory**.

FTP Services

Titan Server supports multiple protocols: FTP, FTPS and SFTP. The **FTP Services Wizard** allows you to configure the basic parameters necessary for running FTP on the server.

- **Enable FTP Services** - Enable this option to have Titan start the FTP server subsystems. If this option is disabled, FTP and FTPS will not be available on the server.
- **FTP Port** - Select the appropriate port for FTP. The default port is **port 21**.
- **Enable anonymous FTP access** - If this option is enabled, users will be able to connect to the Titan server using **anonymous** as the username.
- **This server is sitting behind a router** - Enable this option if Titan will be installed behind a firewall or router. This feature is very important. If it is not configured properly, users will still be able to connect to your server, but they may not be able to transfer files or view directory listings. For more information, see [Using Titan with a Router](#).
- **External WAN IP address of router** - If Titan is behind a router/firewall, enter the public/external IP address of the router. This will be used by Titan in the FTP **PASV** response. Titan will return the external IP address to the client so that the client can then open a data connection back to Titan by way of the router.
- **Use internal server IP in PASV response** - If Titan is behind a router/firewall, and you plan to have clients who are outside of the firewall and clients who are inside the firewall on your corporate LAN, then enable this feature so that local LAN clients can connect passively and receive the internal LAN IP from Titan in the **PASV** response.

SSL/FTPS Services

Some editions of Titan support FTPS, or FTP over an SSL secured connection.

- **Enable SSL/TLS access on this server** - Enable this option to have Titan start the FTPS subsystems. If this option is disabled, FTPS will not be available on the server.
- **Enable explicit SSL/TLS access (User connects using the AUTH SSL command)** – Enable this option to use explicit mode FTPS. When you use explicit mode, the FTP client will send an AUTH SSL or AUTH TLS command to the server if it intends to secure the connection.
- **Enable implicit SSL/TLS (User connects to a special port for SSL/TLS services)** – Enable this option to use implicit mode FTPS. When you use implicit mode, FTPS is initiated to the FTP server on a separate secure port, usually port 990. All traffic over that port is secured.
- **Implicit SSL/TLS port** – Use the up/down arrows to select the port. Port 990 is the default port for implicit SSL/TLS.
- **Use the following certificate for this server** – Use the drop-down arrow to select the certificate, or click **Certificate Management** to create or import a certificate. For more information about configuring FTPS/SSL in Titan, see the [Titan FTPS/SSL Quick Start Guide](#).

● **Enter the password associated with this certificate** - Type the **Password** for the selected certificate.

● **Certificate Store Folder**- This is the location where Titan will store all certificates for this server. **NOTE:** Local paths and UNC shares are supported; do not use a mapped drive because mapped network drives are not accessible from the Titan service.

For more information, see [Server FTPS Settings](#).

SSH/SFTP Services

Some editions of Titan support **SFTP**, a specialized subsystem of SSH.

● **Enable SFTP access on this server** - Enable this option to have Titan start the SFTP subsystems. If this option is disabled, SFTP will not be available on the server.

● **SFTP Port** - Port used for SFTP connections. The default SFTP port is port 22.

● **Use the following host key for this server** - Select an existing host key to be used by the server. If no host keys are available, use the **Host Key Management** utility to create a new server host key pair.

● **Host Key Management** - Launches the Titan **Host Key Management** utility that allows you to create, import, export, and manage SSH host keys used by Titan.

● **Host key password** - Enter the password used to secure the private key portion of the selected host key. Titan will not accept host key pairs that are not secured by a password. Passwords used to secure the private key portion of a host key pair must be at least 4 characters in length.

● **Host Key Folder** - Enter the fully qualified path where Titan SSH host keys will be stored. **NOTE:** Local paths and UNC shares are supported; do not use a mapped drive because mapped network drives are not accessible from the Titan service.

For more information, see [Server SFTP Settings](#).

HTTP/HTTPS Services

The **HTTP/HTTPS** dialog is used to configure Web services settings for this server.

● **Enable HTTP browser based interface to Titan FTP Server** - Select the check box to enable HTTP protocol on this server.

- **IP Address** - Use the drop-down arrow to select your IP address. Any Available IP Address indicates that the server will listen on all IP addresses that are configured on the computer, along with the local IP address of 127.0.0.0, also known as localhost.
- **Port** - Type in your port number. The default port is port 80.

Enable HTTPS/SSL browser based interface to Titan FTP Server - Select the check box to enable HTTPS protocol on this server.

- **IP Address** - Use the drop-down arrow to select your IP address. Any Available IP Address indicates that the server will listen on all IP addresses that are configured on the computer, along with the local IP address of 127.0.0.0, also known as localhost.
- **Port** - Type in your port number. The default port is port 443.

Use the following certificate for this server - Use the drop-down arrow to select the certificate to be used for this server.

