



# Wing FTP Server

## User Guide

© 2021 Wing FTP Server

# Contents

<b>Chapter 1</b>	<b>1 Wing FTP Server</b>
	1 Overview
	3 Editions
	4 Requirements
	5 Purchase
<b>Chapter 2</b>	<b>6 Quick Start</b>
	6 Quick Start Guide
	12 Router & Firewall
	14 Email Notification
	16 Administration Console
<b>Chapter 3</b>	<b>20 Administration</b>
	20 Settings
	20 General Settings
	21 Listener
	22 IP Access
	23 Admin User
	26 Admin Log
<b>Chapter 4</b>	<b>28 Server</b>
	28 Logs & Status
	28 Server Status
	30 Server Log
	31 Settings
	31 General Settings
	39 SSL Certificate Manager
	46 SSH Host Key Manager
	47 SMTP Server Manager
	49 Gateway Manager
	53 Task Scheduler
<b>Chapter 5</b>	<b>55 Domain</b>
	55 Logs & Status
	55 Domain Status
	57 Domain Log
	57 Audit & Report
	62 Temp Ban
	62 Activity

# Contents

	63	Graphs
	64	Domain Settings
	64	General Settings
	72	Authentication Settings
	82	IP Access
	83	IP Whitelisting
	84	File Access
	85	SFV Directory
	87	Transfer Quota Limit
	88	FTP Messages
	89	Listeners
	89	Users
	89	User > General
	91	User > Directory
	92	User > Subfolder
	92	User > Group
	94	User > Limit
	95	User > Ratio & Quota
	96	User > IP Access
	98	User > File Access
	99	User > Access Time
	99	Event Manager
<b>Chapter 6</b>	<b>105</b>	<b>Advanced Features</b>
	105	Lua Language
	107	Server Variables
	108	Server Lua API
	109	User & Group
	116	Domain Function
	130	System Function
	140	Get/Set Options
	143	Miscellaneous
	147	Administrator
	150	Database Schema

## Overview

## Welcome to Wing FTP Server

Wing FTP Server is an enterprise FTP server software that can be used on **Windows, Linux, and Mac OS**. It supports multiple file transfer protocols, including **FTP, FTPS, HTTP, HTTPS, and SFTP**, giving your clients flexibility in how they connect to the server. And it provides administrators with a web-based interface to administrate the server from anywhere. You can also monitor server performance and online sessions and even receive email notifications about various events taking place on the server.

The major features of Wing FTP Server:

- **Free Edition** - After the trial period, you can continue to use Wing FTP Server as a Free edition for non-commercial use.
- **Cross-Platform** - You can run this FTP server software on Windows, Linux, and Mac OS.
- **Multiple Protocols** - Supports FTP, FTPS, SFTP and HTTP/S web client. Free client [FTP Rush](#) for FTP/SFTP file transfer.
- **Web-Based Administration** - Just with a web browser, you can administrate your FTP server anytime, anywhere.
- **Web-Based Client** - Transfer files via a web browser anytime, anywhere. Also support online text editor, picture viewer, and video player.
- **Android & iOS App** - This [Android & iOS App](#) helps mobile phone user transfer and manage remote FTP files easily.
- **Programmable Event Manager** - Use the Event Manager to execute Lua script, send an email notification or run 3rd party program.
- **Database & LDAP** - Store user data via XML files or ODBC/Mysql database, also support Windows AD or LDAP authentication.
- **Load Balance** - Wing FTP Server can be clustered by [Wing Gateway](#) to meet load balance (or high availability) needs.

- **Multiple Domains** - You can run multiple virtual servers by adding multiple domains (each separated by different ports or IP addresses).
- **Task Scheduler** - Use your Lua scripting tasks to appoint an action at a specified time.
- **Audit & Report** - All the FTP transactions will be captured into a database, you can analyze it and generate reports in real-time.
- **Virtual Directories** - Allows you to map virtual directories to physical directories, and you can also use UNC path or mapped drives for virtual directories.
- **Ratio, disk quota system** - Every user can be assigned individual disk quota, ratio, and bandwidth.
- **IPv6 Support** - You can easily add an IPv6-based listener (or IPv6-based IP access rules).
- **FIPS 140-2** - Uses OpenSSL FIPS 140-2 validated cryptographic module.
- **Two-factor authentication** - You can enable two-factor authentication (TOTP) for Web Client, it will add an extra layer of security to user account.
- **Support 15 languages** - English, Français, Deutsch, Italiano, Türkçe, Český, Româna, Nederlands, Português, Español, 中文, 繁體中文, 日本語, Polski.

Web Client contains basic functions like Windows Explorer and also brings the below excellent features to you:

**Multi-file Uploader** - You can upload multiple files or an entire folder in Web Client easily, also supports HTML5 drag&drop operation.

**Multi-file Downloader** - Using a Chrome (or Firefox) extension "Wing Download Manager", for downloading multiple files/folders in Web Client.

**Online Text Editor** - This function enables you to view or modify some text file without downloading it.

**Search Files** - Users can recursively search for files using exact filename matches or wildcard

matches.

**Built-in Media Player** - This feature helps you to play video files (.MP4, .MOV) and audio files (.MP3, .OGG, .WAV) online.

**Picture Viewer** - You can simply view PNG, JPG, GIF, BMP images in the picture viewer.

**Support for iOS/Android** - The mobile version of Web Client has a simpler interface for iOS or Android's browser.

**WebLink** - Share the files or folders with just a link, you can also set a limited number of downloads or download password.

**File Request Link** - Request files from your client with just a link, you can also set the upload password or expiration date for the link.

**Zip/Unzip** - Creating zip archives or extracting zip archives on the server directly, save your download/upload time.

For more information, please visit our official website:

<https://www.wftpserver.com/>

## Editions

Wing FTP Server is available in 4 editions: Free Edition, Standard Edition, Secure Edition, and Corporate Edition.

You can download the installer file from our [official website](#). If you never install WingFTP before, then you can have a 30-day trial period with full functions. After 30 days, it will become Free Edition automatically if you never register.

Please note, Free Edition is only for **non-commercial use** (personal use), and the other three editions can get FREE email and remote assistance support for a lifetime.

If you are not sure which edition is right for you, please check the editions comparison chart below:

	Free Edition	Standard Edition	Secure Edition	Corporate Edition
Plain FTP (no encryption)	✓	✓	✓	✓
Web Client (HTTP)	✓	✓	✓	✓
Web based administration	✓	✓	✓	✓
iOS & Android App	✓	✓	✓	✓
Max domain number	1	Unlimited	Unlimited	Unlimited
Max user accounts per domain	10	100	200	Unlimited
Max concurrent connections per domain	10	100	200	Unlimited
Virtual directory mapping	✓	✓	✓	✓
Audit & Report	✓	✓	✓	✓
Lifetime technical support		✓	✓	✓
Weblink & upload link		✓	✓	✓
FTPS (Secure FTP over SSL)			✓	✓
HTTPS (Web Client over SSL)			✓	✓
SFTP over SSH2			✓	✓
Store users via ODBC database				✓
Store users via Mysql database				✓
Events manager				✓
Task scheduler				✓
Windows Authentication				✓
LDAP Authentication				✓
Two-factor authentication (TOTP)				✓

## Requirements

To properly run Wing FTP Server on your server machine or computer, the following requirements are recommended:

### Hardware

- CPU: 2 Cores x86 CPU
- Memory: 1GB of system memory (RAM)

### Software

- Operating System: Windows 7/8/10/11, Windows Server 2008/2012/2016/2019/2022, Mac OS 10.5+, Linux (Ubuntu 12.0+, RHEL 7.0+).
- Web browser for Administration & Web Client: Google Chrome, Firefox, Edge, Safari, etc.

## How to upgrade?

- Download the latest version.
- Backup your data files to a safe place (copy the whole "/Data" directory).
- Stop the Wing FTP Server service.
- Install the new version without any change.

## Purchase

Wing FTP Server is distributed under a shareware license, you can download and evaluate a fully functional trial version for 30 days. After 30 days, you can continue to use it as Free edition for **non-commercial use**.

If you need more features available in **Standard / Secure / Corporate** edition, please purchase a license through our website: <https://www.wftpserver.com/order.htm>.

And the license includes a **2-year upgrade protection** by default (the software itself never expires, all updates can be downloaded for free within two years). However, if you want to get a longer protection period, you can choose from the options we offer after you click the purchase link. If your upgrade protection has expired or will expire soon, please contact us at **sales@wftpserver.com**.

On payment approval, we will send you the registration code (license key) within 12 hours, and it will be sent from email address: **support@wftpserver.com (or no-reply@2checkout.com)**. Please make sure your mail server won't block or filter this email address.



## Quick Start Guide

We will help you setup Wing FTP Server in a few minutes.

### Installation

First, you need to download Wing FTP Server's installer file through our website: <https://www.wftpserver.com/download.htm>

For Windows version, launch the installer file "WingFtpServer.exe", and fill out the required options to finish the installation.

For Linux version, you can extract the downloaded installer file "wftpserver-linux-64bit.tar.gz" with the following command line:

```
$ tar xzvf wftpserver-linux-64bit.tar.gz
```

That will extract the files from the package into a directory named "wftpserver", then jump into this directory and setup Wing FTP Server:

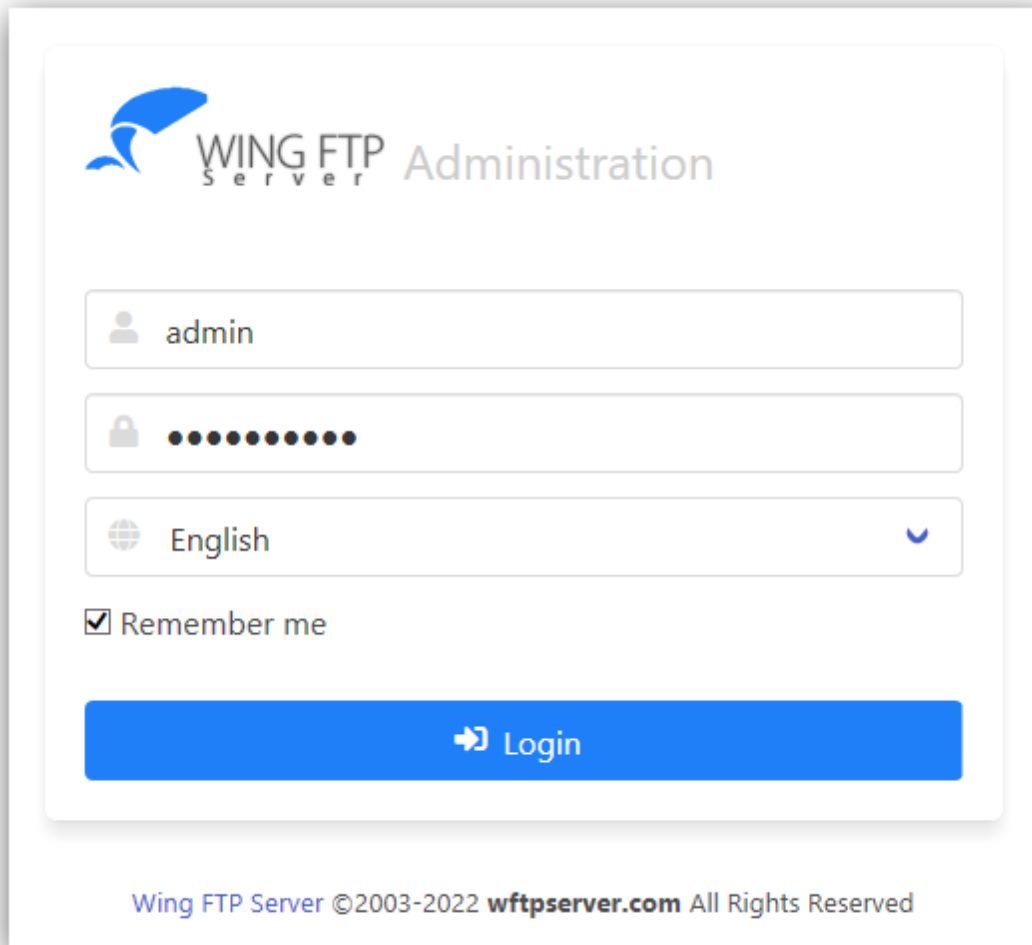
```
$ cd wftpserver
```

```
$ sudo ./setup.sh
```

### Login Administration

When the installation of Wing FTP Server is completed, you can launch a web browser and type in <http://127.0.0.1:5466> to start the web-based administration. 5466 here is the default listener port for web administrator, and you can change it during installation.

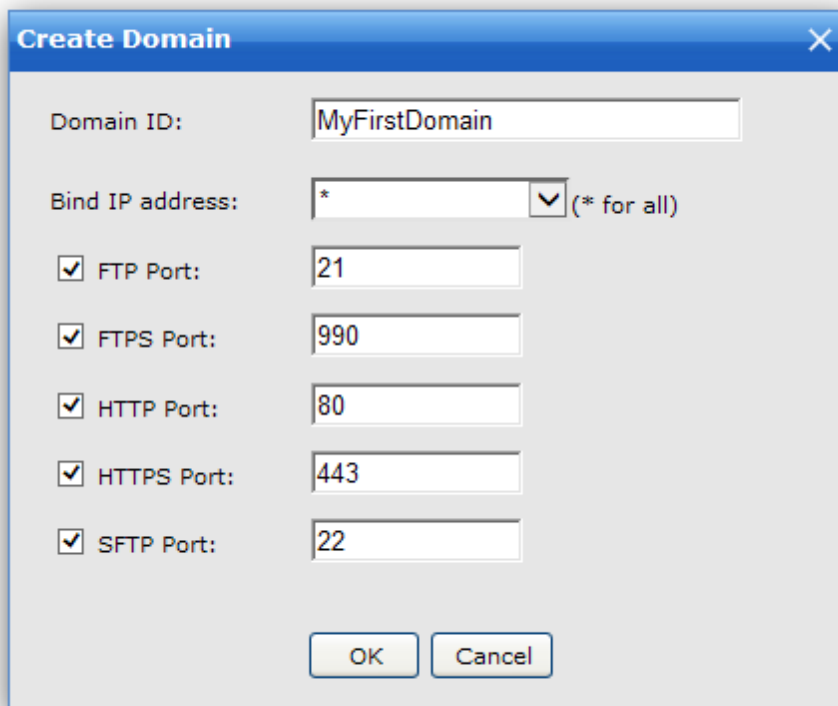
Windows users can launch the administration simply by double-clicking the Wing FTP Server icon in the system tray or on the desktop. Enter your admin name and password created during installation, then click the "Login" button.



The image shows a login form for Wing FTP Server Administration. At the top left is the logo, a blue stylized bird-like shape. To its right is the text "WING FTP Administration" with "Server" in smaller letters below "WING". Below the logo and title are three input fields: the first contains the username "admin", the second contains a password represented by ten black dots, and the third contains the language "English" with a globe icon on the left and a dropdown arrow on the right. Below these fields is a checked checkbox labeled "Remember me". At the bottom of the form is a large blue button with a white right-pointing arrow and the text "Login". At the very bottom of the page, below the form, is the copyright notice: "Wing FTP Server ©2003-2022 wftpserver.com All Rights Reserved".

### Creating Your First Domain

If no domain has been created before, then you will be asked whether you would like to create the first domain. Click "Yes" and it will open the domain creation window:



The screenshot shows a 'Create Domain' dialog box with the following fields and options:

- Domain ID: MyFirstDomain
- Bind IP address: \* (\* for all)
- FTP Port: 21
- FTPS Port: 990
- HTTP Port: 80
- HTTPS Port: 443
- SFTP Port: 22

Buttons: OK, Cancel

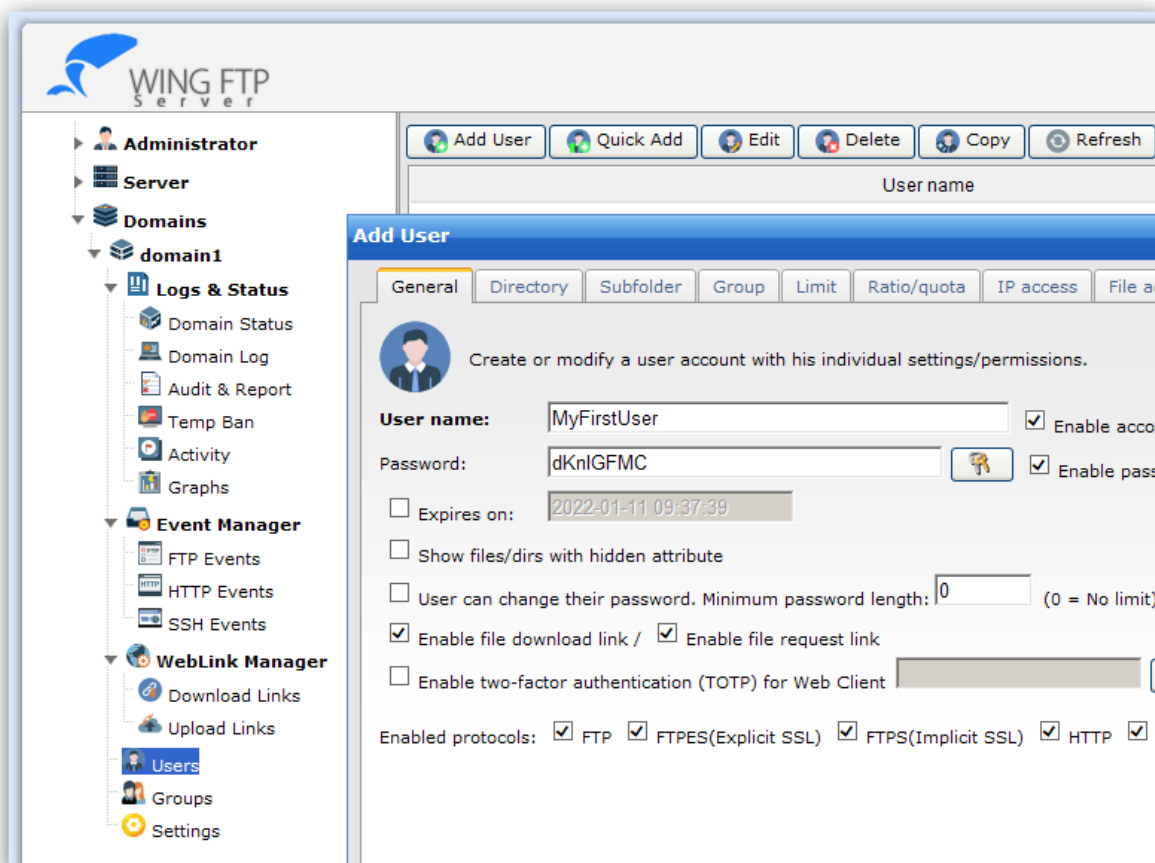
You need to provide a unique name for the domain ID, and it is simply an identifier to distinguish the domain from others.

If you want to keep all the default listeners, just click the button "OK".

Now your first domain has been created successfully! It is so easy, right? More domain settings can be found and configured under "Domain -> Settings".