Certificate Management - Launches the Certificate Manager, which can be used to create, import, and manage certificates.

Enter the password associated with this certificate - Type in the certificate password.

Web Services

Web Interface Module - Select the check box to enable access to this server using the Titan Web User Interface.

Note: The Titan FTP Server Web Interface is an optional module. Contact Sales@southrivertech.com for more information.

Mail Services

Some features of Titan, including the **Events Manager**, have the ability to send e-mail. This page will allow you to configure the base e-mail settings to be used with Titan. For more information on these settings, see [Email Server Settings](#).

Adjusting your Settings

When you have finished creating your server, you can adjust your settings using the Administrator program.

Using Titan with a Router or Firewall

Configuring Titan Server for use with a Router/Firewall—Overview

Most corporate and home networks today rely on a router and/or firewall to protect the internal computers or LAN (Local Area Network) from unauthorized access by outside users. The firewall is designed to block inbound TCP/IP access on any ports that are not designated as open. Opening a port means that TCP/IP traffic is permitted to travel inbound or outbound through the router. A port that is not open blocks traffic from traveling through that port.

Firewalls provide a high level of security by preventing most inbound traffic (but allowing most outbound traffic); and they will prevent any server that is installed on your internal LAN from being accessed by computers located on the Internet.

When installing a server on the internal LAN, additional steps are necessary so that users can gain proper access to the server without creating risk to other computers on the internal LAN. This is done by using port forwarding on the firewall to direct the TCP/IP traffic to the proper computer. Port forwarding is a feature that is used by most router/firewalls. When you are using port forwarding, data arriving on a specific port is redirected to the same port on a different computer.

When using Titan with a router, cable-modem, DSL modem, or firewall, you will probably need to configure Titan so that it knows about the IP address of the router. This is necessary so that when Titan goes in to **Passive mode**, it returns the proper global IP address to the FTP client.



If you would like more information about configuring Titan Server with a router/firewall, see the Configuring Titan FTP Server with a Router/Firewall [Quick Start Guide](#).

Event Handlers

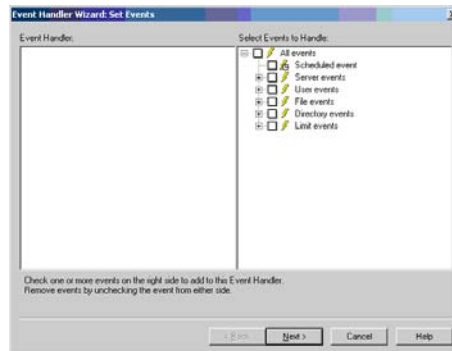
Event Handler Tutorial #1—Flag on Server Start

This tutorial is designed to introduce you to the **Event Handler** interface.

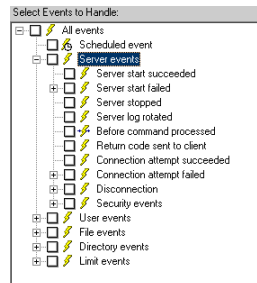
1. Create or select a **test server**. For this example, we will call our server **Test Server #1**.
2. In the Titan Administrator tree pane (left pane), expand the server and click **Events**.



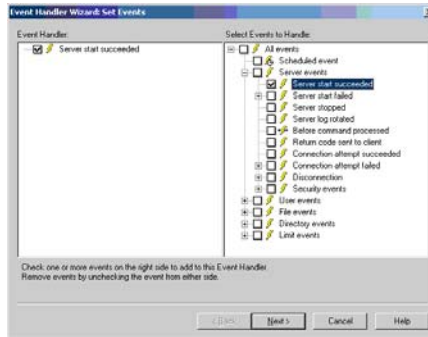
3. The **Event Handlers** tab appears. Click **Add** to launch the **Event Handler Wizard**.



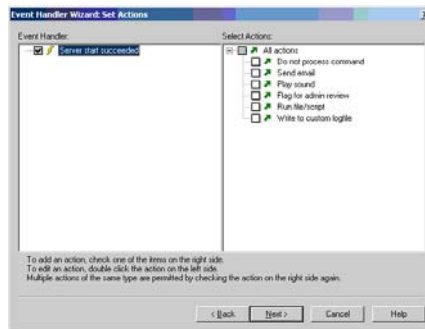
4. Select one or more events to handle. Events are categorized into a tree structure, with *parent* events at the base. Display specific *child* events by clicking "+" (plus sign) to expand a category. Expand the **Server events** category:



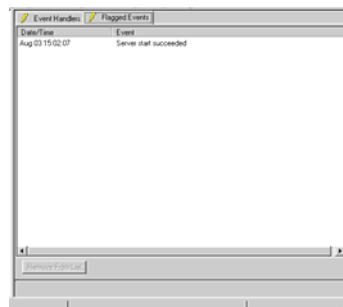
5. Select the **Server start succeeded** event. After you select it, the event appears on the left-hand side.



6. Click Next. You will now specify **event conditions**. Since there are no valid conditions for the **Server start succeeded** event, select **Next**.



7. Specify one or more **actions** that will happen if this event occurs with the necessary conditions. Select **Flag for admin review**. Select **Next** to view the Event Handler Summary.
8. Type unique name for this event handler and then click Finish.
9. It is time to test this Event Handler. Right-click on the **Server node** and select **Stop Server**. After the server has stopped, right-click on the **Server node** again and select **Start Server**. Select the **Events** node and then select the **Flagged Events** tab. You will now see a new flagged event in the list.



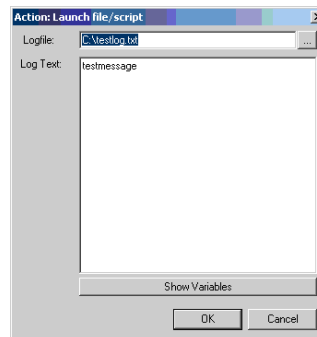
Event Handler Tutorial #2—Logging Every Event

This tutorial is designed to introduce you to some of the more complicated features of the Event Handler interface.

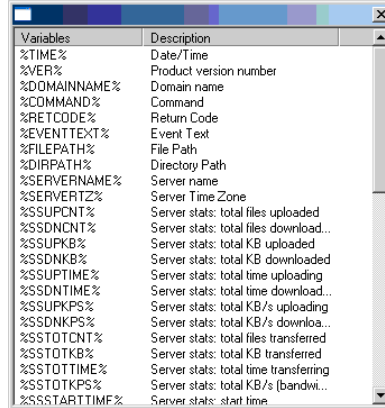
1. Create or select a **test server** (for this example, we will call our server **Test Server #1**).
2. Expand the server and click the **Events** node.



3. On the main tab view, the **Event Handlers** tab appears. Click **Add** to launch the **Event Handler Wizard**.
4. Select **All events**. You will notice that all the "children" events in the tree are automatically checked -- this is because the **All events** event encompasses *every* event that occurs on the server.
5. Select **Next**, and then select **Next** again to go to the **Set Actions** step of the wizard. Select **Write to custom logfile**. The Action dialog appears.



- In the **logfile** field, you can either specify an existing file, or create a new one. For this example we will set this field to **C:\testlog.txt**. In the **Log Text** field, you can specify a custom log message that will be written to this logfile. Clear this text box and then select **Show Variables** to see a list of status variables that can be written to the logfile.



Variables	Description
%TIME%	Date/Time
%VER%	Product version number
%DOMAINNAME%	Domain name
%COMMAND%	Command
%RETCODE%	Return Code
%EVENTTEXT%	Event Text
%FILEPATH%	File Path
%DIRPATH%	Directory Path
%SERVERNAME%	Server name
%SERVERTZ%	Server Time Zone
%SSUPCNT%	Server stats: total files uploaded
%SSDNCNT%	Server stats: total files download...
%SSUPKB%	Server stats: total KB uploaded
%SSDNKB%	Server stats: total KB downloaded
%SSUPTIME%	Server stats: total time uploading
%SSDNTIME%	Server stats: total time download...
%SSUPKPS%	Server stats: total KB/s upload...
%SSDNKPS%	Server stats: total KB/s downloa...
%SSTOTCNT%	Server stats: total files transferred
%SSTOTKB%	Server stats: total KB transferred
%SSTOTTIME%	Server stats: total time transferring
%SSTOTKPS%	Server stats: total KB/s (bandwi...
%SSSTARTTIME%	Server stats: start time

By entering these variables into the **Log Text** field, you can create a custom log message based on the current state of the server. This functionality is very similar to the custom messages found under the **Connections->Messages** tab.

- For our example, we will set up a simple custom message. Click a variable in the **Variables dialog** to insert that variable into the last text box in which you were typing. If you were previously typing in the **Log Text** field, click **%TIME%**. You will see that this variable will appear wherever the cursor was in the Log Text field.



You can also type the variables instead of selecting them.

- We will continue to edit the **Log Text** field until it is set to:
time:%TIME%, event:%EVENTTEXT%
- Select **OK**. You are returned to the **Set Actions** page of the wizard. The new **Logfile Action** appears on the left-hand side.

If you would like to edit the Action, double-click on the checked action on the left. To remove an Action, clear the box on the left-hand side. You can add any number of Actions to an Event Handler.

10. To practice, create another logfile that writes to **C:\testlog2.txt** with a slightly different message.

11. When you are finished, create a unique name for this Event Handler. Click **Finish**. The event handler will appear in the Event Handler list.