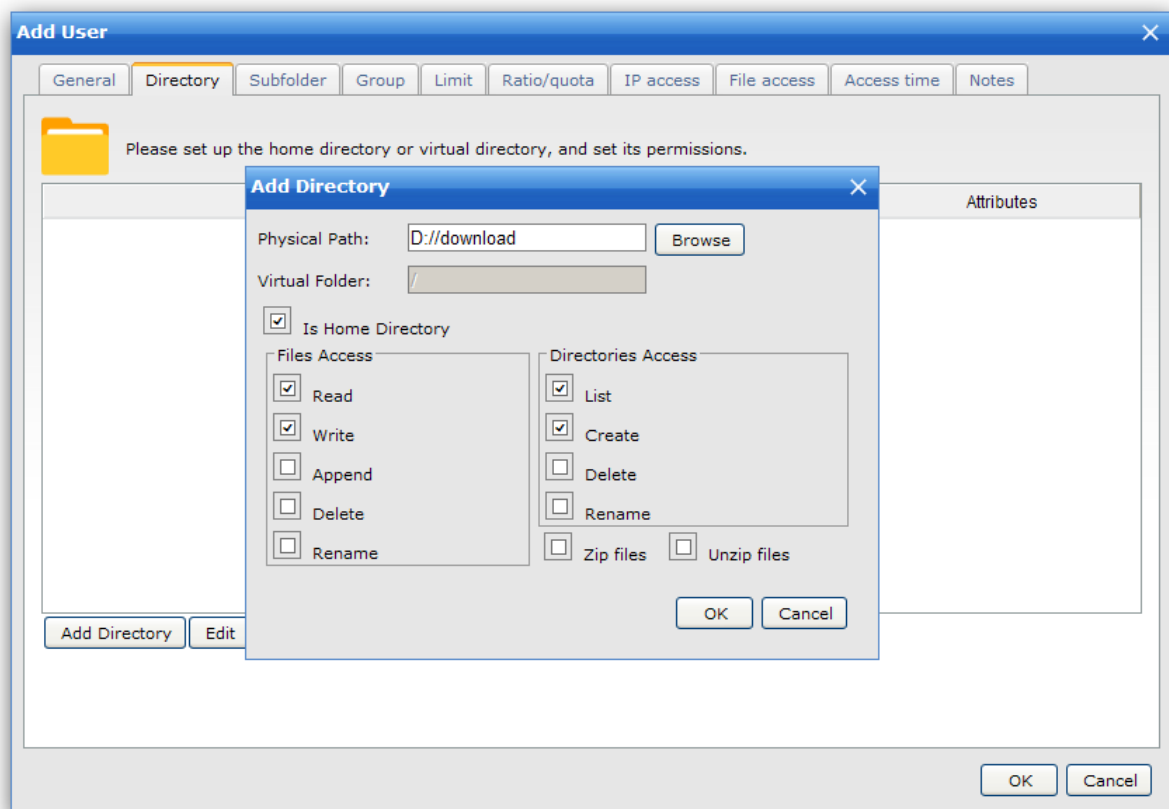
## Creating Your First User Account

After your first domain is created, you need to add a user account for this domain. Navigate to "Domains -> YourDomain -> Users", then click the button "Add User":



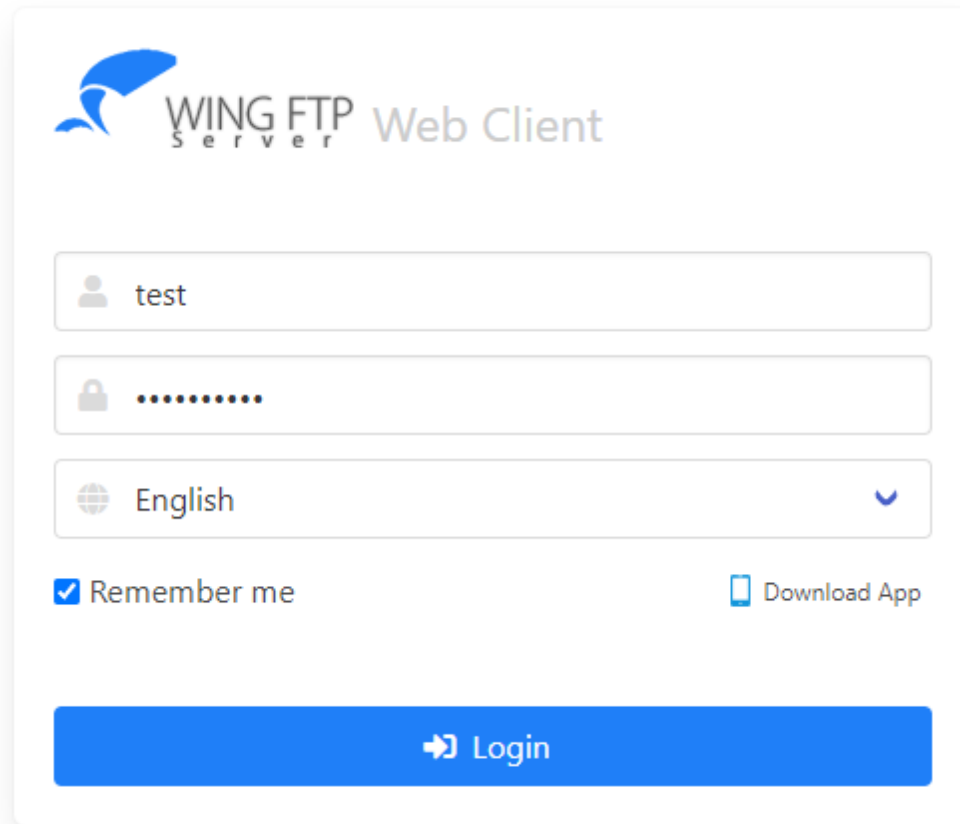
You need to input the username/password for the user account. To create an anonymous account, just check the option "anonymous."

Adding a home directory is also necessary. Just switch to the tab "Directory" and click the button "Add Directory," then choose a physical folder by clicking the "Browse" button (or enter a UNC path directly). The final step is to grant access rights for that folder, and the default access rights are "File Read" and "Directory List," which allows users to list files or download files.



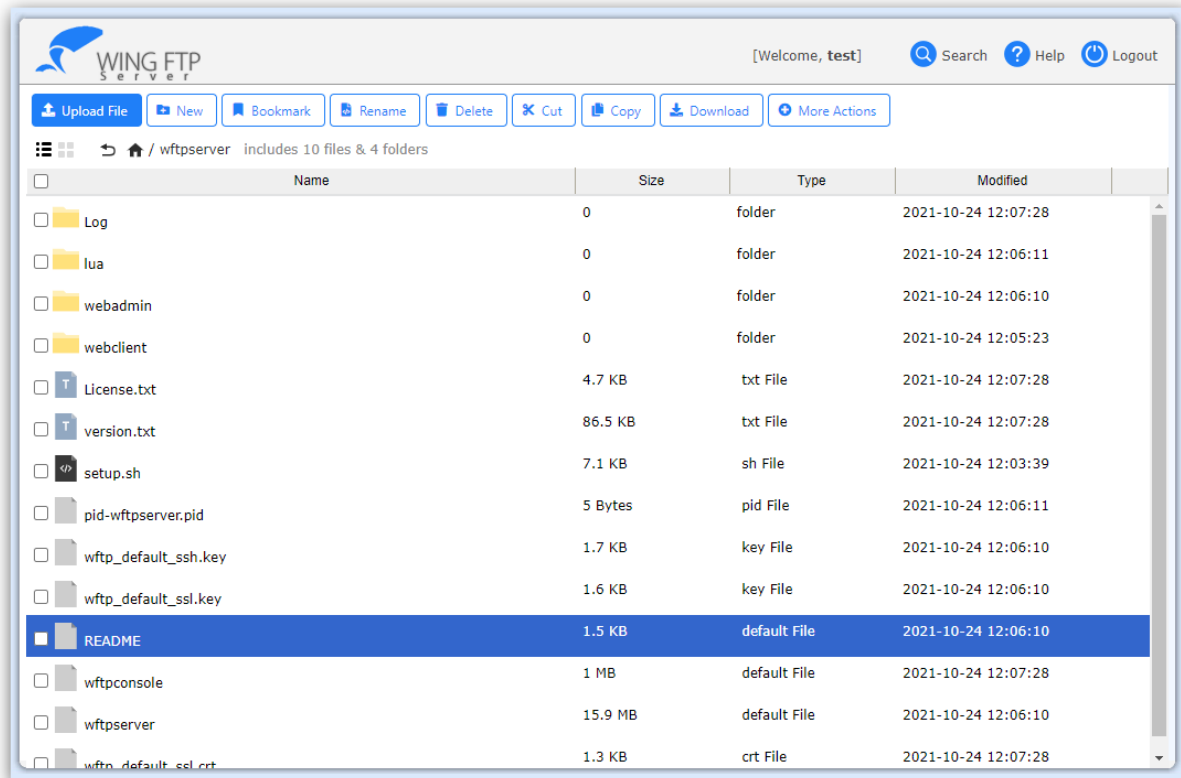
## Testing

Congratulations! Your Wing FTP Server is now accessible and ready for file sharing. If you've enabled the HTTP protocol during domain creation, you can now test it with a web browser. Enter <http://127.0.0.1>, and you will be taken to the login page shown below:



The screenshot shows the login interface for the WING FTP Server Web Client. At the top left is the logo, which consists of a blue stylized wing icon followed by the text "WING FTP Server" in a smaller font and "Web Client" in a larger font. Below the logo are three input fields: the first contains the username "test", the second contains a masked password represented by ten dots, and the third is a language dropdown menu currently set to "English". Below these fields are two options: a checked checkbox labeled "Remember me" and a link with a mobile phone icon labeled "Download App". At the bottom of the form is a large blue button with a white right-pointing arrow and the text "Login".

Type your account name and password, and click the button "Login".



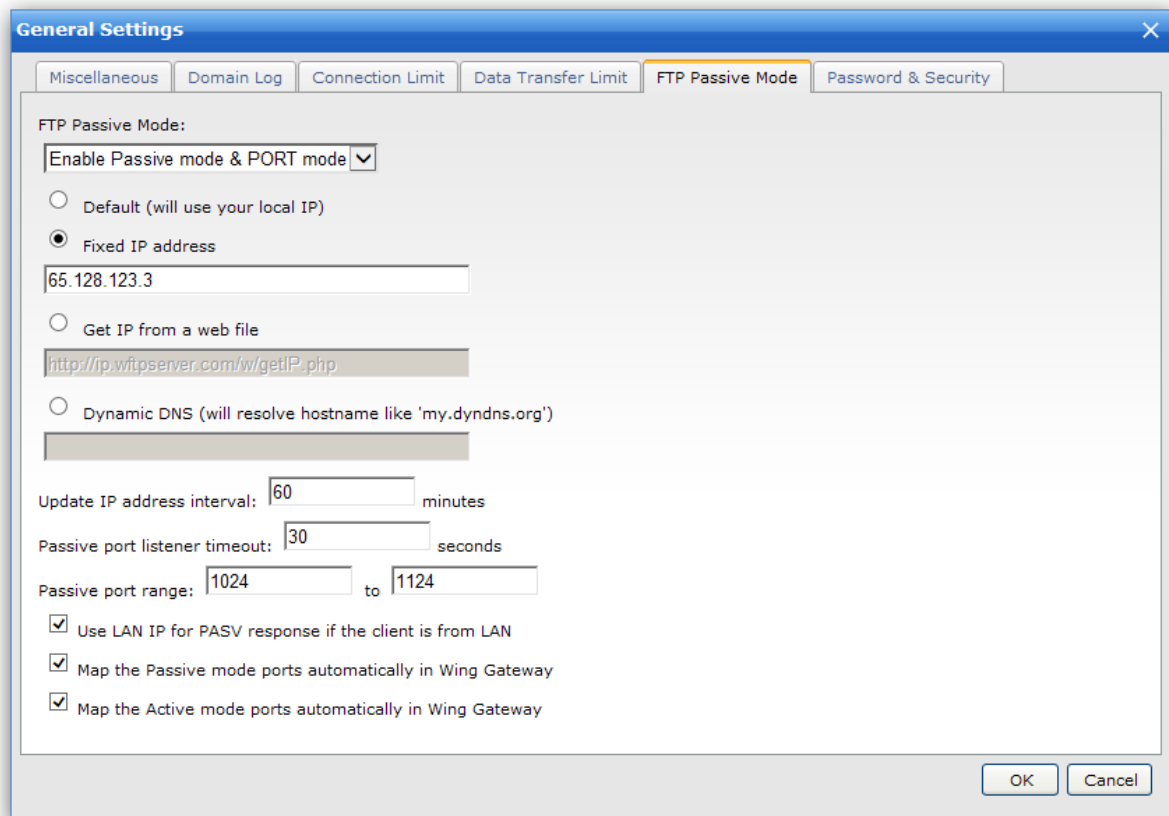
Once you can see the file listing like the above screenshot, you have installed Wing FTP Server successfully.

## Router & Firewall

### Configure FTP service with a router or firewall.

If your WingFTP server machine is in a private local-area network (LAN), and the FTP client connections are from the internet, then you need to forward the necessary FTP ports in your router/firewall.

### Step 1 – Configure Wing FTP Server



1. Login WingFTP's Administration and navigate to "Domains -> Settings -> General Settings -> FTP Passive Mode".
2. Select the option "Fixed IP address" and type the external IP address of your router/firewall. If you don't know the external IP address, just select the option "Get IP from a web file".
3. Set the passive port range (FTP data ports) between 1024 and 1124.
4. Click the button "OK" to save the changes.

## Step 2 – Configure your router/firewall

Just take a few steps to forward the FTP listener ports and FTP passive data ports. Our test scenario is based on a NETGEAR router (Nighthawk R7000), other devices may have the similar process.

1. Login the NETGEAR Administration.
2. Go to the page "Advanced Setup -> Port Forwarding", and click the button "Add Custom Service".
3. Our WingFTP server machine is 192.168.1.123, so just forward the FTP listener ports **21** and **990** to 192.168.1.123.



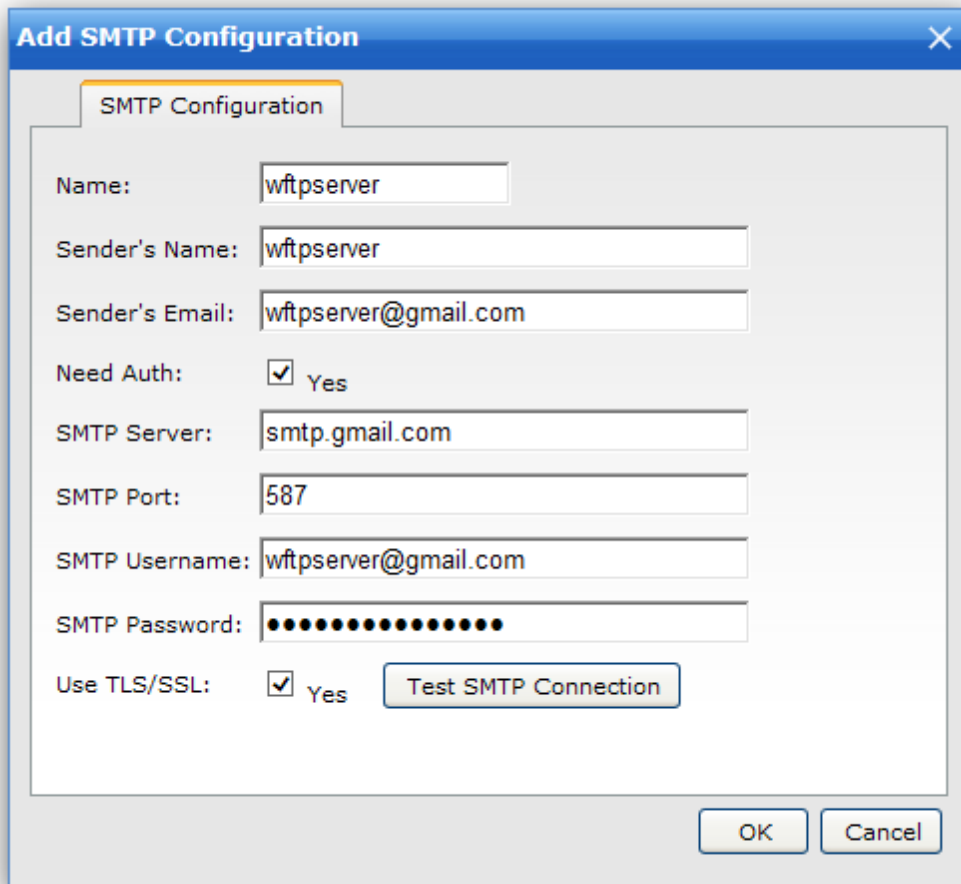
4. Forward the passive ports **1024 - 1124** to 192.168.1.123. It is necessary so that the server will open a data channel for the client when FTP client issues a passive mode transfer.

## Email Notification

**First, you need to add an SMTP server configuration with a few steps:**

1. Navigate to "Domain -> Settings -> General Settings", and click the button "Add SMTP Configuration".
2. If your SMTP server requires authentication, please check on the option "Need Auth".
- 3: Type in the SMTP server address and port.
- 4: Type in your username (email address) and password.
- 5: You should check on the option "Use TLS/SSL" if your SMTP server requires a secure connection.

Here is an example for the Gmail account settings:

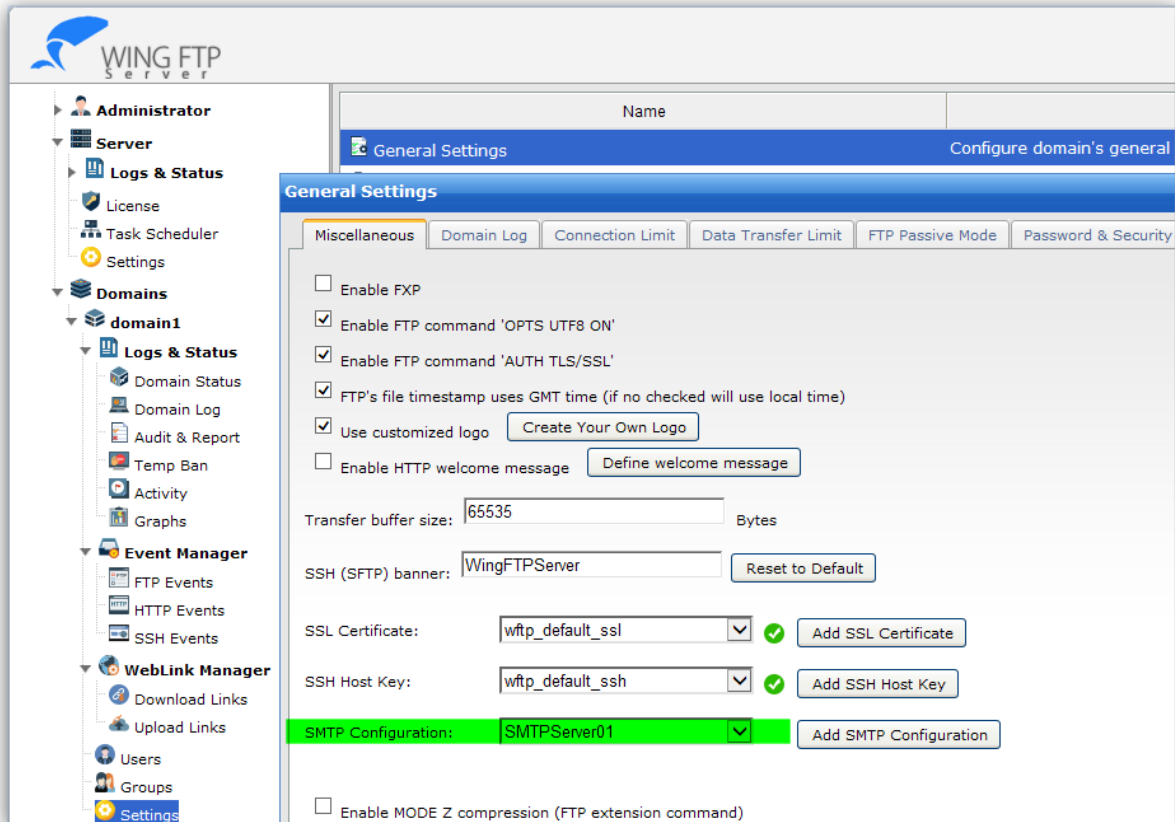


The screenshot shows a dialog box titled "Add SMTP Configuration" with a close button (X) in the top right corner. The dialog has a tab labeled "SMTP Configuration". Inside the dialog, there are several input fields and checkboxes:

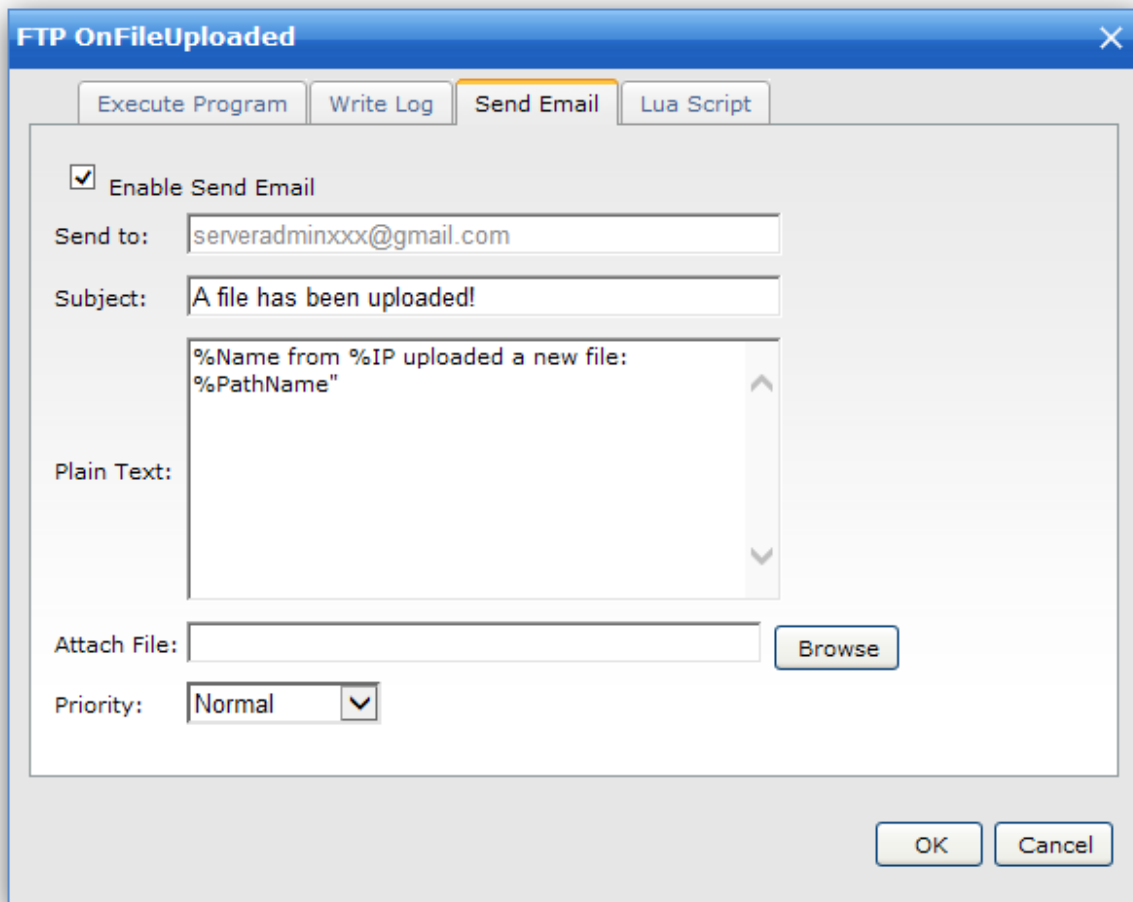
- Name: wftpserver
- Sender's Name: wftpserver
- Sender's Email: wftpserver@gmail.com
- Need Auth:  Yes
- SMTP Server: smtp.gmail.com
- SMTP Port: 587
- SMTP Username: wftpserver@gmail.com
- SMTP Password: [masked with dots]
- Use TLS/SSL:  Yes

At the bottom right of the dialog, there is a "Test SMTP Connection" button, and at the very bottom, there are "OK" and "Cancel" buttons.

After adding a SMTP configuration, you need to select the configuration under Domain Settings, which is shown as below:



Now you can Event Manager to send email notification when a file is uploaded, just open the event "Domain -> Event Manager -> FTP Events -> OnFileUploaded", and check on "Enable Send Email" like the below:



## Administration Console

Administration Console is a handy tool for administrators, which is like the Linux "Shell" or the Windows "Command Prompt." You can run administration commands or Lua scripts from the console.

Wing FTP Server provides two types of consoles - **Application Console** and **Web-based Console**.

**Application Console** is an internal application of Wing FTP Server. You can run this application with the command prompt(or the Linux "shell") on the machine that is installed with Wing FTP Server. This application can be very useful when you are running Linux without GUI, and you don't want to manage your server from a remote web browser.

Open a command window (or Linux shell) and navigate to WingFTP's installation folder, then start the program "wftpconsole" with the required parameters. If you are not sure how to do this, type command "wftpconsole --help" and you will see a list of all allowed options as follows:

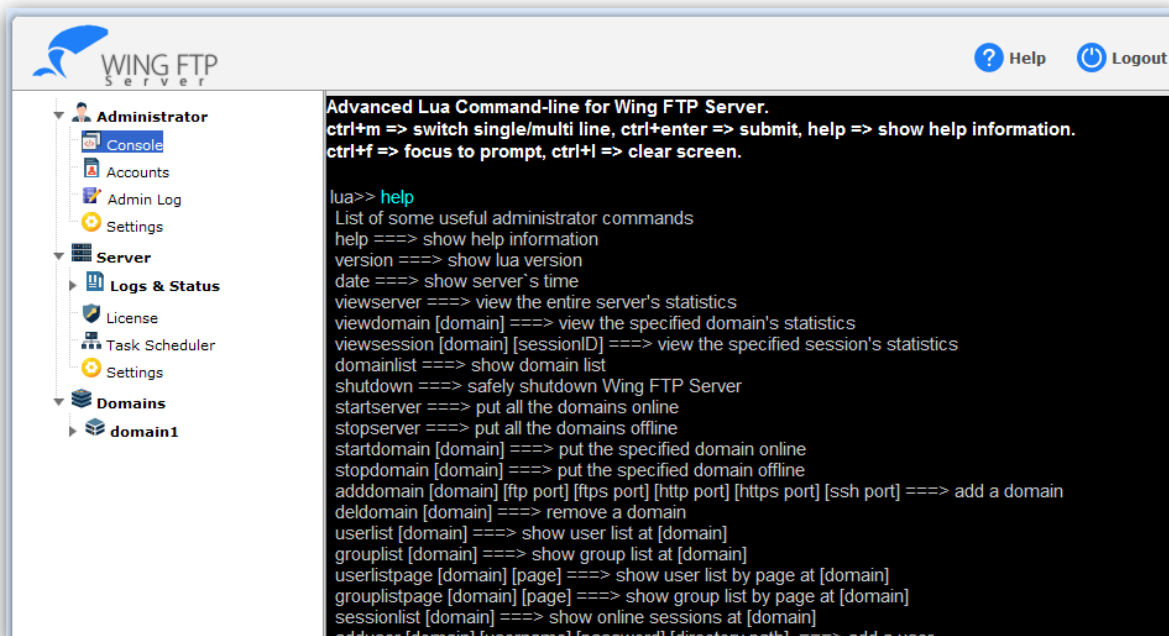
```
Allowed options:
--help           Show this message
-u [ --username ] arg  Username (Required field)
-p [ --password ] arg Password (Required field)
-h [ --host ] arg    Remote host IP address,default is localhost
```

-P [ --port ] arg Remote host port,default is 5466  
 -f [ --file ] arg Parse and execute local Lua <file>  
 -s [ --ssl ] Use SSL connection

- Note 1: After successful connection to the local administration, please type the following command to log in:  
`wftpconsole -u admin -p admin12345`
- Note 2: The "lua>>" prompt indicates that you are in the Wing FTP Server administration, and it is ready to receive your commands.
- Note 3: By entering command "help," you will get a list of useful administration commands.
- Note 4: You can input Lua script and execute it, like "print(1+2+3)".
- Note 5: Quit the console by typing "exit".

### Web-based Console

The Web-based Console is quite similar to the Application Console. It is located in the Web-based Administration. When you have logged on the Web admin, click on the left panel of "Administration -> Console" and the Web-based Console will appear on the right panel. Type the "help" command, and then you can see a list of administration commands.



Type the "domainlist" command to get the domain list on your server.

```

lua>> domainlist
domain1
domain2
  
```

Type the "userlist" command with a specified domain name to get the user list for that domain.

```
lua>> userlist domain1
aa
bb
cc
dd
```

## Lua Scripts

Apart from the Administration commands, normal lua script code can also be run here:

```
lua>> print (3+5)
8
```

```
lua>> for i=1,20 do print(i) print(" ") end
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```


There are lots of server APIs and variables available for your scripts.


```
lua>> print(c_GetAppPath())
C:\dev\ftpserver\debug\

lua>> print(c_GetUserList("domain1"))
aa
bb
cc
dd

lua>> print("%ServerName %ServerVersion %CurrentTime")
Wing FTP Server Corporate Trial 3.0 Wed, 01 Jul 2009 16:17:17 GMT
```

You can switch between the Single/Multiple line mode by clicking the icon before the "lua>>" prompt or by pressing "Ctrl + M".

Single Line:  lua>>

Multiple Lines:  lua>>

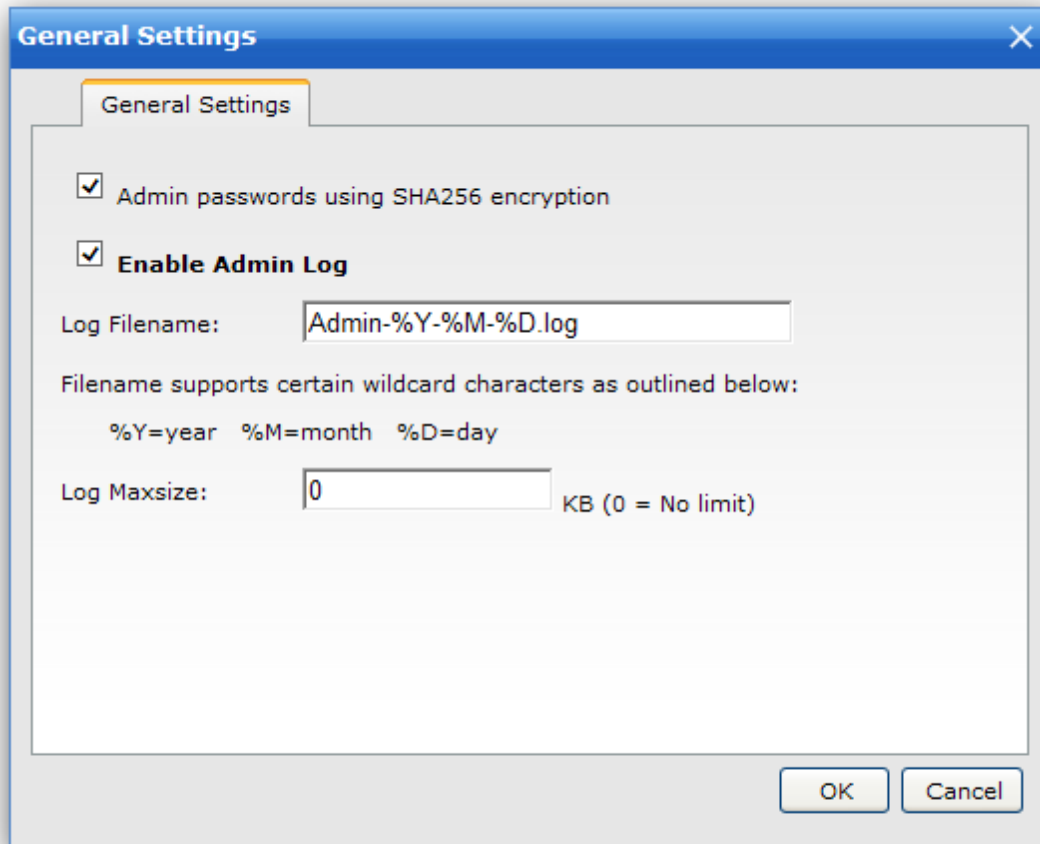
Press "Ctrl+Enter" to submit your multi-lines Lua Scripts code.

```
lua>> print(c_GetAppPath())
lua>> for i=1,20
lua>> do
lua>> print(i)
lua>> print(" ")
lua>> end
lua>>
C:\dev\ftpserver\debug\1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

## Settings

### General Settings

Here you can configure general settings for the web administration.



#### **Admin passwords using SHA256 encryption**

The admin password will be stored with SHA256 hash by default, when this option is disabled, the admin password will use MD5 hash.

#### **Enable Admin Log**

Check on this option to save the admin logs to the log files.

#### **Log Filename**

All the admin log files will be stored under the folder "Log/Admin", the log filename supports certain wildcard characters as outlined below:

%D - The current day of the month

%M - The name of the current month

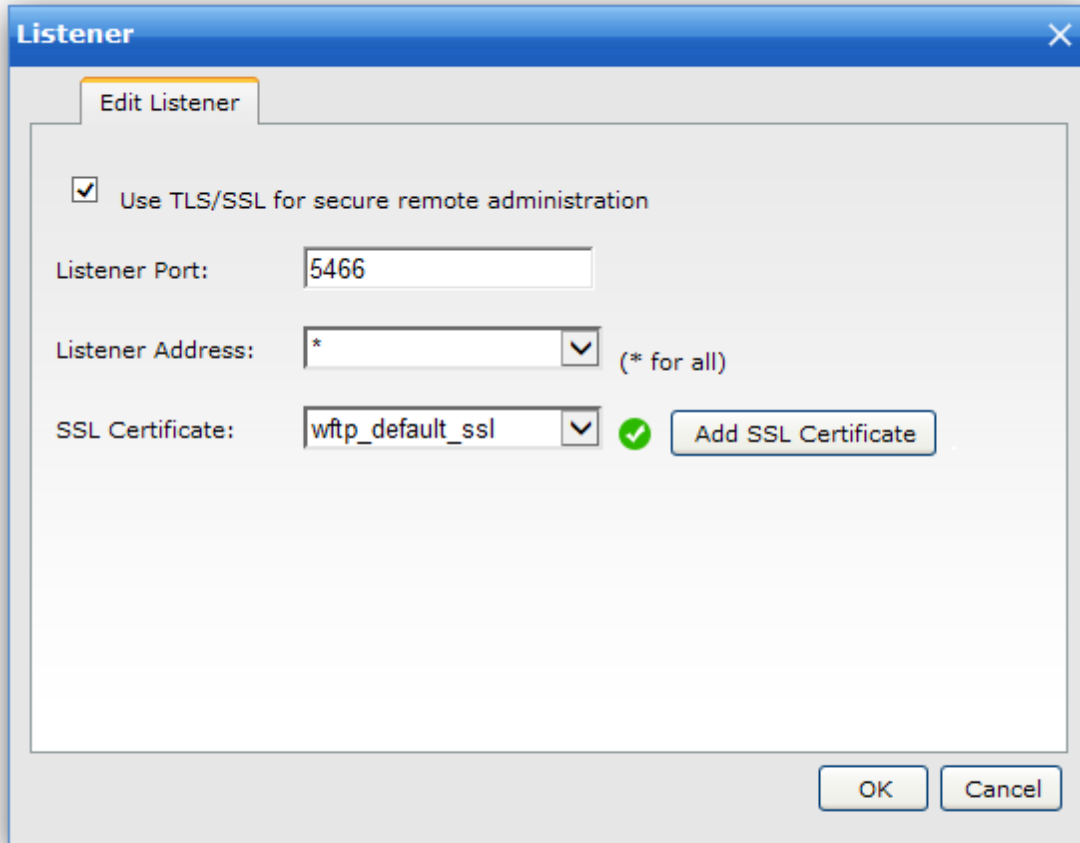
%Y - The 4-digit value of the current year (e.g., 2009)

#### **Log Maxsize**

The log file will no longer be written in when it reaches the size limit you set here.

## Listener

Here you can configure web admin's listener.



### **Use TLS/SSL for secure remote administration**

If you want to use HTTPS for remote administration, please check on this option and choose an SSL certificate.

### **Listener Port**

This web admin's listener port, default is 5466.

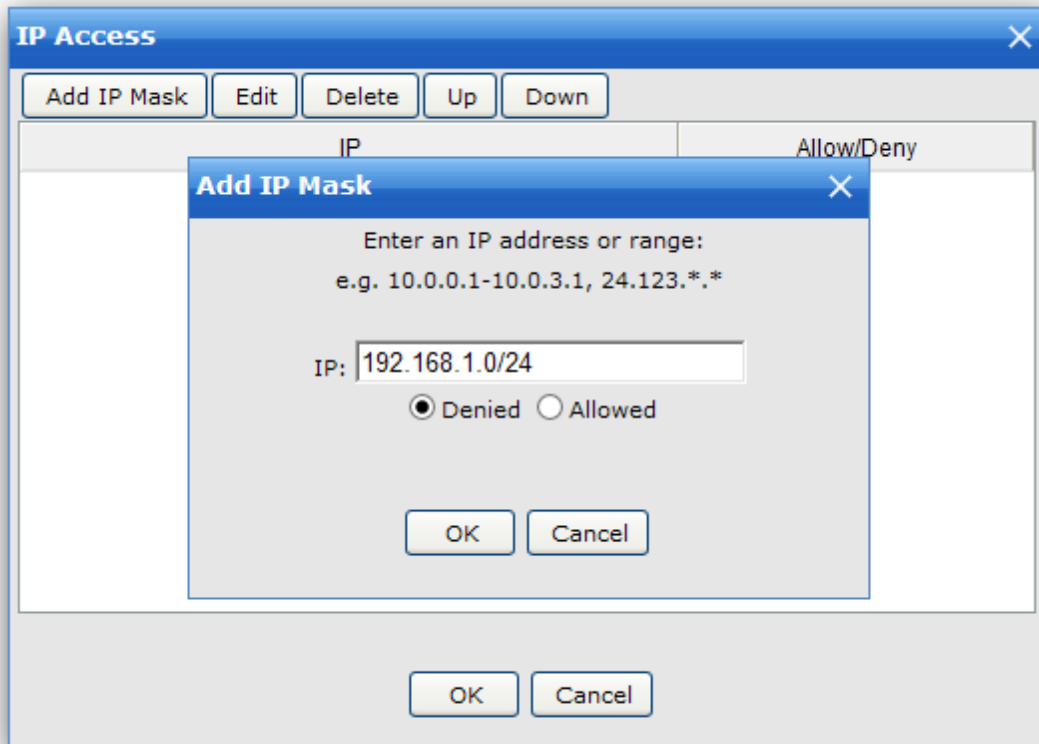
### **Listener Address**

This web administration listener address, default "\*" means the admin port listens on all available IPs. If you change it into a single IP address, then the admin port will listen on that IP address only.



## IP Access

Configure IP access rules to allow or deny access for the administrators.



If you do not specify any IP address, the administrators can log on the server from any IP address. But if you set allowed rules, the administrators will only gain access from the IP addresses specified in the allowed rules. If you set denied rules, the administrators will have access to the server from any IP address except those specified in the denied rules. ***The order of the rules is also critical.***

For example:

**Allow 192.168.1.188**

Refuse connections from any IP except 192.168.1.188.

### Administrator from localhost

Remember whatever IP rules you add, administrators can always login from the local IP address 127.0.0.1

### Supported wildcards

Wing FTP Server supports IP address ranges and wildcards, which are shown as below:

xxx.xxx.xxx.xxx

IP address must be exactly matched(e.g., 192.168.1.1).

xxx.xxx.xxx.xxx-yyy

A specified range of IP addresses, e.g. 192.168.1.10-25.

xxx.xxx.xxx.xxx-yyy.yyy.yyy.yyy

A specified range of IP addresses, e.g. 192.168.1.0-192.168.5.255.

\*.\*.\* or xxx.\*.\* or xxx.xxx.\*.\* or xxx.xxx.xxx.\*

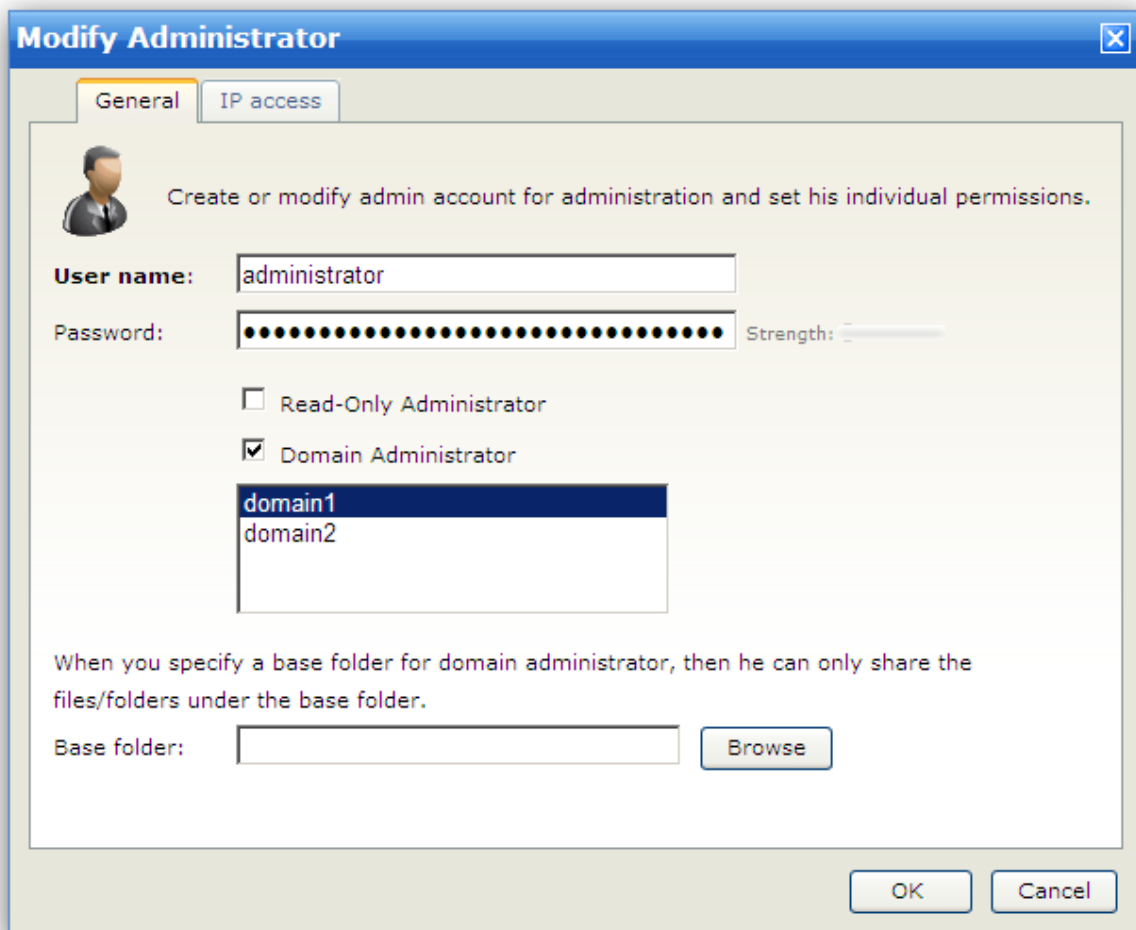
Any valid IP address value (For example, 192.168.\*.\* represents any IP between 192.168.0.0 and 192.168.255.255).