12. It is now time to test this Event Handler. First stop and then start the server (right-click the server and then select **Stop Server**, and then select **Start Server**). You can open Explorer to see the logfiles you specified. Make sure that the **%** variables resolved correctly to their real-time values. The **C:\testlog.txt** file should read something like this:

time:2005-08-04 10:49:14 , event:Server stopped

time:2005-08-04 10:49:14 , event:Server start succeeded

Event Handler Tutorial #3—Single Use Account

This tutorial will introduce you to some of the more complicated features of the **Event Handler** interface.

It is sometimes necessary to create temporary accounts. Previously, one way to do this was to create an account and set the expiration date so that the account would only work for a set number of days. However, it might also be necessary to create a temporary account that is disabled after a single use.

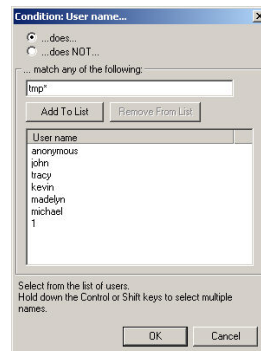
1. First, create a test account, which you will disable. Click **Add** to create a new Event Handler.
2. Under the **Set Events** page, select **User login attempt successful** (under **User Events**)
3. Under the **Conditions** page, select **User login attempt successful** (on the left-hand side) and then select the condition **User name**. When the **User name in list** dialog appears, select the **temporary account** from the list and click **OK**.
4. Under the **Actions** page, select **Disable user account**. Continue through the wizard until the event handler is created.
5. To test the Event Handler, log on as the temporary user. Now log out and try again. The first login attempt should succeed while the second will fail.

A more organized approach

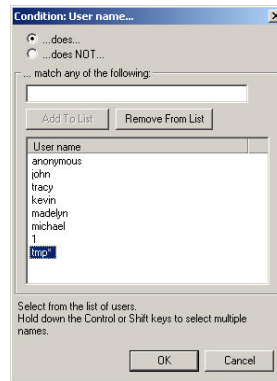
In the previous example, the Event Handler will only work for a single user account. This solution is not very practical if you are going to be constantly creating temporary accounts, especially if this process is automated by way of the **srxCFG** Utility or **srxCOM** interface.

Instead, you can use wildcards inside the **username filter** so that any account that begins with a **tmp** in the username will be considered a temporary (single use) account.

1. Create a test user called **tmp1**.
2. Create an Event Handler that handles the **User login attempt successful** event.
3. On the **Set Conditions** page, select the **User name** condition. Instead of selecting one of the user names in the list, type **tmp*** into the text field near the top.



4. Select **Add To List** and then select the **tmp*** entry in the list.

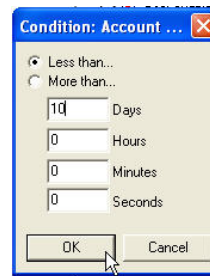


5. Select **OK** and then finish creating the Event Handler.
6. Test the Event Handler by logging in using the **tmp1** user. Verify that the account correctly disables, and then try testing with a new account (for example, **tmp2**).

Event Handler Tutorial #4—Email When Account Expiring

This tutorial shows you how to create an Event Handler that will automatically send an e-mail when a user logs in and the user's account has less than 10 days until it expires.

1. Launch the **Titan Administrator**. Expand the server in the tree pane (left-hand pane) and click **Events**.
2. Under the **Events Handler** tab, click **Add**.
3. Expand **User Events**, select **User login attempt successful**, and then click **Next**.
4. Select the Condition **User account expiration date**. Select **Less than** and type **10** in the **Days** text box. Click **OK**, and then click **Next**.



5. Select the Action **Send Email** and then click **Next**. Type your information in the e-mail window and then click **OK**.
6. Click **Next**. Type a unique name for this Event Handler. Click **Finish**. This Event Handler now appears under the Event Handler tab.

Event Handler Tutorial #5—User Logs

This simple Event Handler allows you to create a custom log for each user account that will access the server. Every event that is caused by this user will cause a custom log message to be written to that user's log file.

1. Launch the **Titan Administrator**. Expand the server in the tree pane (left-hand pane) and click **Events**.
2. Under the **Events Handler** tab, click **Add**.
3. Expand **User Events**, select **All Events**, and then click **Next**.
4. Select the Action **Write to custom logfile**. In the Logfile Name field, type **C:\srFtpLogs\%SERVER%\%USERNAME%.log**. In the Log Text field type **time:%TIME%, event:%EVENTTEXT%**. Click **OK**.
5. Click **Next**. Type a unique name for this Event Handler. Click **Finish**. This Event Handler now appears under the Event Handler tab.

Log Example: On a server where there are two accounts, "bob" and "anonymous", the following logs would be created:

C:\srFtpLogs\testserver\bob.log

C:\srFtpLogs\testserver\anonymous.log

Note: Make sure that the directory exists where you want the log file to be created; otherwise, the file will not be created. In this example, the log file is stored in the same directory as the default location for the server log.