CIDR convention is also supported :

192.168.0.0/24 (represents any IP between 192.168.0.0 and 192.168.0.255)

## Admin User

An admin user account is required to administrate your server.



User Name

Also known as the "Login ID," The administrator will log in using this username. The user name must be unique to the server and can not contain any of the following special characters: \ / < > |:? \*.

## **Password**

A password is required for a session to be authenticated by the Server. A strong password contains at least 6 characters including upper and lower case letters and at least one number. Restrictions can be placed on the length and complexity of passwords through limits configuration.

## **Read-Only Administrator**

If you enable the option "Read-Only Administrator," then that administrator can only view server settings instead of changing them.

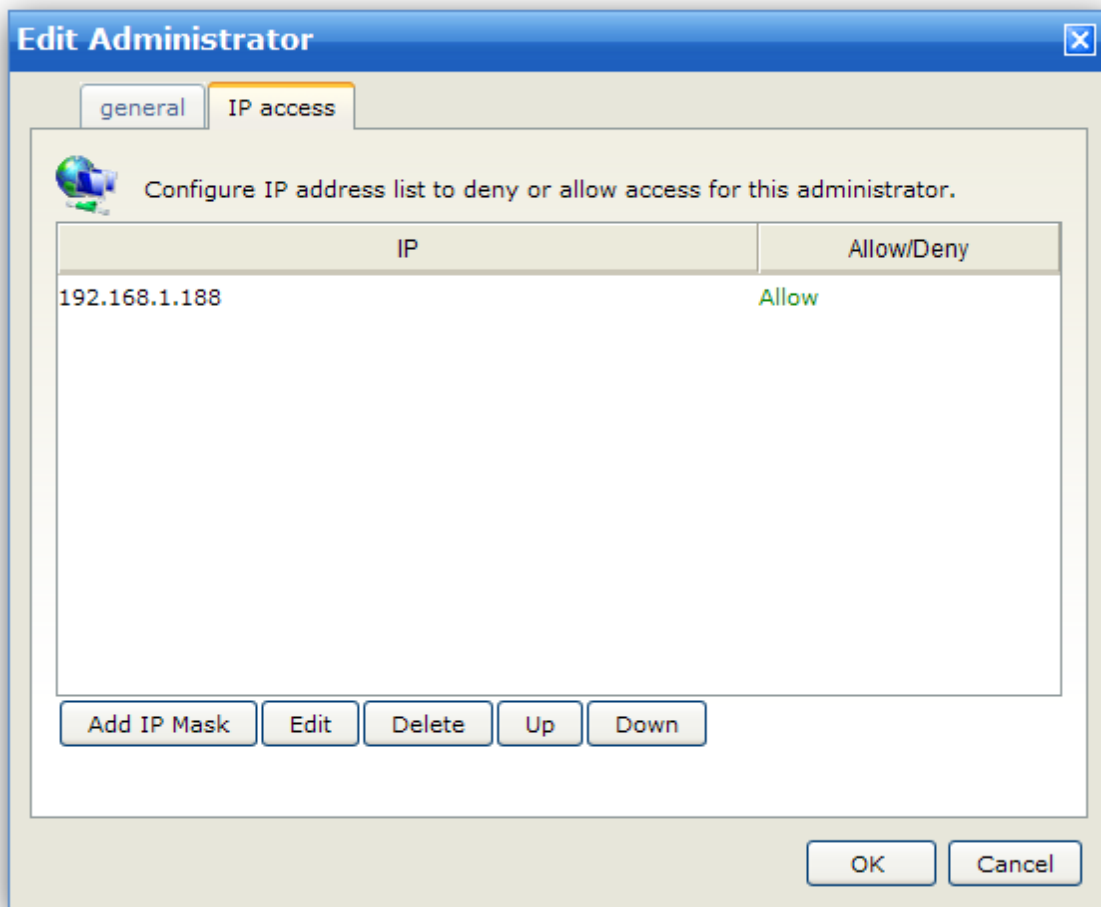
## **Domain Administrator**

If you enable the option "Domain Administrator," you must select one or more domains below, then that administrator can only administrate the selected domains (such as managing users, groups, or activities).

## **Base folder**

You must enable the option "Domain Administrator" first, to specify a base folder for that domain administrator. Then he can only be able to share files/folders under the base folder.

## **IP Access Tab**



You can define IP access rules to allow/deny users' access based on IP address for this administrator.

If you do not specify any IP address, this administrator can log on the server from any IP address. But if you set allowed rules, the administrator will only gain access from the IP addresses specified in the allowed rules. If you set denied rules, the administrator will have access to the server from any IP address except those specified in the denied rules. ***The order of the rules is also critical.***

For example:

**Allow 192.168.1.188**

Refuse connections from any IP except 192.168.1.188.

#### Administrator from localhost

Remember whatever IP rules you add, administrators can always login from the local IP address 127.0.0.1

#### Supported wildcards

Wing FTP Server supports IP address ranges and wildcards, shown as below:

xxx.xxx.xxx.xxx

The IP address must be exactly matched(e.g., 192.168.1.1).

xxx.xxx.xxx.xxx-yyy

A specified range of IP addresses, e.g. 192.168.1.10-25.

\*.\*.\* or xxx.\*.\* or xxx.xxx.\*.\* or xxx.xxx.xxx.\*

Any valid IP address value (For example, 192.168.\*.\* represents any IP between 192.168.0.0 and 192.168.255.255).

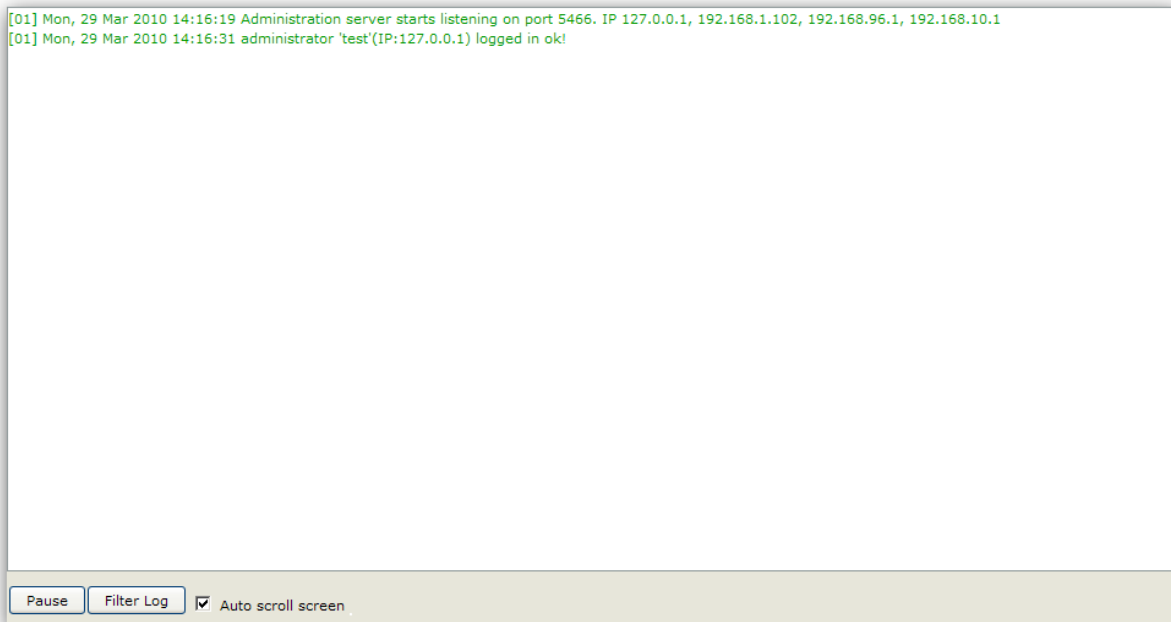
CIDR convention is also supported :

192.168.0.0/24 (represents any IP between 192.168.0.0 and 192.168.0.255),

218.123.2.1/29 (represents any IP between 218.123.2.1 and 218.123.2.9)

## Admin Log

Here you can check logged administration activities.



### Pause

Click this button to pause the log refreshing temporarily. This is useful on busy systems as it allows particular sections of the log to be highlighted and copied before it is scrolled out of view. Once finished, click this button again to resume the log refreshing.

### Filter Log

To quickly find and read through specific sections of the log, you will need the log filter. It is implemented based on search strings. Click the button to bring up the Filter Log dialog. By providing a search string, you can refresh the log to only display log entries containing the search string. To see the entire contents of the log again, open the Filter Log dialog and clean up the search string.

## Logs & Status

### Server Status

View the server status, download/upload speeds, sent/received bytes, running time ...

View statistics about the entire server	
Statistics	Value
Server Running Time	02:11:59
Sessions	0
Highest Num. Sessions	1
24 Hours Sessions	2
Avg. Session Length	00:02:31
Longest Session	00:10:05
Total Sessions	2
Download Speed	0 Bytes/s
Avg. Session Download Speed	0 Bytes/s
Total Download	0 Bytes
Total Download Files	0 files
Upload Speed	0 Bytes/s
Avg. Session Upload Speed	0 Bytes/s
Total Upload	0 Bytes
Total Upload Files	0 files

#### Sessions

The number of online sessions under this server.

#### Highest Num Sessions

The highest number of concurrent sessions that has been recorded since being placed online.

#### 24 Hrs Sessions

The number of sessions that connected to the server in the past 24 hours.

#### Average Session Length

The average staying time under this server.

#### Longest Session

The longest staying time for a session under this server.

**Total Sessions**

The total number of sessions that have connected since being placed online.

**Download Speed**

Cumulative download bandwidth currently being used.

**Avg. Download Speed**

The average download bandwidth used since being placed online.

**Total Download Bytes**

The total amount of data downloaded since being placed online.

**Total Download Files**

The total number of files downloaded since being placed online.

**Upload Speed**

Cumulative upload bandwidth being currently being used.

**Avg. Upload Speed**

The average upload bandwidth used since being placed online.

**Total Upload Bytes**

The total amount of data uploaded since being placed online.

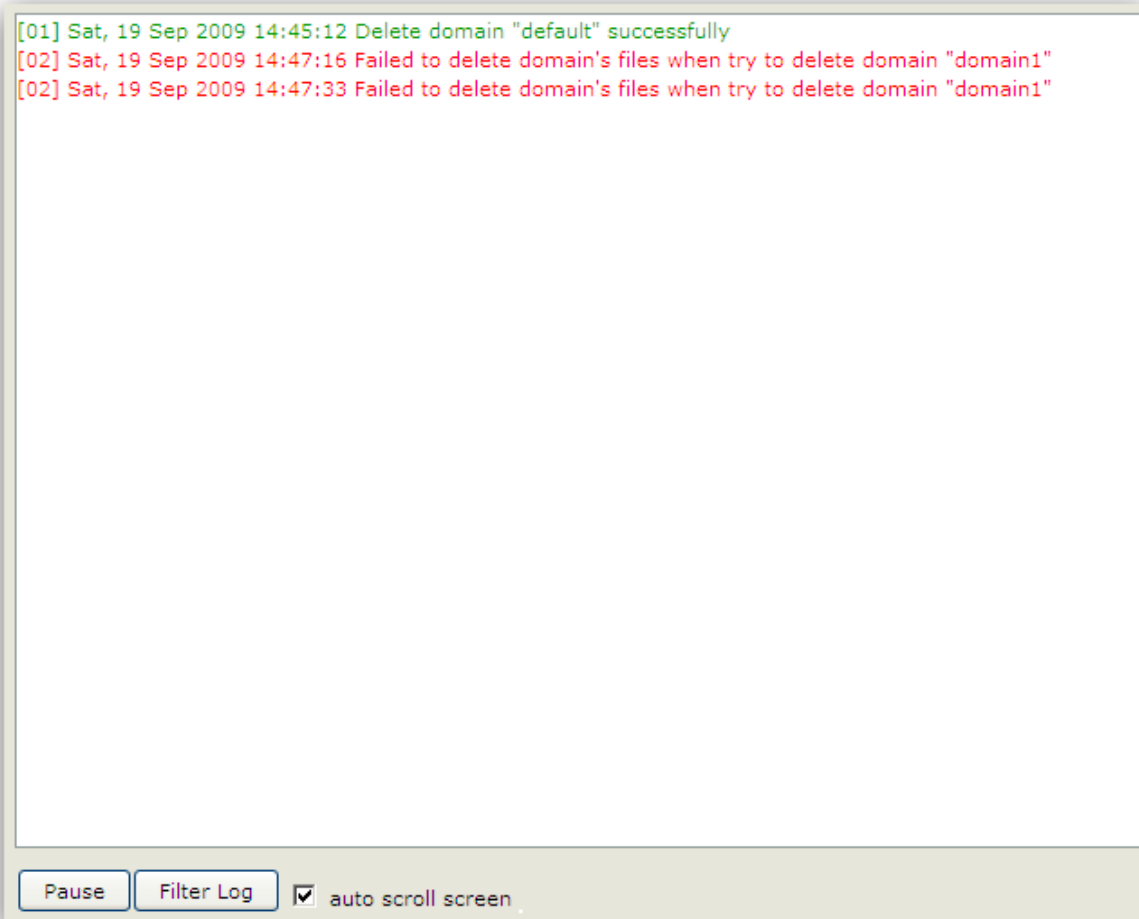
**Total Upload Files**

The total number of files uploaded since being placed online.



## Server Log

View the server logs here.



```
[01] Sat, 19 Sep 2009 14:45:12 Delete domain "default" successfully
[02] Sat, 19 Sep 2009 14:47:16 Failed to delete domain's files when try to delete domain "domain1"
[02] Sat, 19 Sep 2009 14:47:33 Failed to delete domain's files when try to delete domain "domain1"
```

Pause Filter Log  auto scroll screen

### Pause

Click this button to pause the log refreshing temporarily. This is useful on busy systems so a particular section of the log can be highlighted and copied before it is scrolled out of view. Once finished, click this button again to resume the log refreshing.

### Filter Log

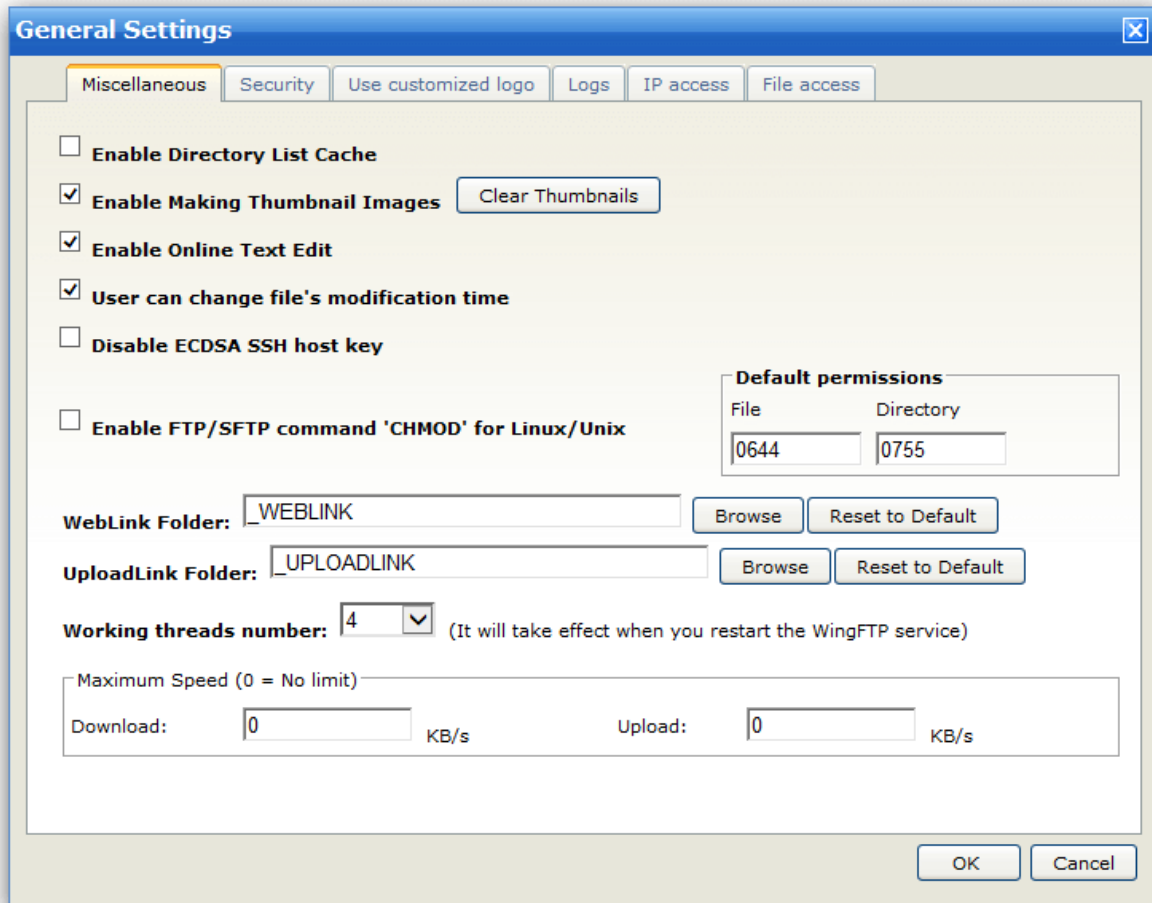
To quickly find the related server logs.

## Settings

## General Settings

## Miscellaneous

## Server Miscellaneous Settings

**Enable Directory List Cache**

This feature allows the server to temporarily keep the directory list for a specified amount of time. And it will accelerate your file system speed.

**Enable Making Thumbnail Images**

This option makes the server generate thumbnail images for web clients.

**Enable Online Text Edit**

This feature allows text files editable online when using a web client.

**User can change file's modification time**

Allow or Disallow FTP/SFTP client to change the file modification time.

**Disable ECDSA SSH host key**

ECDSA SSH host key is enabled by default, anyway, you can disable it.

### Enable FTP command 'SITE CHMOD' for Linux/Unix

When enabled, the FTP client can use the command "SITE CHMOD permissions filename" to change the permissions of a file/folder. Note: only available on Unix-like operating systems.

### WebLink Folder

All the WebLink files are stored under "Data/\_WEBLINK" by default, and you can also specify a folder (like a shared folder) to store them.

### UploadLink Folder

All the UploadLink files are stored under "Data/\_UPLOADLINK" by default, and you can also specify a folder (like a shared folder) to store them.

### Working threads number

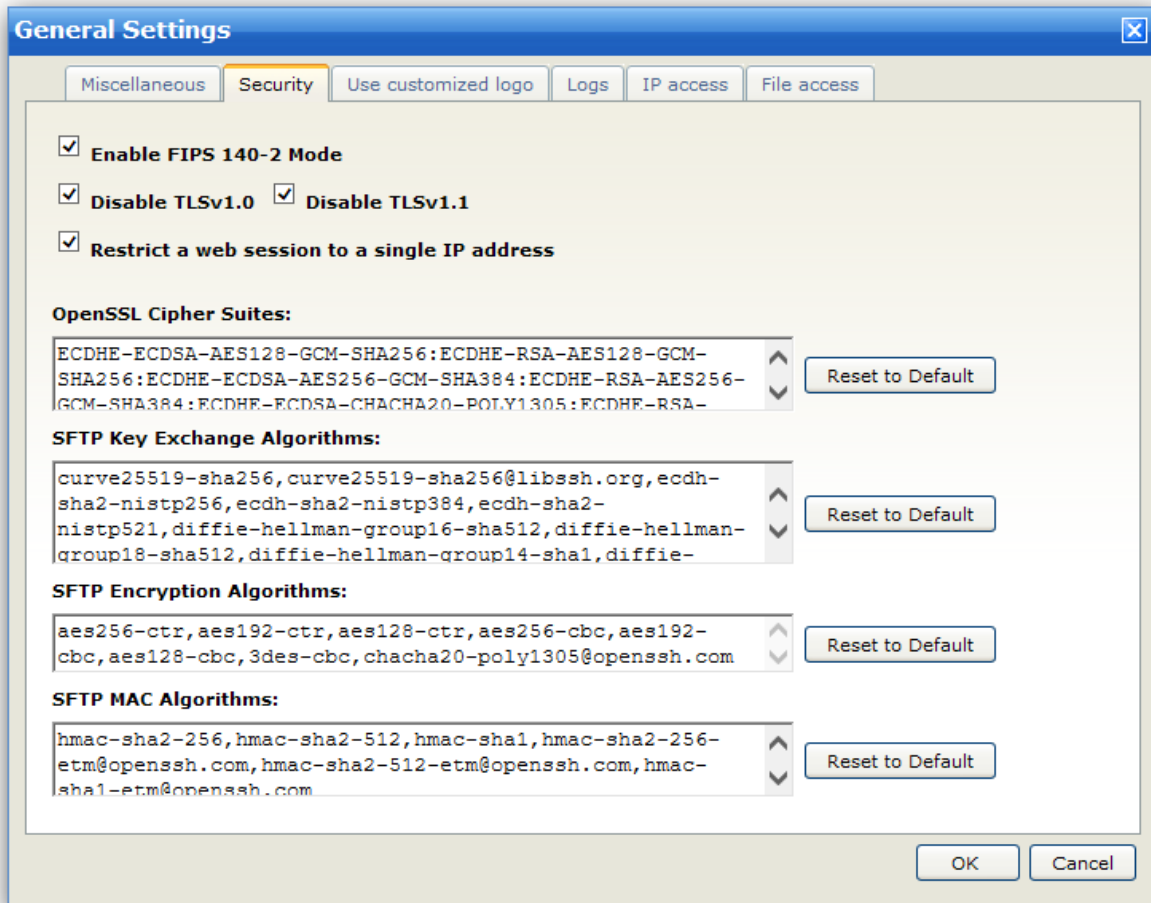
The number of threads the server will create to handle client requests. If you want to get high performance, increase this number, but it will take more CPU usage.

### Maximum Download/Upload Speed

Limits the maximum download/upload bandwidth for the entire server.

### Security

Configure security-related settings here.



**Enable FIPS 140-2 Mode**

When this option is enabled, Wing FTP Server will run all the cryptographic operations according to the FIPS 140-2 standard.

**Disable TLSv1.0/TLSv1.1**

When this option is enabled, Wing FTP Server will run with the protocol TLSv1.2 only.

**Restrict a web session to a single IP address**

This option is enabled by default, it will restrict a web session to a single IP address when the client IP changes, the web session will also expire. When the client IP changes frequently, you may consider to disable it.

**OpenSSL Cipher Suites**

You can use the customized OpenSSL ciphers to configure the Cipher Suite that is permitted to negotiate in the SSL handshake phase by the client. You usually just need to use the default OpenSSL ciphers.

If you need a strong cipher suites, maybe you can have a look at this KB article: <https://bbs.wftpserver.com/viewtopic.php?f=5&t=2936>

**SFTP Key Exchange Algorithms**

The allowed SFTP Key Exchange Algorithms.

**SFTP Encryption Algorithms**

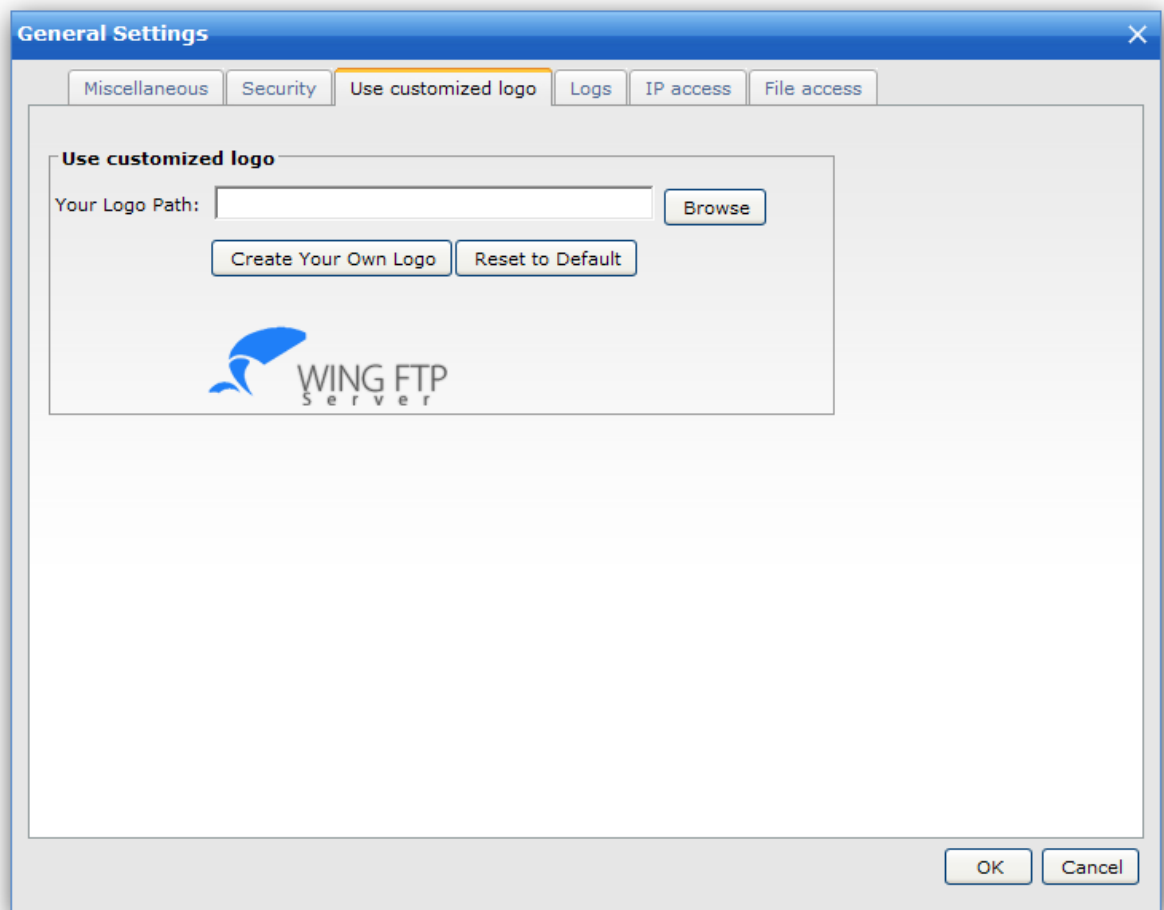
The allowed SFTP Encryption Algorithms.

**SFTP MAC Algorithms**

The allowed SFTP MAC Algorithms.

## Customized Logo

Use a customized logo to be displayed on the login and Web Client pages.



### Your Logo Path

Specify your company logo from a local image file.

### Create Your Own Logo

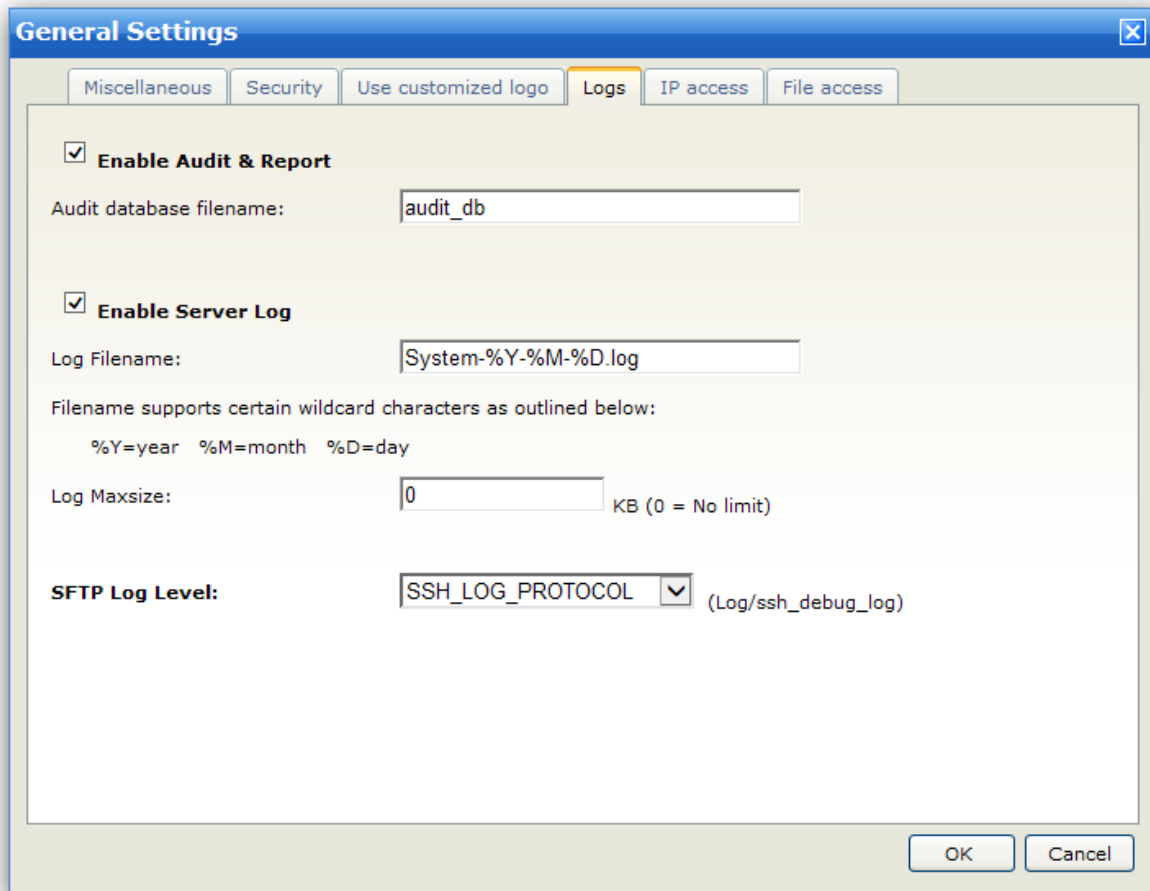
Create your own logo from the image file you specified.

### Reset to default

Reset your logo to the default "Wing FTP Server" logo image.

## Logs

Here you can enable "Audit & Report" and "System Log."



### Enable Audit & Report

When this box is ticked, Wing FTP Server will capture all the transactions into a database for you to analyze and generate reports in real-time.

### Log Filename

The log file must be given a name before information can be saved to a file. Your system log file will be put in the directory "Log\System" by default. The log filename supports certain wildcard characters, as outlined below.

- %D - The current day of the month
- %M - The name of the current month
- %Y - The 4-digit value of the current year, (e.g., 2009)

### Log Maxsize

The log file will no longer be written in when it reaches the limit you set here.

## SFTP Log Level

The SFTP debug logs will be stored in the file "Log/ssh\_debug\_log" by default, and there are 5 logs levels you can choose:

```
// No logging at all  
SSH_LOG_NOLOG
```

```
// Only warnings  
SSH_LOG_WARNING
```

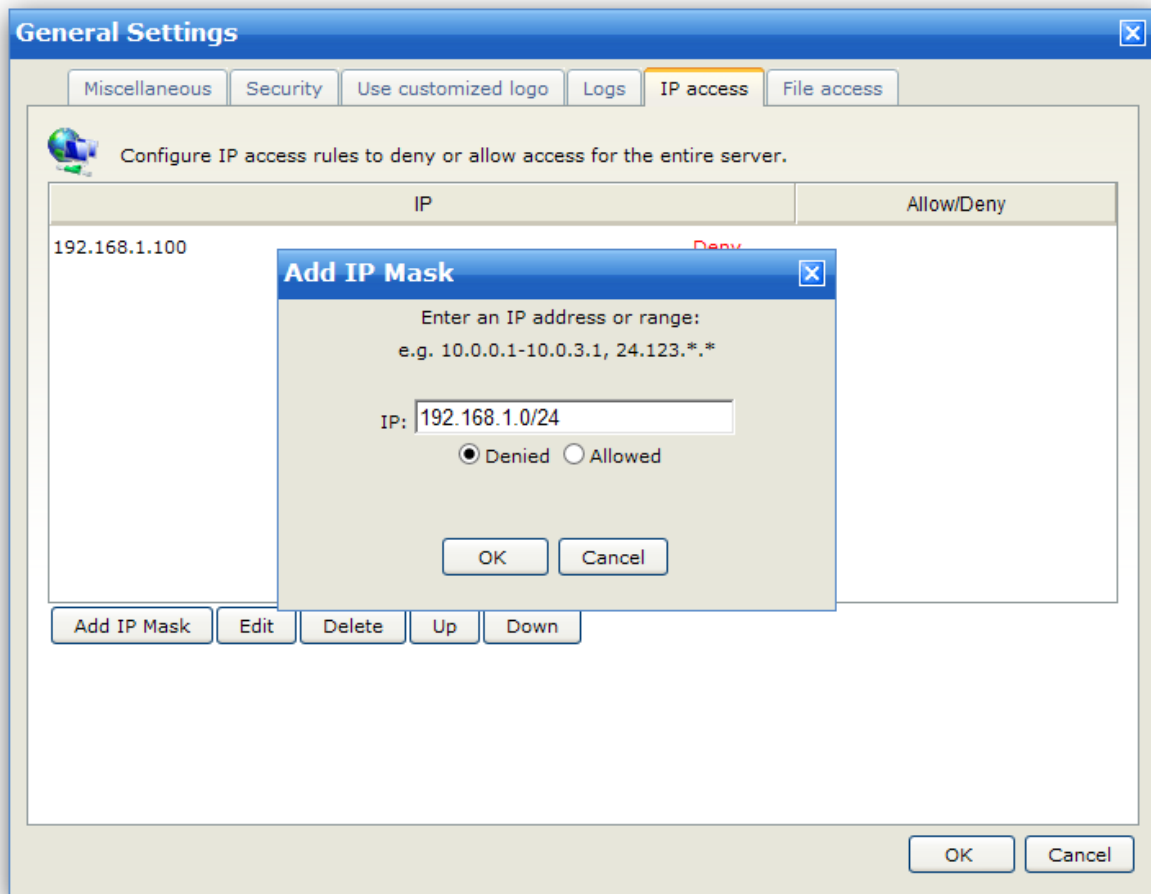
```
// High-level protocol information  
SSH_LOG_PROTOCOL
```

```
// Lower-level protocol information, packet level  
SSH_LOG_PACKET
```

```
// Every function information  
SSH_LOG_FUNCTIONS
```

## IP Access

Configure IP access rules to allow or deny access for the entire server.



You can define IP access rules to allow/deny users' access based on IP address for the entire server.

If you do not specify an IP address, all the users can logon to the server from any IP address. But if you set allowed rules, the users can only obtain access by IP addresses specified in the allowed rules. If you set denied rules, the users can have access to the server by any IP address except those specified in the denied rules.

For example:

**Allow 127.0.0.1**

Refuse connections from any IP except 127.0.0.1.

Supported wildcards

Wing FTP Server supports IP address ranges and wildcards, which are listed as below:

`xxx.xxx.xxx.xxx`

IP address must be exactly matched(e.g., 192.168.1.1).

`xxx.xxx.xxx.xxx-yyy`

A specified range of IP addresses, e.g. 192.168.1.10-25.

`xxx.xxx.xxx.xxx-yyy.yyy.yyy.yyy`

A specified range of IP addresses, e.g. 192.168.1.0-192.168.5.255.

`*.*.*` or `xxx.*.*` or `xxx.xxx.*.*` or `xxx.xxx.xxx.*`

Any valid IP address value (For example, 192.168.\*.\* represents any IP between 192.168.0.0 and 192.168.255.255).

CIDR convention is also supported :

192.168.0.0/24 (represents any IP between 192.168.0.0 and 192.168.0.255)

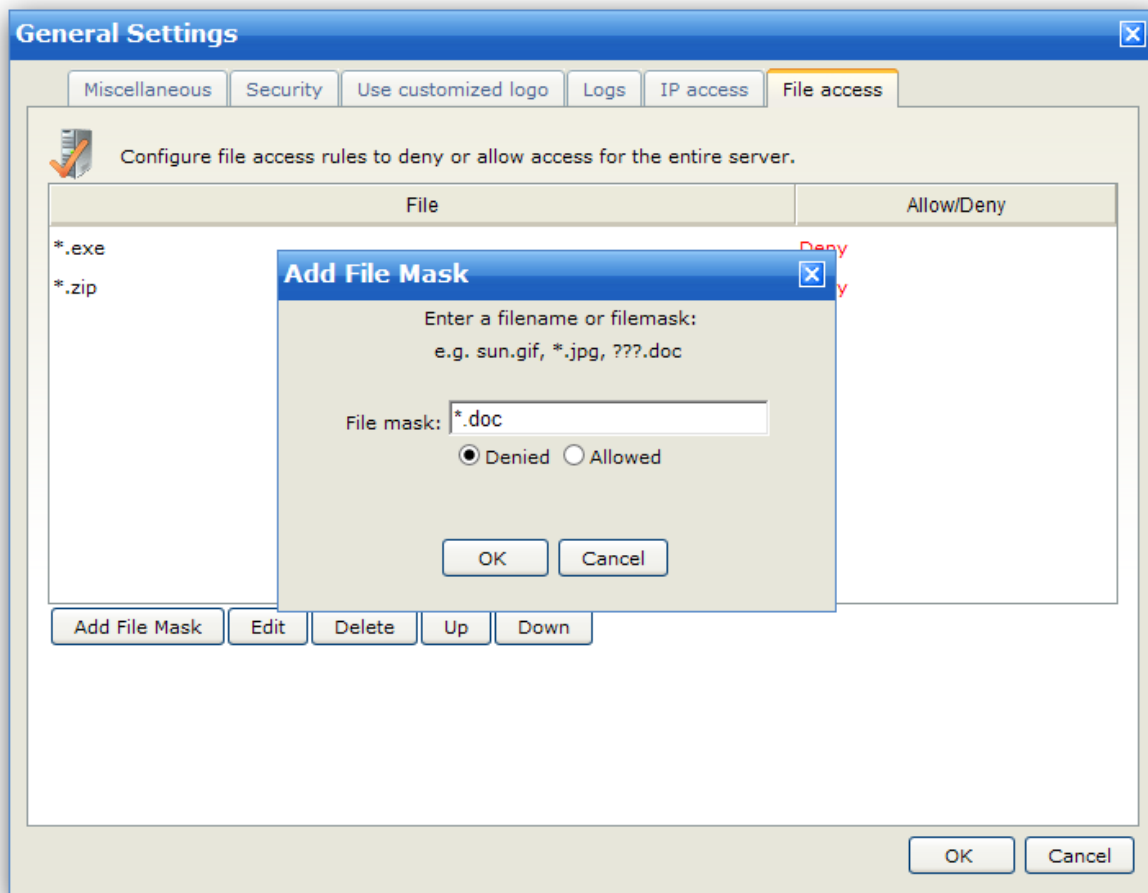
Priority

The Server IP Access rules has priority over Domain IP Access rules.



**File Access**

Configure File access rules to allow or deny access for the entire server.



Banned files are files that can't be accessed on the server. You can specify file/path mask (?, \* supported) :  
\*.jpg, c:\path\images\_200?\

Using this form, you can define rules for the entire server to allow or refuse access according to file names. If you do not specify any file/path, all users will be able to access all the files on the server. But if you set an allow list, all users will only be able to access the files in the list. If you set a deny list, all users can access all the other files except those in the deny list. The order of the rules is very important too.

For example

**Allow \*.rar**

Can not access/store any files except \*.rar.

**Deny \*.zip**

Can access/store any files except \*.zip.

**Rule list**

The rule list shows the current list and order of file access rules. Rules can be added and removed from the list using the Add and Delete buttons.

Also, the order of the rules can be altered using the Up and Down buttons on the right of the rule list.