Event Handler Tutorial #6—Move File After Upload

This tutorial shows you how to move a file immediately after it has been uploaded. This is useful under circumstances where a file will be uploaded to a certain directory and then moved to a different folder, hard drive, or network connection that is not directly accessible by way of the FTP server.

In this example, we have our FTP server set up to expose **C:\server** to public access; and the **C:\private** directory is not accessible by way of FTP.

1. Launch the **Titan Administrator**. Expand the server in the tree pane (left-hand pane) and click **Events**.
2. Under the **Events Handler** tab, click **Add**.
3. Expand **File Events**, select **File upload/write: Upload/Write successful**, and then click **Next**.
4. Select the Condition **Filename**. In the pop-up window select **Does** (match) and then type **C:\Server***. Click **Add to List** and then click **OK**.
5. Select the Action **Run file/script**. In the file/script field, type **C:\move.bat**. Type the parameter **%FILEPATH%** and then click **Add**. Type the parameter **C:\private\%FILENAME%** and then click **Add** again. When you are finished, click **OK**. The Event Handler appears in the left-hand pane.
6. Click **Next**. Type a unique name for this Event Handler. Click **Finish**. This Event Handler now appears under the Event Handler tab.

NOTE: Always wrap a parameter inside **double quotes** if that parameter does or could contain a space. Otherwise, the command line parser will split a single parameter into several space-delimited parameters.

The **C:\move.bat** file should look like this:

```
move %1 %2
```

Or, if your system does not have a move command:

```
copy %1 %2
```

```
del %1
```

Troubleshooting

FAQ - Frequently Asked Questions

How do I Backup and Restore a Titan Server?

Using **Regedit** - Export the Server's directory under **hkey_localmachine\software\south river technologies\server\Titan FTP Server**. Save the %.reg file and copy it to your computer (or a new computer). Double-click on the *.reg file that you copied (or while on the new machine). This will restore all of your servers. You need to restart the new machine for everything to take effect properly. Once you have restarted the new machine, launch your Titan Administrator, and make sure that the IP settings for each server have been changed to the new server address.

Can I configure the FTP server to listen for standard FTP and explicit SSL on multiple ports?

Titan can have FTP listening on one port and can have SSL/FTPS listening on one other port. If you want to have SSL/FTPS listening on multiple ports, then you must configure another FTP server under the domain and set up the other SSL/FTPS ports.

Please explain why, when a user is created, Titan automatically gives them full access to the home directory, but when you delete the access from the user, it puts it back?

When you create a user, it will give them full rights to their home directory. If you would like to change/limit access to their home directory, then you can alter the **Directory Access Rule** for the user's home directory. If you delete the rule, Titan will restore it to the default permissions, so if you want the user to have no access to their home directory, then you need to leave an empty rule in the list.

How do I start the Titan service from the command prompt?

At the command prompt, type in **srxtitan /start**.

Statistics Tracking--How do you create reports with this information?

Titan will store the statistics information in an ODBC database. You can use any generic ODBC reporting utility, such as Crystal Reports, to access and format that information. **NOTE:** SRT supports Titan configurations using SQL Server 2005 or later or SQL Server 2005 Express or later, test or production environment. No other databases are supported.

Why doesn't the FTP server that I configured start automatically when the computer is booting up?

- Make sure that you have checked the box **Start Titan FTP Server Service** when Windows boots in your Titan **Domain** Configuration.
- Make sure you have checked the box **Start this server when Titan FTP Server Service** starts in your Titan **Server** Configuration.
- Make sure Titan FTP Server Daemon service is set to **Automatic** so the service starts automatically under your Windows service administration.
- Please check the Titan log files for any errors. The log files will display an error if the FTP server is unable to start.
- It may be you may have a NIC card that does not come online with a valid IP address before the Titan service starts, or you could have another service that is trying to use port 21.
- You may have to change the binding order of your NICs.
- To check if other services may be using port 21: run a **netstat -a -n -p tcb -b** from a command line interface before you start your Titan servers and check to see if another program may be using FTP port 21.
- Make sure the FTP Publishing Service is set to automatic or turned on, as it will affect Titan. (Microsoft updates may automatically install this service.)

Can I use IE7 with Titan?

If you are using IE7 and need to access a Titan Server, you should consider using the **Lock User In Home Directory** feature. This is because IE7 will issue a **CWD/** immediately after connecting to the FTP Server. If **Lock User In Home Directory** is not enabled, a **CWD/** command usually results in Titan trying to do a **CWD c:\srtFtpData** which, for security reasons, is locked down and people do not have any access to it. If you are not able to use the **Lock User in Home Directory feature**, for whatever reason, you need to change your root directory level Directory Access to at least **View Directory/List** for your IE7 users, or they will have problems accessing your Titan Server.

Does Titan FTP provide an API?

Yes, there is a COM interface called srxCOM and there is a command line utility called srxCFG.exe. See the [srxCOM](#) and [srxCFG](#) topics in this guide for more information.