### Supported wildcards

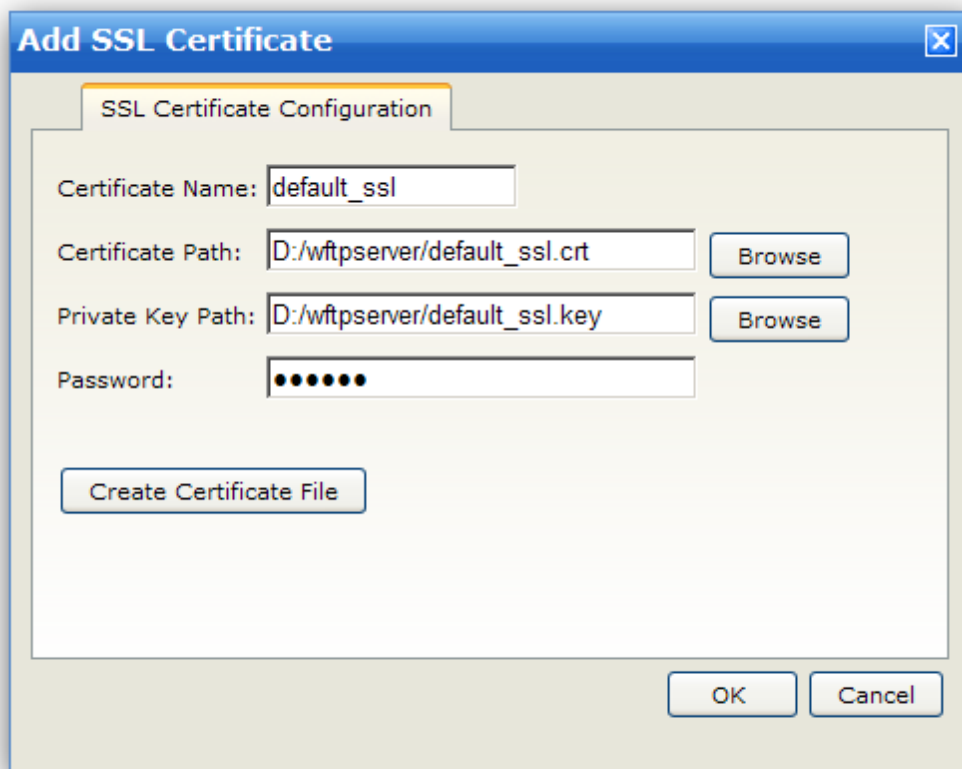
To define access rules, you can use the wildcards ? and \*.

### Priority

The server's File Access list has priority over the domain's.

### SSL Certificate Manager

SSL Certificate is used for encrypting data exchanges between the user and the server. Without certificates, you can not use the transfer protocols FTPS and HTTPS.



Click the "Add SSL Certificate" button to choose an existed SSL certificate.

Wing FTP Server supports standard PEM format only, so normally you need to provide a cert file, a key file and the key password (if existing).

**Note:** If you have an existed PFX cert (which is normally used to store just one private key and it's associated certificate chain), then you will need to convert it into a PEM format cert first.

If you do not have any certificate file, click the "Create Certificate File" button, then a certificate creation form will appear.

**Create SSL Certificate**

Certificate Name:

Output directory:

Key size:  ▾

2-digit country code:

Full state or province:

Locality(city):

Organization:

Organization unit:

Domain Name/Common Name:

Contact E-mail:

Password:

**Certificate Name**

This name will be used for the filename of cert files.

For example, you name the certificate "new\_ssl\_cert", and choose "d:/certfiles" as the output directory, then WingFTP will generate three files under "d:/certfiles":

"new\_ssl\_cert.crt": the self-signed certificate file.

"new\_ssl\_cert.key": the private key file. Please keep it secret to avoid other party misusing your SSL certificate.

"new\_ssl\_cert.csr": the Certificate Signing Request file. You can send it to the Certificate Authorities(CAs) to apply for a signed certificate.

**Output directory**

Output directory is where the created cert files are to be placed.

**Key Size**

Key size for the private key file.

**Full state or province**

Set this to the state or province where you are located.

**Locality(City)**

Set this to the city or town where you are located.

## **Organization**

Set this to the name of your company or organization, like "Wing FTP Software Co., Ltd." .

## **Organization unit**

This should be set to the organizational unit that owns the certificate, like "Software Department " or "IT".

## **Domain Name/Common Name**

Set this to the domain name of the web client/web admin, like "demo.wftpservers.com" .

## **Contact E-mail**

Set this to the e-mail address you want the clients to see.

## **Country**

This has to be the 2-letter abbreviation of the country where you are located (as defined in the ISO-3166 standard), like "US" for the United States.

## **Valid country codes are:**

Afghanistan AF  
Albania AL  
Algeria DZ  
American Samoa AS  
Andorra AD  
Angola AO  
Anguilla AI  
Antarctica AQ  
Antigua and Barbuda AG  
Argentina AR  
Armenia AM  
Aruba AW  
Australia AU  
Austria AT  
Azerbaijan AZ  
Bahamas BS  
Bahrain BH  
Bangladesh BD  
Barbados BB  
Belarus BY  
Belgium BE  
Belize BZ  
Benin BJ  
Bermuda BM  
Bhutan BT  
Bolivia BO  
Bosnia and Herzegovina BA  
Botswana BW  
Bouvet Island BV  
Brazil BR  
British Indian Ocean Territory IO  
  
Brunei Darussalam BN  
Bulgaria BG  
Burkina Faso BF

Burundi BI  
Cambodia KH  
Cameroon CM  
Canada CA  
Cape Verde CV  
Cayman Islands KY  
Central African Republic CF  
Chad TD  
Chile CL  
China CN  
Christmas Island CX  
Cocos (Keeling) Islands CC  
Colombia CO  
Comoros KM  
Congo CG  
Congo, The Democratic Republic Of The CD  
Cook Islands CK  
Costa Rica CR  
Croatia HR  
Cuba CU  
Cyprus CY  
Czech Republic CZ  
Denmark DK  
Djibouti DJ  
Dominica DM

Dominican Republic DO  
East Timor TP  
Ecuador EC  
Egypt EG  
El Salvador SV  
Equatorial Guinea GQ  
Eritrea ER  
Estonia EE  
Ethiopia ET  
Falkland Islands (Malvinas) FK  
Faroe Islands FO  
Fiji FJ  
Finland FI  
France FR  
French Guiana GF  
French Polynesia PF  
French Southern Territories TF  
Gabon GA  
Gambia GM  
Georgia GE  
Germany DE  
Ghana GH  
Gibraltar GI  
Greece GR  
Greenland GL  
Grenada GD  
Guadeloupe GP  
Guam GU  
Guatemala GT

Guinea GN  
Guinea-Bissau GW

Guyana GY  
Haiti HT  
Heard Island and Mcdonald Islands HM  
Holy See (Vatican City State) VA  
Honduras HN  
Hong Kong, China HK  
Hungary HU  
Iceland IS  
India IN  
Indonesia ID  
Iran, Islamic Republic Of IR  
Iraq IQ  
Ireland IE  
Israel IL  
Italy IT  
Jamaica JM  
Japan JP  
Jordan JO  
Kazakstan KZ  
Kenya KE  
Kiribati KI  
Korea, Democratic People's Republic Of KP  
Korea, Republic Of KR  
Kuwait KW  
Kyrgyzstan KG  
Lao People's Democratic Republic LA  
Latvia LV

Lebanon LB  
Lesotho LS  
Liberia LR  
Libyan Arab Jamahiriya LY  
Liechtenstein LI  
Lithuania LT  
Luxembourg LU  
Macau, China MO  
Macedonia, The Former Yugoslav Republic Of MK  
Madagascar MG  
Malawi MW  
Malaysia MY  
Maldives MV  
Mali ML  
Malta MT  
Marshall Islands MH  
Martinique MQ  
Mauritania MR  
Mauritius MU  
Mayotte YT  
Mexico MX  
Micronesia, Federated States Of FM  
Moldova, Republic Of MD  
Monaco MC

Mongolia MN  
Montserrat MS  
Morocco MA  
Mozambique MZ

Myanmar MM  
Namibia NA  
Nauru NR  
Nepal NP  
Netherlands NL  
Netherlands Antilles AN  
New Caledonia NC  
New Zealand NZ  
Nicaragua NI  
Niger NE  
Nigeria NG  
Niue NU  
Norfolk Island NF  
Northern Mariana Islands MP  
Norway NO  
Oman OM  
Pakistan PK  
Palau PW  
Palestinian Territory, occupied PS  
Panama PA  
Papua New Guinea PG  
Paraguay PY  
Peru PE  
Philippines PH  
Pitcairn PN  
Poland PL  
Portugal PT  
Puerto Rico PR  
Qatar QA  
Romania RO

Russian Federation RU  
Rwanda RW  
Saint Helena SH  
Saint Kitts and Nevis KN  
Saint Lucia LC  
Saint Pierre and Miquelon PM  
Saint Vincent and The Grenadines VC  
Samoa WS  
San Marino SM  
Sao Tome And Principe ST  
Saudi Arabia SA  
Senegal SN  
Seychelles SC  
Sierra Leone SL  
Singapore SG  
Slovakia SK  
Slovenia SI  
Solomon Islands SB  
Somalia SO

South Africa ZA  
South Georgia and The South Sandwich Islands GS  
Spain ES  
Sri Lanka LK  
Sudan SD  
Suriname SR

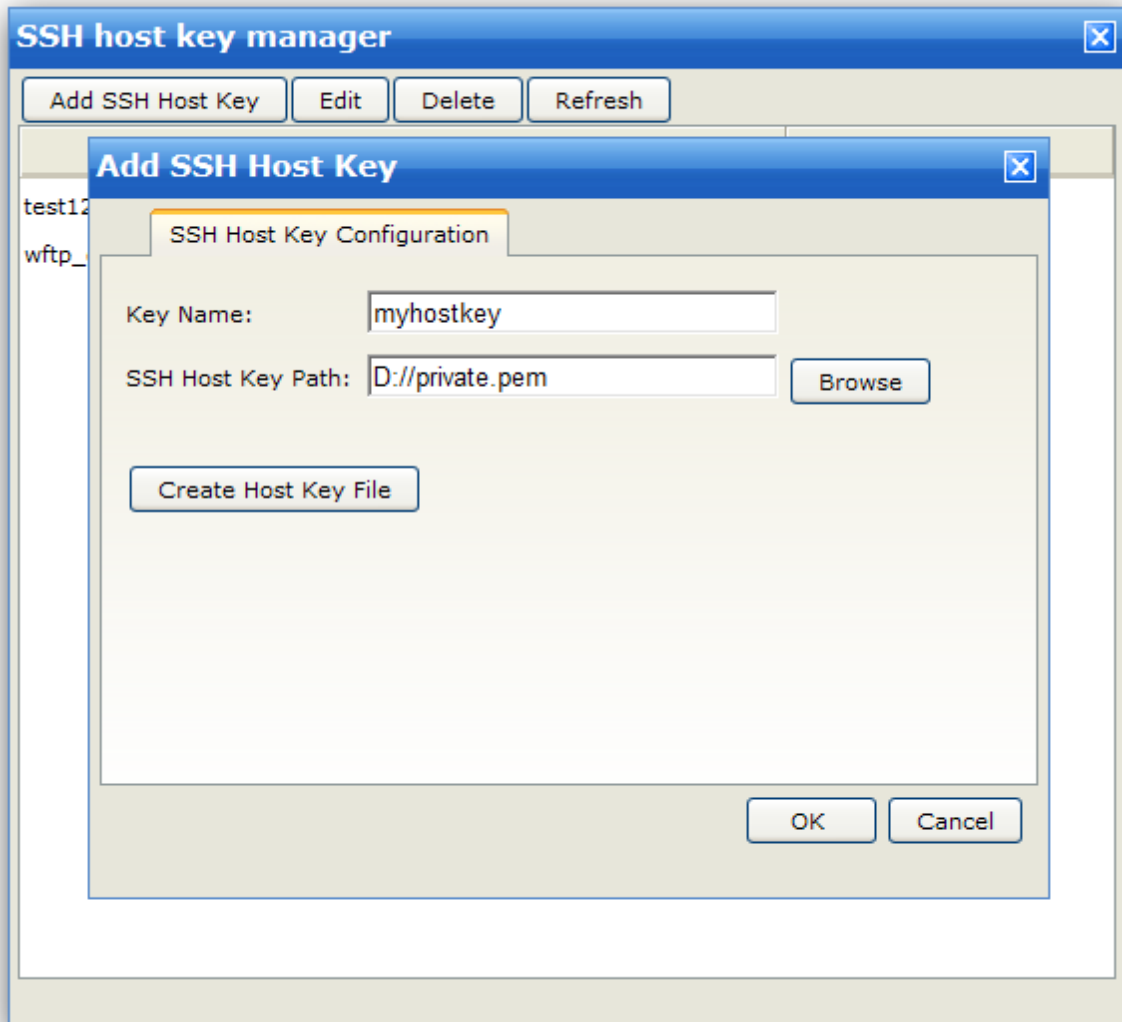
Svalbard and Jan Mayen SJ  
Swaziland SZ  
Sweden SE  
Switzerland CH  
Syrian Arab Republic SY  
Tajikistan TJ  
Tanzania TZ  
Thailand TH  
Togo TG  
Tokelau TK  
Tonga TO  
Trinidad and Tobago TT  
Tunisia TN  
Turkey TR  
Turkmenistan TM  
Turks and Caicos Islands TC  
Tuvalu TV  
Uganda UG  
Ukraine UA  
United Arab Emirates AE  
United Kingdom GB  
United States US  
Uruguay UY

Uzbekistan UZ  
Vanuatu VU  
Venezuela VE  
Viet Nam VN  
Virgin Islands, British VG  
Virgin Islands, U.S. VI  
Wallis and Futuna WF  
Western Sahara EH  
Yemen YE  
Yugoslavia YU  
Zambia ZM  
Zimbabwe ZW



## SSH Host Key Manager

SSH host key is used for SFTP connection (not for the SSH public key authentication).

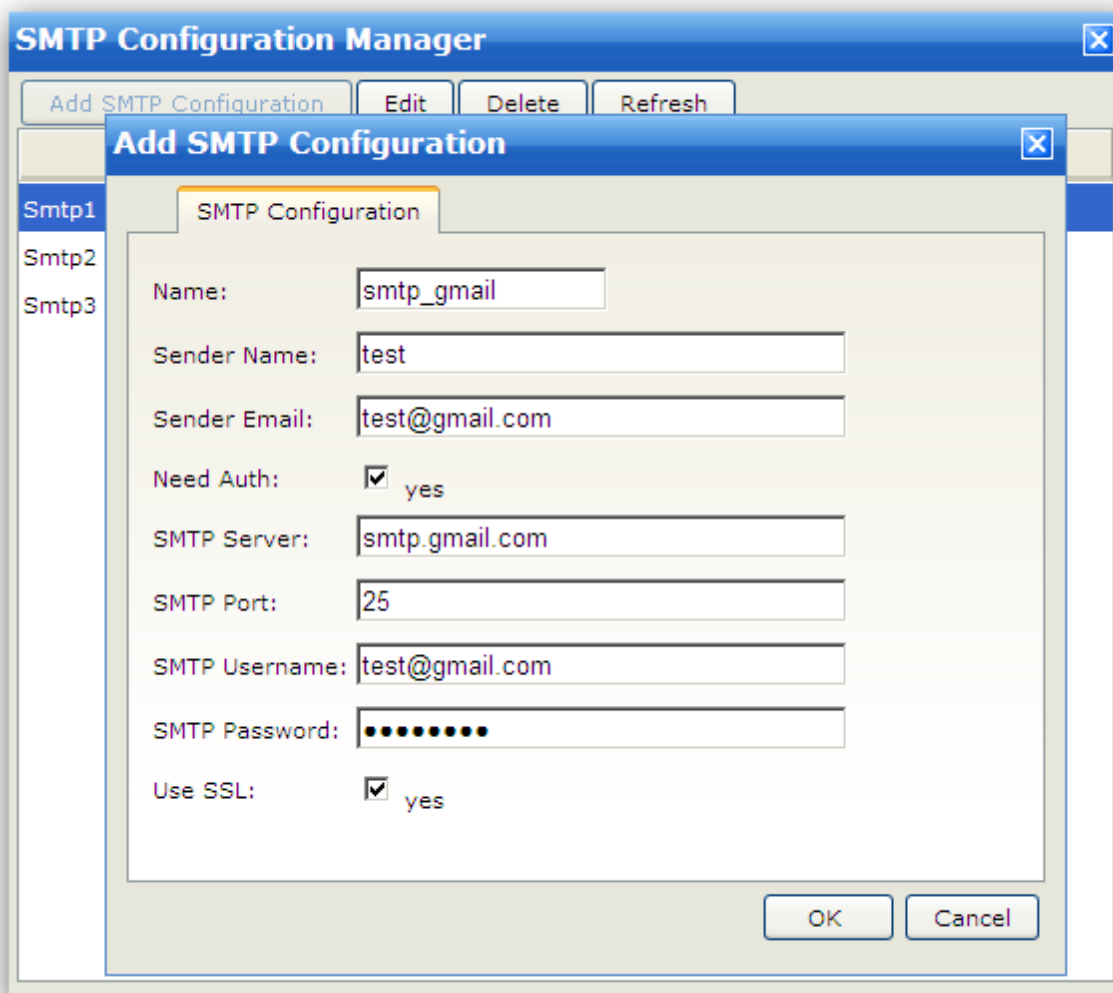


Click "Add SSH Host Key" button to add an SSH host key. First you need to specify a key name, then you can choose a SSH host key from your hard disk. If you do not have any SSH host key, just click the "Create Host Key File" button and create a SSH host key then.



## SMTP Server Manager

You can add/modify SMTP configuration under "SMTP Server Manager", it can be used to send email in Web Client or Lua script.



Click the "Add SMTP Configuration" button to add an SMTP server settings.

**Name:**

The identifier for you to distinguish this SMTP server from other SMTP servers.

**Sender Name:**

FTP server will use this as the sender's name.

**Sender Email:**

FTP server will use this as the sender's email address.

**Need Auth:**

If your SMTP server needs authentication, just enable this option and input the username/password.

**SMTP Server:**

The server address of this SMTP server. It should be a domain name (e.g., smtp.mail.com) or an IP address (e.g., 231.56.789.24).

**SMTP Port:**

The listener port of this SMTP server, default SMTP port is 25.

**SMTP Username:**

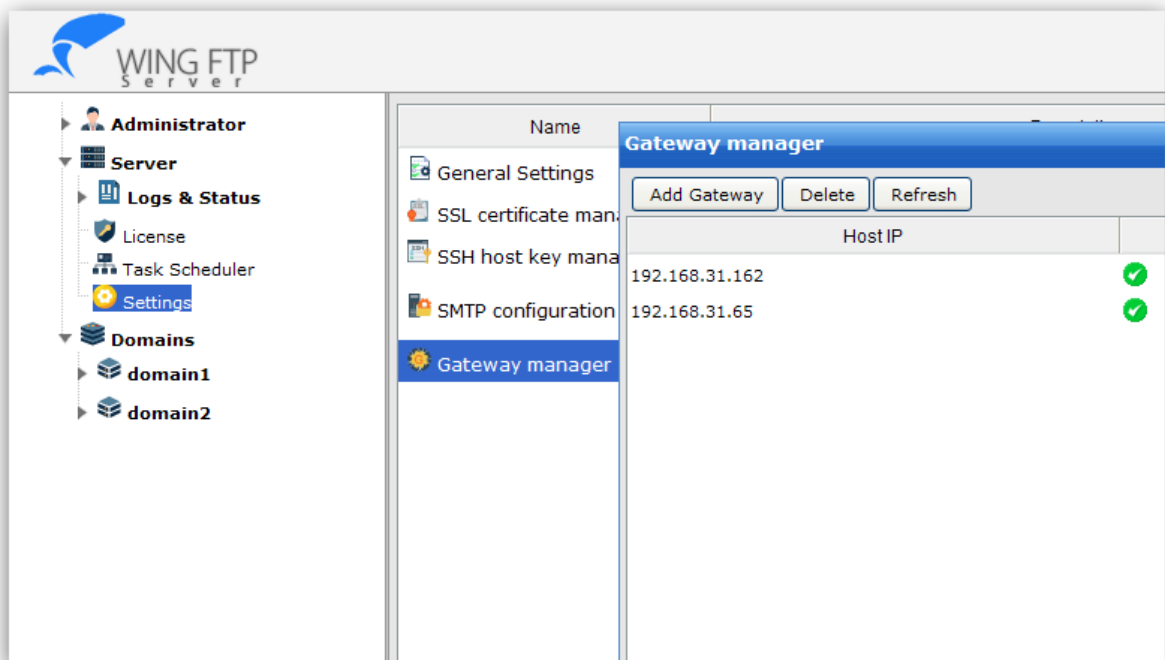
The account name is associated with authentication for the SMTP server.