Why do I receive the error “Directory Listing not found” or “Disconnected from server: ECONNABORTED - Connection aborted” when using Filezilla 3.1.0.1 and Auth/TLS?

Due to an incompatibility in the SSL subsystems when using FileZilla 3.1.0.1 for Auth/TLS communications, you may receive the following error:

“Directory Listing not found or Disconnected from server: ECONNABORTED - Connection aborted”

You will need to upgrade to Titan version 6.24 or later.

As a workaround, you can use any version of FileZilla previous to version 3.1.0.1 or most other FTP clients until you upgrade.

Why do I receive an error upon startup of TITAN Administrator stating that IP may be incorrect or port may be in use?

Java has made a recent change that takes over port 31000. Please run a **netstat -a -n -p tcp -b**. If the Java service (or any other than Titan) is running on port 31000, you will need to change your Administration port in Titan to a non-used port (i.e., 31010 or other not well-known port that is not currently in use).

Why do I receive the error code “425 Cannot open data connection”?

This error indicates that the FTP client is running in Active/PORT mode and you have a firewall in front of your client PC.

You should contact the user and have them configure their FTP client to run in PASV/Passive mode instead.

How can I upgrade a TITAN SERVER to a new box?

If you are moving to a new box with the same operating system, you should follow this procedure:

1. Using Regedit: Export the Servers directory under `hkeylocalmachine\software\south river technologies\server`. Save the `%.reg` file and copy it to your new machine.
2. On the new box, you will install Titan, then import the registry settings from your old Titan Server Settings from the `%.reg` file.

If you are moving the server from an old Win2k operating system box to a new Win2k3 operating system box, you may have problems activating the license and will have to perform an upgrade. This upgrade would be available at a discounted price. Please let us know if this is the case and we will put you in contact with one of our sales professionals to get a newer version license so you can upgrade the software.

How do I configure Public Key authentication with Titan?

If you plan to use public key authentication with Titan, then Titan must have a copy of each client's public key before the client can connect. To do this, run the Titan Admin utility and then drill down until you find the Username who will be using public key authentication with the server. Click the **SFTP** tab for this user and then use the Host Key Management utility to Import the public key for that user. Once the key is imported, return to the main user's SFTP tab and select their public key from the list of public keys in the drop-down list box. This will assign that public host key to the user.

Also, on the main SFTP tab for the Server, make sure you have enabled the Allow Trusted Host Keys When Accessing this Server option. This option tells Titan to send public key as an authentication type (along with password). If you enable the Require Trusted Host Keys option, then Titan will only send Public Key as the authentication method.

For more information about how to use public key authentication with Titan, see the [Titan SFTP quick start guide](#).

Why do I receive the "Error 1610" message while importing SSHKEYGEN host keys in Titan for SFTP?

If you receive the following error when you are trying to import SSH-KEYGEN host key pairs into Titan Server :

"Unable to import host key due to invalid format or bad password. Make sure the SSH key is OpenSSH format (Error 1610)", there are two options to fix this issue. This problem most commonly occurs when you are using Linux.

Option #1, Performed on the client:

1. Download Puttygen, available for download at <http://www.putty.nl/download.html>
2. Run Puttygen. Select Conversions>Import Key.
3. Select the Private Key and click Open.
4. Type the password for the Public Key and click Save Public Key.
5. Send the public key file to the Server Administrator to import into Titan Server.

Option #2, Performed on the client:

If you have created an OpenSSH key pair with the ssh-keygen command, you can use the following command to create a usable public key:

```
ssh-keygen -e -f <private_key_name> > <public_key_name>
```

For example:

```
ssh-keygen -e -f $HOME/.ssh/id_dsa > $HOME/.ssh/SSH_dsa.pub
```

This command will read the private key and generate a public key that can be used by Titan Server.

What should I do if I receive the "Error 1610" message while trying to import Public Keys created by FTP Clients that are not SSHKEYGEN created?

1. Generate Key(s) in Putty Keygen.
2. Type in password for key and change conversion option to Export ssh.com key.
Note: you must add the *.pub extension to the file name.
3. Export the private key from puttykeygen (note that you do not have to change file extensions on this file).
4. You should now have two key files. Import the *.pub key into the Titan user's account via the SFTP tab under the user's configuration options.
Note: You must import the public key into Titan and replace the existing putty key. You must replace the older putty generated public key with the one you are importing into Titan, so click Yes when prompted.
5. Click Close once imported, then attach the public key to the user via the SFTP tab on the user's configuration.
6. Open FTP client software and import the private key into user's configuration.
7. Browse to the private keyfile, type the keyfile password, and click OK.

To test this, make sure that the Titan server is configured to **Require trusted host keys** when accessing this server at the server level. Have your test user try to connect using SFTP. For more information about using Titan with Host Keys, see the [Titan Host Key quick start guide](#).

How do I configure an SFTP Server and create a HOSTKEY Pair in Titan?

You can run the standard Titan FTP Server wizard to configure a standard server. Once the server has been created, click the SFTP/SSH tab for the server and enable SFTP/SSH. You will then create a hostkey pair for use by the server. Once this has been completed, you must open port 22 for standard SFTP/SSH.

HOW TO CREATE YOUR HOSTKEY PAIR:

On your server, you must create a Host Key Pair that will be used/assigned to the SFTP Server. Use the Host Key Management utility in the Titan Admin console to generate the key pair and to assign it to the Server. You might not need to send it to the client; however, you can if you want to. Just Export the Public Key and send that .pub file to the client.

If the client intends to use Public Key Authentication instead of the default Password Authentication, you must configure Titan for Public Key Authentication and the SFTP/SSH client Administrator must export the client's Public Host Key and send it to you so that you can import it into Titan.

This entire process is outlined in detail in our [SFTP/Host Key Quick Start Guide](#) on our Web site.

Troubleshooting - set logging level to debug

If you are experiencing problems with your server, please set your logging level to Debug (change your rotation schedule to Daily so the log files won't grow too large) and send a copy of your log files attached to a new support ticket. This will allow SRT to troubleshoot your support ticket more quickly and efficiently.

What characters can't be used in a file name?

You can't use any of the following characters in a file name: \ / ? : * " > < |

SRT Knowledgebase

Visit our [Knowledgebase](#) to read help desk articles and answers to frequently asked questions.

Reporting Problems

To report a problem, visit the Titan support site at <http://www.SouthRiverTechnologies.com/support/>.

Please furnish our Support Engineers with the following information:

- The Windows platform you are running.
- The Titan Server version that you are using.
- Client used and version.
- A detailed description of the problem. Include file name and complete sub-directory name, if applicable.
- Attach a copy of the log file to your e-mail.

Contact Information

South River Technologies is located in Annapolis, Maryland.

Headquarters:	South River Technologies 2635 Riva Road Suite 100 Annapolis, MD 21401 U.S.A.
Telephone:	1-410-266-0667
Fax:	1-410-266-1191
World Wide Web:	www.southrivertech.com
Office hours:	Monday to Friday 8:30 A.M. to 5:30 P.M. Eastern Time, GMT-5:00
Sales Telephone:	1-410-266-0667
Sales Fax:	1-410-266-1191
Sales Email:	sales@southrivertech.com
Online Support:	http://www.srthelpdesk.com
Support Email:	support@southrivertech.com

Compare Feature Sets

Features	Small Business Edition	Professional Edition	Enterprise Edition
Maximum number of user accounts	25	200	Unlimited
Maximum number of concurrent connections	25	200	Unlimited
Maximum number of server configurations (on one physical PC)	1	2	Unlimited
Remote Administration			Yes
Windows NT/SAM Authentication			Yes
FTPS (FTP over SSL/TLS)		Yes	Yes
SFTP (SSH's Secure File Transfer Protocol)			Yes
Event Management			Yes
Web-based User Interface			Yes (\$)

Index

A

ABOR, 190
Actions, 107
Active Directory, 249
ADDGROUP, 145
Adding
 Users, 57
Adding, 57
ADDSERVER, 138
ADDUSER, 151
Administrator
 Launching, 11
Administrator, 11
Advanced Connections Settings, 29, 63, 83
APPE, 191
AUTH, 213

B

BANUSER, 155
Before Events, 108

C

CCSN, 214
CDUP, 192
characters, 249
COMB, 215
Command Overview, 187
Conditions, 105
Contact Information, 254
CPSV, 216
CRC File Integrity Checking, 123
Creating
 Groups, 56
 New, 18, 56
 New Groups, 56
 New Server, 232
 New Servers, 18
 Users, 76
Creating, 18
Creating, 56
Creating, 56
Creating, 76
Creating, 232
Custom Message Variables, 119
Custom Messages, 33, 67, 87, 119
CWD, 193

D

DELE, 194
Deleting
 Groups, 57
 Servers, 19
 Users, 77
Deleting, 19
Deleting, 57
Deleting, 77
DELGROUP, 146
DELSERVER, 139
DELUSER, 152

Directory Access Settings, 36, 70, 90
Domain Overview, 13
DQTA, 216

E

EPRT, 217
EPSV, 218
Error 1610, 249
Event Handler Tutorial, 239, 241, 244, 246,
 247, 248
Event Handlers, 54, 239, 241, 244, 246, 247,
 248
Event Handling
 Introduction, 97
Event Handling, 97
Events, 54, 55, 97, 98, 108, 239, 241, 244,
 246, 247, 248

F

FEAT, 219
file name, 249
Files/Directories Settings, 34, 68, 88
Firewall, 238
Flagged Events, 55
FTP Return Codes, 184