**Use TLS/SSL:**

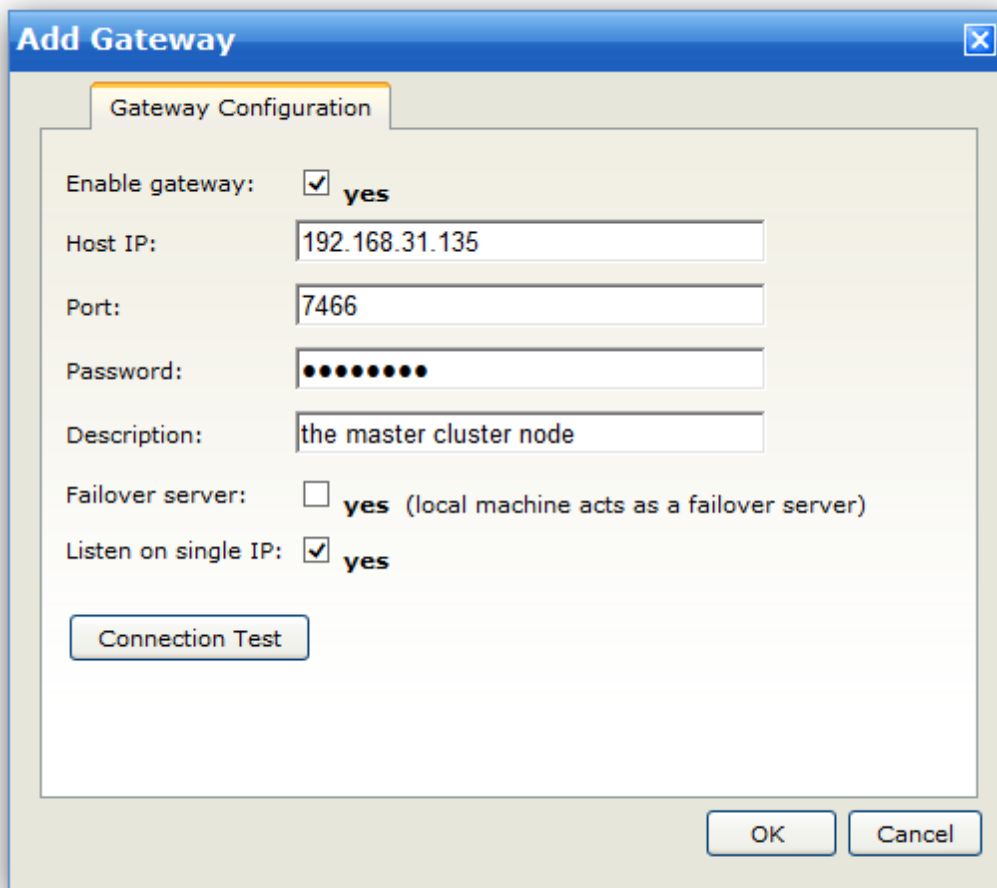
Whether the SMTP server requires TLS/SSL connection.

**Gateway Manager**

Since Wing FTP Server v5.1.2, we released a new product - Wing Gateway, it can provide reverse proxy and load balance service for Wing FTP Server, the gateway software needs to be downloaded and purchased separately, and you usually also need to install it another server machine (located in the DMZ). To add a new gateway server, just navigate to "Server -> Settings -> Gateway manager," and click on the button "Add Gateway":



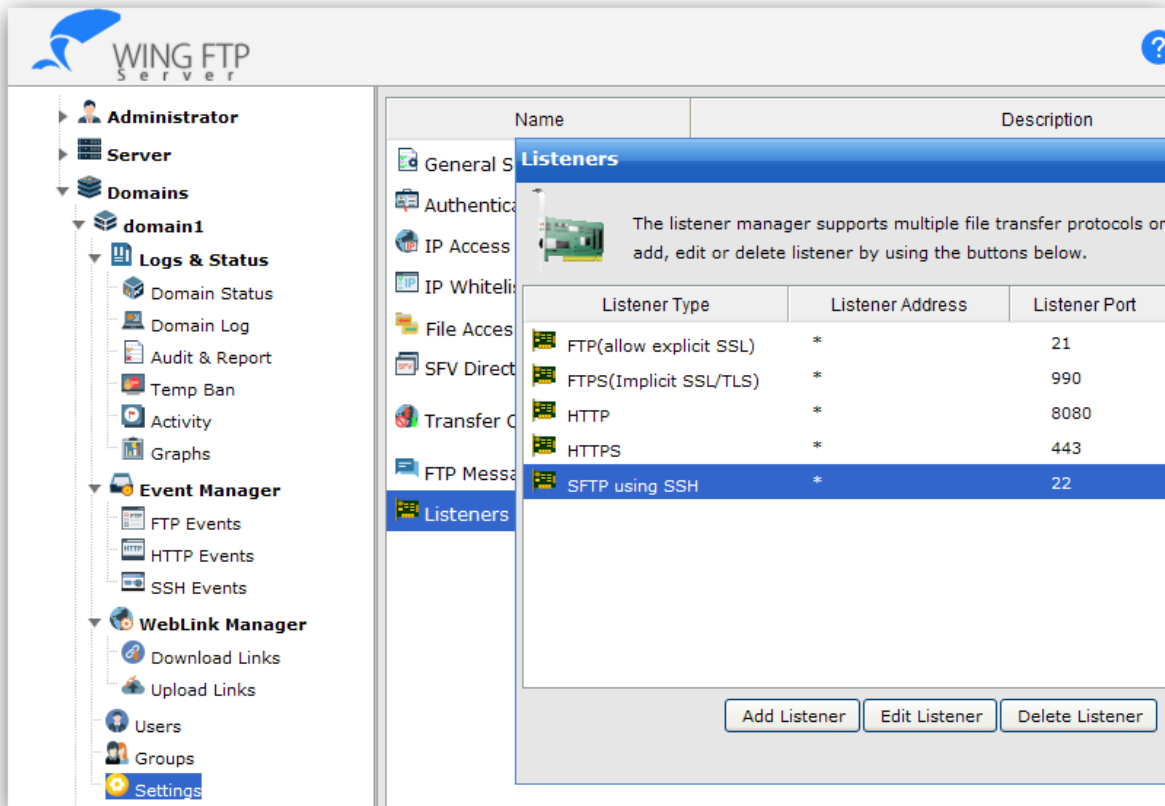
And just input a few necessary fields like "Host IP," "Port," and "Password," and then click "OK":



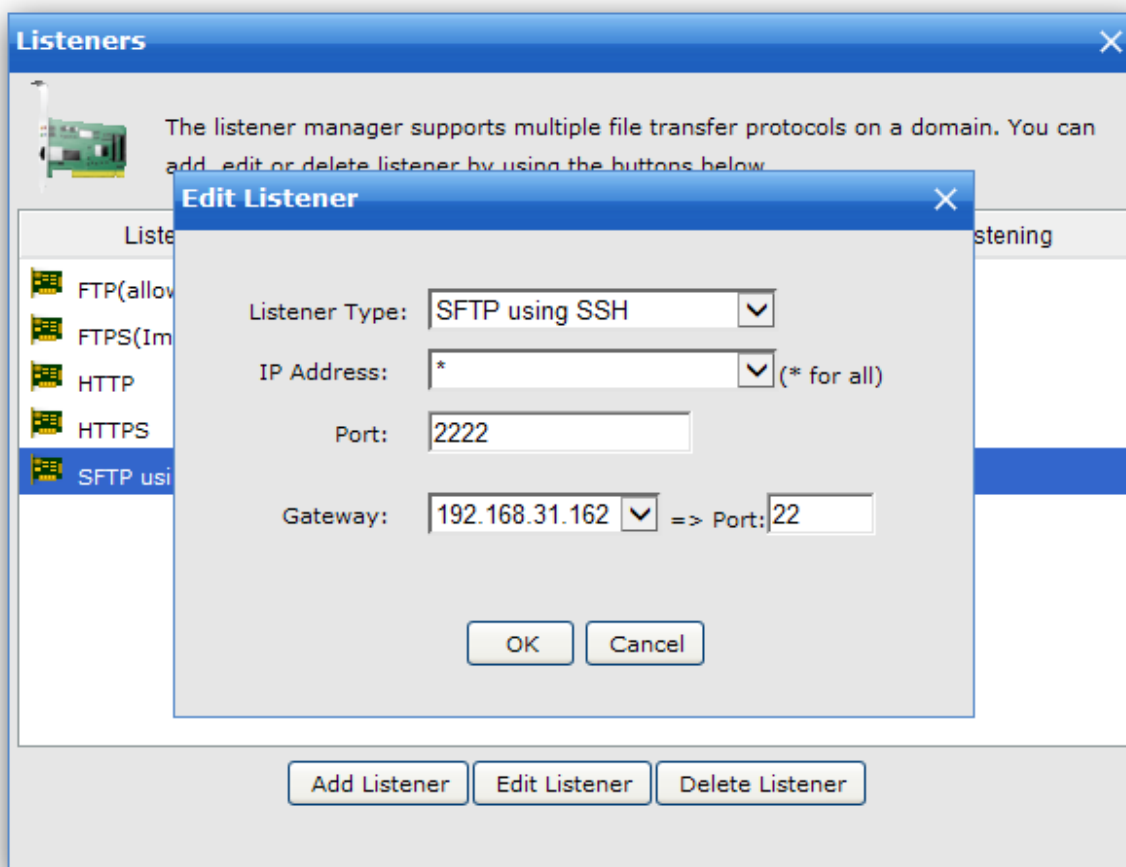
In the above screenshot, you can configure the following settings:

- **Enable gateway:** enable or disable this gateway
- **Host IP:** the IP address of this gateway
- **Port:** the port of this gateway, default communication port is 7466. Please make sure Wing FTP server can access this port, and also make sure Wing Gateway can reach Wing FTP server.
- **Password:** the communication password to that gateway
- **Description:** the description note
- **Failover server:** if you enable this checkbox, then the current Wing FTP machine will play as a failover server. It will standby until the master node fails or goes down.
- **Listen on single IP:** if the gateway server has multiple IP addresses, FTP/HTTP/SFTP listener will listen on the Host IP only when this option is enabled.
- **Connection Test:** click this button to verify whether gateway settings are correct

After adding the Wing Gateway, you need to add the listener proxy for the gateway, just navigate to "Domain -> Settings -> Listeners":



Double click on the existed listener (or add listener if no listener exists), then select the gateway host and input a port which gateway will be listening:

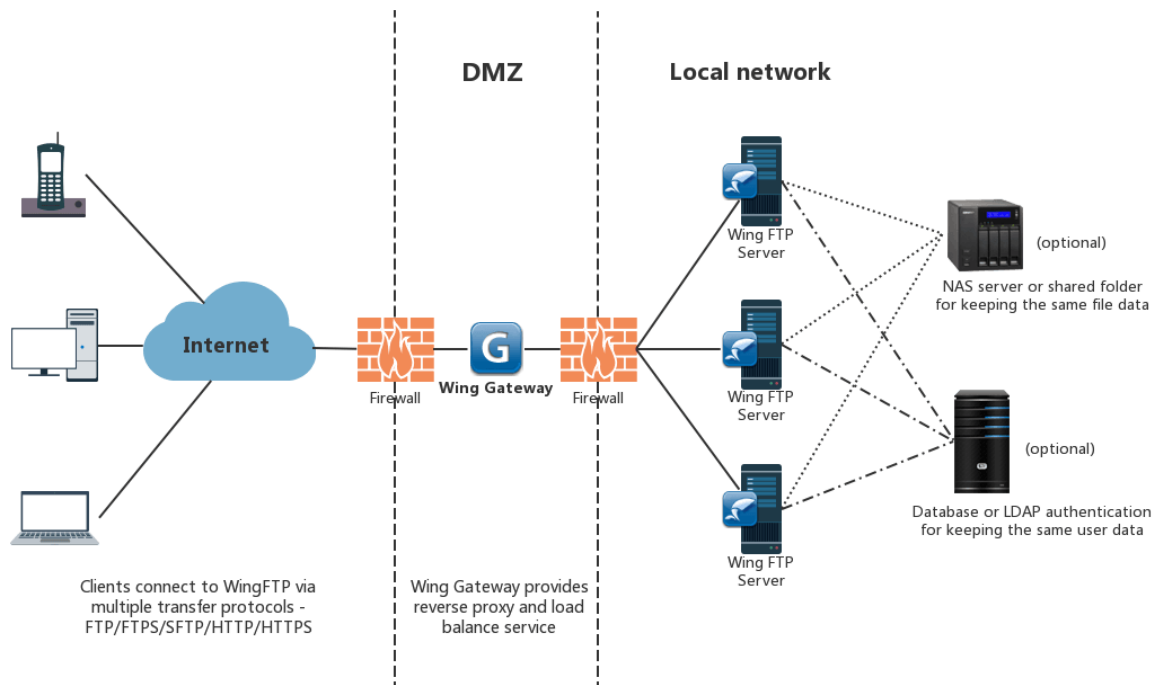


As the above screenshot settings, now Wing Gateway (192.168.31.162) will listen and accept connections on its port 22, and forward all the connections (on port 22) to Wing FTP server:2200.

That's all things about Wing Gateway settings! Very easy, right?

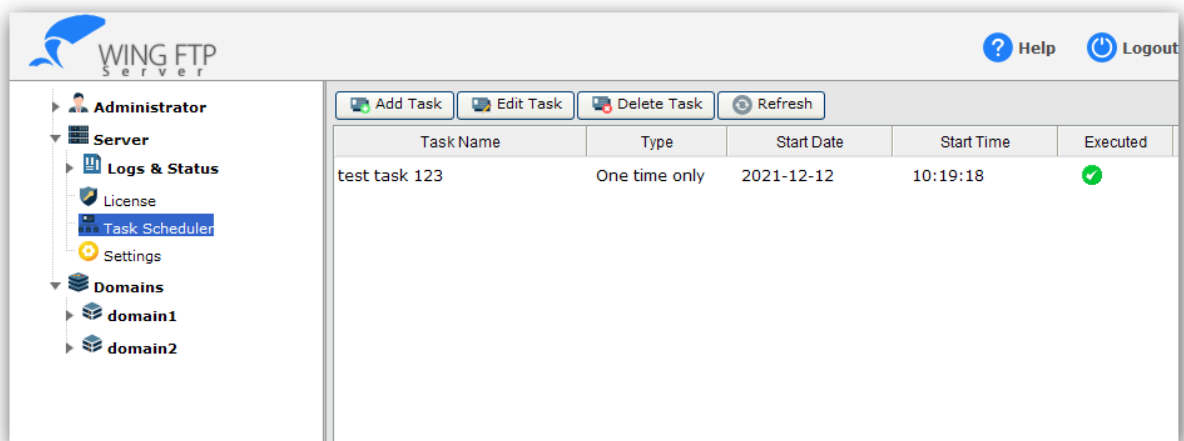
So if you have several Wing FTP server machines, and want to make an FTP server cluster, you just need to add the same settings on another Wing FTP server machine, then Wing Gateway will distribute incoming connections equally among multiple Wing FTP server machines.

Additionally, for keeping the same user data in all the cluster nodes, you need to use database or LDAP authentication, and for keeping the same file data in all the cluster nodes, you need to use NAS server or shared folders. So here is an example of deployment diagram for Wing FTP server cluster with Wing Gateway:



## Task Scheduler

Task Scheduler is used to take action at a specified time.



Click the button "Add Task," then you can define a task by Lua scripting. For example, you can let the server start or stop at a specified time by adding a task scheduler.

### Add Task



**Add Task**

Task Editor

Task Name: Test Task 123

Type: One time only

Start Date: 2021-12-12

Start Time: 11:20:55

Lua Script:

Common API

```
c_KickAllSessions("domain1")
c_AddSystemLog("Kick all the sessions", 1)
```

Lua API Help

OK Cancel

Task Name: The identifier for this task  
From Date: The task will start after this date  
Start Time: The task will start after this time  
Type: One Time/Once Hour/Once Day/Once Week/Once Month  
Lua Script: Lua script you want to run

## Logs & Status

### Domain Status

View the current domain status, the number of total sessions, download/upload speed, sent/received bytes, total running time, and so on.

View statistics about this domain[domain1]	
Statistics	Value
Domain Running Time	02:10:01
Sessions	0
Highest Num. Sessions	1
24 Hours Sessions	2
Avg. Session Length	00:05:03
Longest Session	00:10:05
Total Sessions	2
Download Speed	0 Bytes/s
Avg. Session Download Speed	0 Bytes/s
Total Download	0 Bytes
Total Download Files	0 files
Upload Speed	0 Bytes/s
Avg. Session Upload Speed	0 Bytes/s
Total Upload	0 Bytes
Total Upload Files	0 files

#### Sessions

The number of online sessions under this domain.

#### Highest Num Sessions

The highest number of concurrent sessions that have been recorded since being placed online.

#### 24 Hrs Sessions

The number of sessions that have connected in the past 24 hours.

#### Average Session Length

The average staying time under this domain.

#### Longest Session

The longest staying time for a session under this domain.

#### Total Sessions

The total number of sessions that have connected since being placed online.

### **Download Speed**

Cumulative download bandwidth currently used.

### **Avg. Session Download Speed**

The average session download bandwidth used since being placed online.

### **Total Download Bytes**

The total amount of data downloaded since being placed online.

### **Total Download Files**

The total number of files downloaded since being placed online.

### **Upload Speed**

Cumulative upload bandwidth currently used.

### **Avg. Session Upload Speed**

The average session upload bandwidth used since being placed online.

### **Total Upload Bytes**

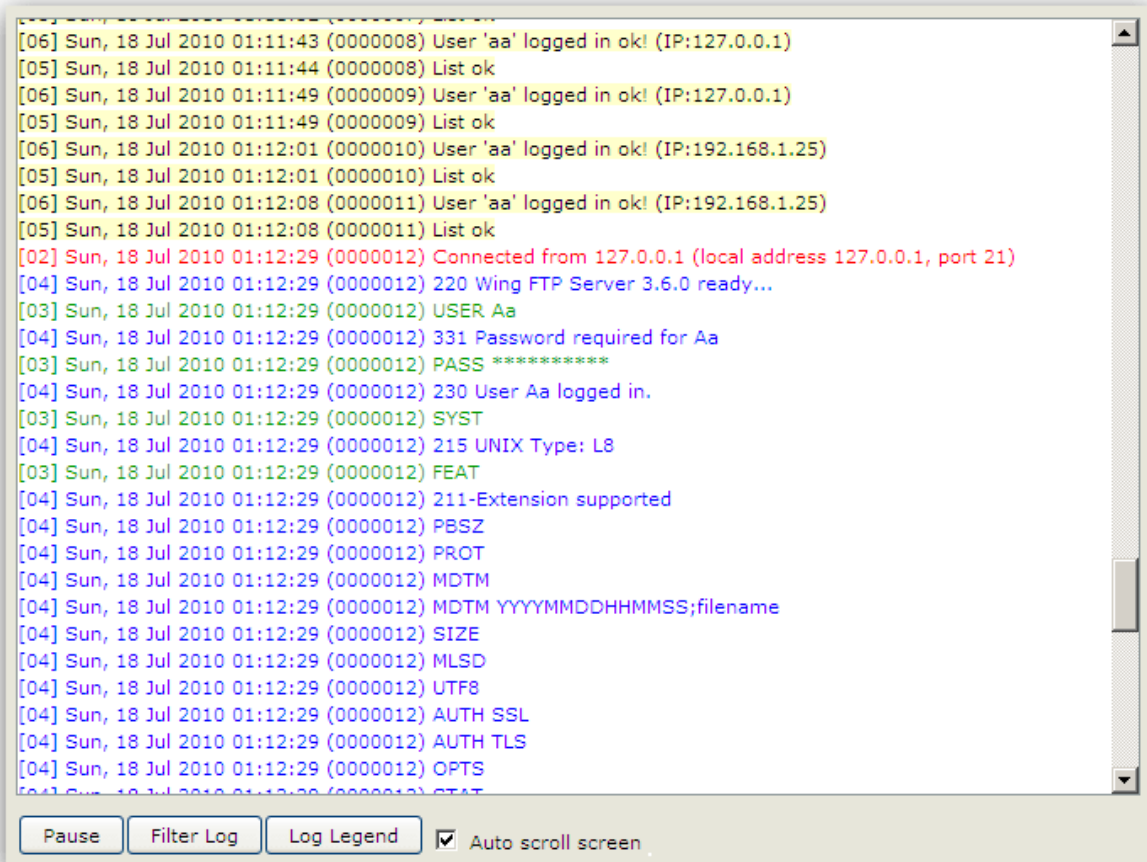
The total amount of data uploaded since being placed online.

### **Total Upload Files**

The total number of files uploaded since being placed online.