G

General Connections Settings, 28, 62, 82
General Group Settings, 60
General Server Settings, 20
General User Settings, 79
GETGATTR, 147
GETSATTR, 140
GETUATTR, 153
Group Configuration Attributes, 144, 171
Group Users Settings, 61
Groups
 Creating, 56
 Deleting, 57
Groups, 9, 56
Groups, 56
Groups, 57
Groups, 57
Groups, 58
Groups, 60
Groups, 61
Groups, 81
Groups, 144
Groups, 171
Groups Settings, 81
GRP_Create, 172
GRP_Delete, 172
GRP_Enum, 173
GRP_GetAttr, 176
GRP_GetMembers, 174
GRP_SetAttr, 177
GRP_SetMembers, 175

H

HELP, 195

HOSTKEY Pair, 249

I

IE7, 249
 Inherited Settings, 113
 Installation, 7
 Introduction
 Event Handling, 97
 Introduction, 97
 IP Access Settings, 31, 65, 85

K

KICKSESS, 156
 KICKUSER, 156

L

Launching
 Administrator, 11
 Launching, 11
 LIST, 196
 LISTGROUPS, 149
 LISTSERVERS, 142
 LISTUSERS, 155

M

MDTM, 220
 MKD, 197
 MLSD, 223
 MLST, 223
 MODE, 198

N

NAT, 249
 New
 Creating, 18, 56
 New, 18
 New, 56
 New, 232
 New Groups
 Creating, 56
 New Groups, 56
 New Server
 Creating, 18, 232
 New Server, 18
 New Server, 232
 NLST, 224
 NOOP, 198

O

OPTS, 225
 Overview, 157, 188

P

PASS, 199
 PASV, 200
 PBSZ, 226
 PORT, 201
 Program Group Items, 9
 PROT, 227
 PSWD, 228

Public Key, 249
 PWD, 202

Q

QUIT, 202

R

REIN, 203
 Remote Administration, 125
 Removing
 Users, 58
 Removing, 58
 REST, 203
 RESTARTSERVER, 143
 RETR, 204
 RMD, 205
 RNFR, 205
 RNT0, 206
 Router, 238

S

Server Activity, 51
 Server Advanced Settings, 21
 Server Configuration Attributes, 132, 160
 Server Disk Quotas, 40, 74, 94
 Server FTP Settings, 26
 Server FTPS/SSL Settings, 41
 Server Log, 46, 47
 Server Log Settings, 47
 Server SFTPS Settings, 43
 Servers
 Deleting, 19
 Servers, 17, 18
 Servers, 19
 Servers, 20
 Servers, 21
 Servers, 26
 Servers, 40
 Servers, 41
 Servers, 43
 Servers, 46
 Servers, 47
 Servers, 51
 Servers, 74
 Servers, 94
 Servers, 132
 Servers, 160
 Servers Overview, 17
 SETGATTR, 148
 SETSATTR, 141
 SETUATTR, 154
 SFTP, 96, 127, 187, 188, 249
 SFTP Commands Overview, 187
 SFTP Return Codes Overview, 188
 SFTP Support, 127
 SFTP tab, 96
 Shared Attributes, 114
 SITE, 207, 229
 SITE_ZONE, 229
 SIZE, 229
 Spy Session, 53
 Spy User, 52
 SRX_Connect, 158
 SRX_Disconnect, 159

SRX_GetErrStr, 159
 SRXCFG Command Line Utility, 130
 SSHKEYGEN, 249
 SSL Support, 126
 Starting
 Titan Service, 8
 Starting, 8
 STARTSERVER, 142
 STAT, 208
 Statistics Settings, 49
 STOPSERVER, 143
 STOR, 209
 STOU, 210
 STRU, 230
 SVR_Create, 166
 SVR_Delete, 167
 SVR_Enum, 167
 SVR_GetAttr, 169
 SVR_Restart, 169
 SVR_SetAttr, 170
 SVR_Start, 168
 SVR_Stop, 168
 SYST, 211
 System Requirements, 6

T

Terminology, 10
 Titan Service
 Starting, 8
 Titan Service, 8
 TYPE, 211

U

upgrade, 249

Upload/Download Ratios, 32, 66, 86
 USER, 212
 User Authentication Basics, 76
 User Configuration Attributes, 150, 178
 User Groups Settings, 81
 Users
 Adding, 57
 Creating, 76
 Deleting, 77
 Removing, 58
 Users, 57
 Users, 58
 Users, 61
 Users, 76
 Users, 76
 Users, 77
 Users, 79
 Users, 81
 Users, 150
 Users, 178
 Users Settings, 61
 Using Titan, 238
 USR_Create, 179
 USR_Delete, 180
 USR_Enum, 180
 USR_GetAttr, 181
 USR_SetAttr, 181

V

Virtual Folders, 128

X

XCRC, 231