## Domain Log

View the domain logs here.



```
[06] Sun, 18 Jul 2010 01:11:43 (0000008) User 'aa' logged in ok! (IP:127.0.0.1)
[05] Sun, 18 Jul 2010 01:11:44 (0000008) List ok
[06] Sun, 18 Jul 2010 01:11:49 (0000009) User 'aa' logged in ok! (IP:127.0.0.1)
[05] Sun, 18 Jul 2010 01:11:49 (0000009) List ok
[06] Sun, 18 Jul 2010 01:12:01 (0000010) User 'aa' logged in ok! (IP:192.168.1.25)
[05] Sun, 18 Jul 2010 01:12:01 (0000010) List ok
[06] Sun, 18 Jul 2010 01:12:08 (0000011) User 'aa' logged in ok! (IP:192.168.1.25)
[05] Sun, 18 Jul 2010 01:12:08 (0000011) List ok
[02] Sun, 18 Jul 2010 01:12:29 (0000012) Connected from 127.0.0.1 (local address 127.0.0.1, port 21)
[04] Sun, 18 Jul 2010 01:12:29 (0000012) 220 Wing FTP Server 3.6.0 ready...
[03] Sun, 18 Jul 2010 01:12:29 (0000012) USER Aa
[04] Sun, 18 Jul 2010 01:12:29 (0000012) 331 Password required for Aa
[03] Sun, 18 Jul 2010 01:12:29 (0000012) PASS *****
[04] Sun, 18 Jul 2010 01:12:29 (0000012) 230 User Aa logged in.
[03] Sun, 18 Jul 2010 01:12:29 (0000012) SYST
[04] Sun, 18 Jul 2010 01:12:29 (0000012) 215 UNIX Type: L8
[03] Sun, 18 Jul 2010 01:12:29 (0000012) FEAT
[04] Sun, 18 Jul 2010 01:12:29 (0000012) 211-Extension supported
[04] Sun, 18 Jul 2010 01:12:29 (0000012) PBSZ
[04] Sun, 18 Jul 2010 01:12:29 (0000012) PROT
[04] Sun, 18 Jul 2010 01:12:29 (0000012) MDTM
[04] Sun, 18 Jul 2010 01:12:29 (0000012) MDTM YYYYMMDDHHMMSS;filename
[04] Sun, 18 Jul 2010 01:12:29 (0000012) SIZE
[04] Sun, 18 Jul 2010 01:12:29 (0000012) MLSD
[04] Sun, 18 Jul 2010 01:12:29 (0000012) UTF8
[04] Sun, 18 Jul 2010 01:12:29 (0000012) AUTH SSL
[04] Sun, 18 Jul 2010 01:12:29 (0000012) AUTH TLS
[04] Sun, 18 Jul 2010 01:12:29 (0000012) OPTS
[04] Sun, 18 Jul 2010 01:12:29 (0000012) STAT
```

Pause Filter Log Log Legend  Auto scroll screen

### Pause

Click this button to pause the log refreshing temporarily. This is useful on busy systems, so a certain section of the log can be highlighted and copied before it is scrolled out of view. Once finished, click this button again to resume the log refreshing.

### Filter Log

To quickly find the related domain logs.

### Audit & Report

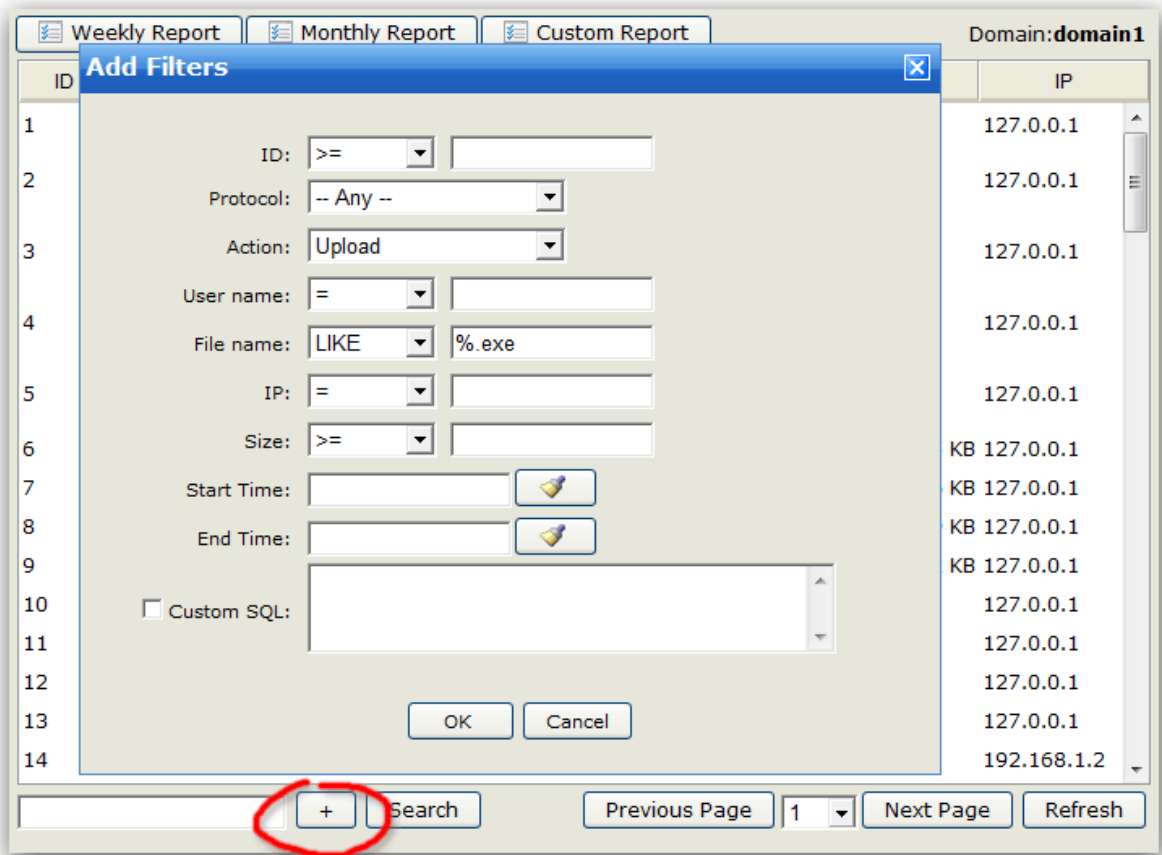
If you enable the option "Audit & Report" in the system settings, Wing FTP Server will record all the transactions into an SQLite database, and then you can analyze it and generate reports in real-time.

Weekly Report   Monthly Report   Custom Report   Domain: domain1

ID	Protocol	Action	User name	Time	File name	Size	IP
14901	HTTP	Download	aa	2013-04-17 16:56:25	COPYING	24.42 KB	127.0.0.1
14902	HTTP	Download	aa	2013-04-17 16:56:25	CPackConfig.cmal	1.9 KB	127.0.0.1
14903	HTTP	Download	aa	2013-04-17 16:56:25	CTestConfig.cmal	259 B	127.0.0.1
14904	HTTP	Download	aa	2013-04-17 16:56:25	DefineOptions.cm	0.87 KB	127.0.0.1
14905	HTTP	Download	aa	2013-04-17 16:56:25	authentication.dc	13.16 KB	127.0.0.1
14906	HTTP	Download	aa	2013-04-17 16:56:25	CMakeLists.txt	60 B	127.0.0.1
14907	HTTP	Download	aa	2013-04-17 16:56:25	command.dox	2.3 KB	127.0.0.1
14908	HTTP	Download	aa	2013-04-17 16:56:25	doxy.config.in	63.25 KB	127.0.0.1
14909	HTTP	Download	aa	2013-04-17 16:56:25	doxy.trac.in	63.28 KB	127.0.0.1
14910	HTTP	Download	aa	2013-04-17 16:56:25	forwarding.dox	6.39 KB	127.0.0.1
14911	HTTP	Download	aa	2013-04-17 16:56:25	guided_tour.dox	13.76 KB	127.0.0.1
14912	HTTP	Download	aa	2013-04-17 16:56:26	introduction.dox	1.65 KB	127.0.0.1
14913	HTTP	Download	aa	2013-04-17 16:56:26	linking.dox	0.78 KB	127.0.0.1
14914	HTTP	Download	aa	2013-04-17 16:56:26	mainpage.dox	5.55 KB	127.0.0.1
14915	HTTP	Download	aa	2013-04-17 16:56:26	scp.dox	7.09 KB	127.0.0.1
14916	HTTP	Download	aa	2013-04-17 16:56:26	sftp.dox	10.94 KB	127.0.0.1
14917	HTTP	Download	aa	2013-04-17 16:56:26	shell.dox	10.17 KB	127.0.0.1
14918	HTTP	Download	aa	2013-04-17 16:56:26	std.dox	10.18 KB	127.0.0.1

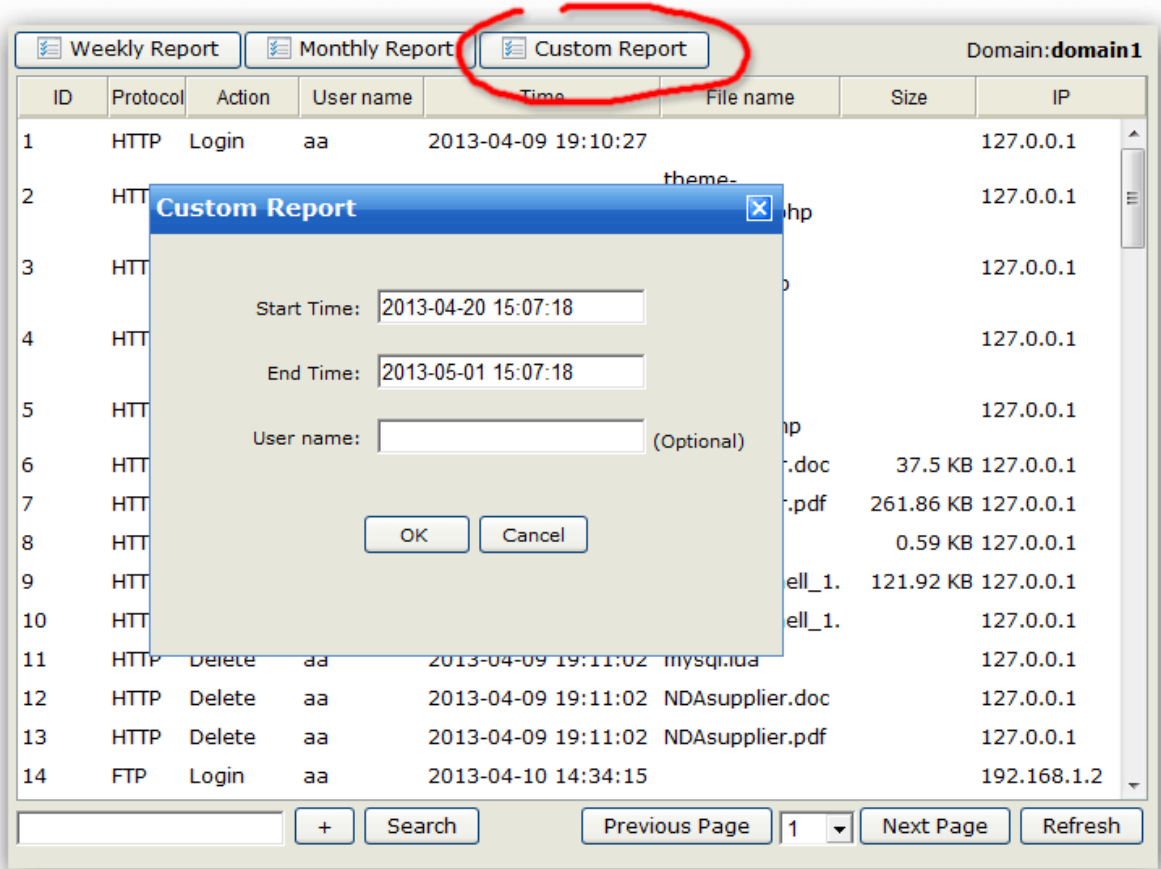
+   Search   Previous Page   150   Next Page   Refresh

In the above interface, you can view all the transactions page by page, and you can also add filters to display the data that meet the filtering criteria (just need to click the button "+" ).



For a "LIKE" filter, you can use the wildcards "%", "\*" and "\_" for the username, filename and IP address. "%" and "\*" matches zero or more characters, and "\_" matches a single character. And for the actions "Create Dir", "Delete Dir", "Rename", the field "File name" means the full file path, like "d:/download/backup/wingftp".

You can generate "Weekly Report" "Monthly Report" and "Custom Report", and for the custom report, you can specify the start time, the end time, or the username.



Finally, you will get a statistical report with general statistics and top downloaded/uploaded files, top logged in IP addresses, and top logged in user accounts. You can print the report out or save it as an HTML file.

**Monthly Report**

**[domain1] Monthly Report**  
From 2013-04-02 15:13:49 To 2013-05-02 15:13:49

[Print](#) [Save As](#)

Statistics	
Total Sessions	14867
Unique IP	8
Total Download Files	122
Total Upload Files	8
Total Download	24.29 MB
Total Upload	549.99 KB
Total Download Files (Weblink)	5
Total Download (Weblink)	9.83 MB

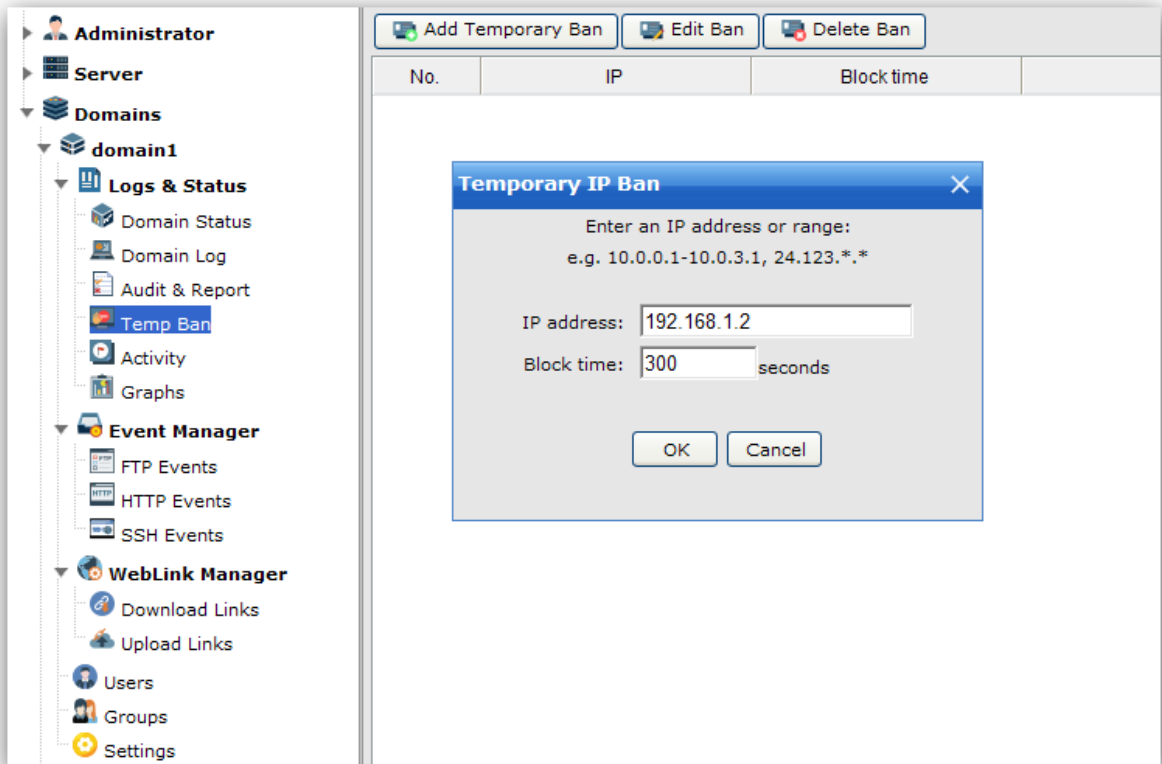
  

Top 10 Downloads	
File name	Count



## Temp Ban

You can add, edit, or delete a Temporary IP Ban rule in the below interface.



## Activity

You can see all the online sessions here.

To view the detailed information and access logs on a specific session, just double-click on the session.

Domain: <b>domain1</b> Online Sessions: 5						
No.	Protocol	User name	IP address	Last command	Last directory	File
3	HTTP	aa	192.168.31.65	CREATE THUMB	/	
5	SSH(SFTP)	aa	127.0.0.1	SSH_FXP_CLOSE	/react-3.0	
6	FTPS(EXPLICIT)	aa	192.168.31.65	PWD	/	
8	HTTP	<WEBLINK>	192.168.31.65	DOWNLOAD FILE	/	wftpservice-linu 64bit_new.tar.
9	HTTP	<WEBLINK>	192.168.31.65	DOWNLOAD FILE	/	WingGateway- linux-64bit.tar.

### Kick

Immediately kick the selected session, and there are some options available:

- Disconnect - Immediately disconnects the session.
- Disconnect and ban IP - Immediately disconnects the session and bans their IP address for the specified number of minutes, preventing them from immediately reconnecting.
- Disconnect and block IP permanently - Immediately disconnects the session and adds a deny IP access rule for their IP address, preventing them from ever reconnecting from the same IP address.

Disable user account

Check on this option to disable the user account, disabled accounts remain on the server but cannot log in.

Kill the sessions from the same IP address

If the selected session's IP is 192.168.31.100, then check on this option will kill all the connections from 192.168.31.100.

### Kick All

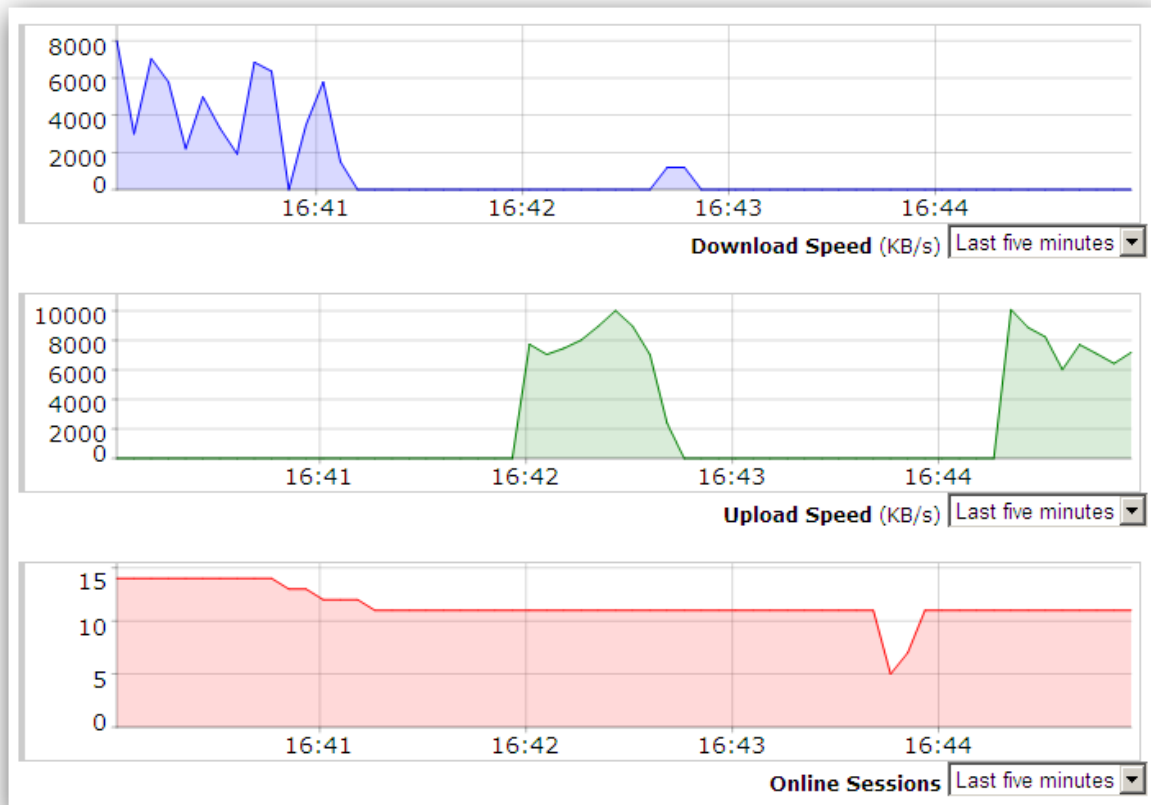
Kick all the online sessions immediately.

### View Log

View the detailed information and access logs on a selected session.

### Graphs

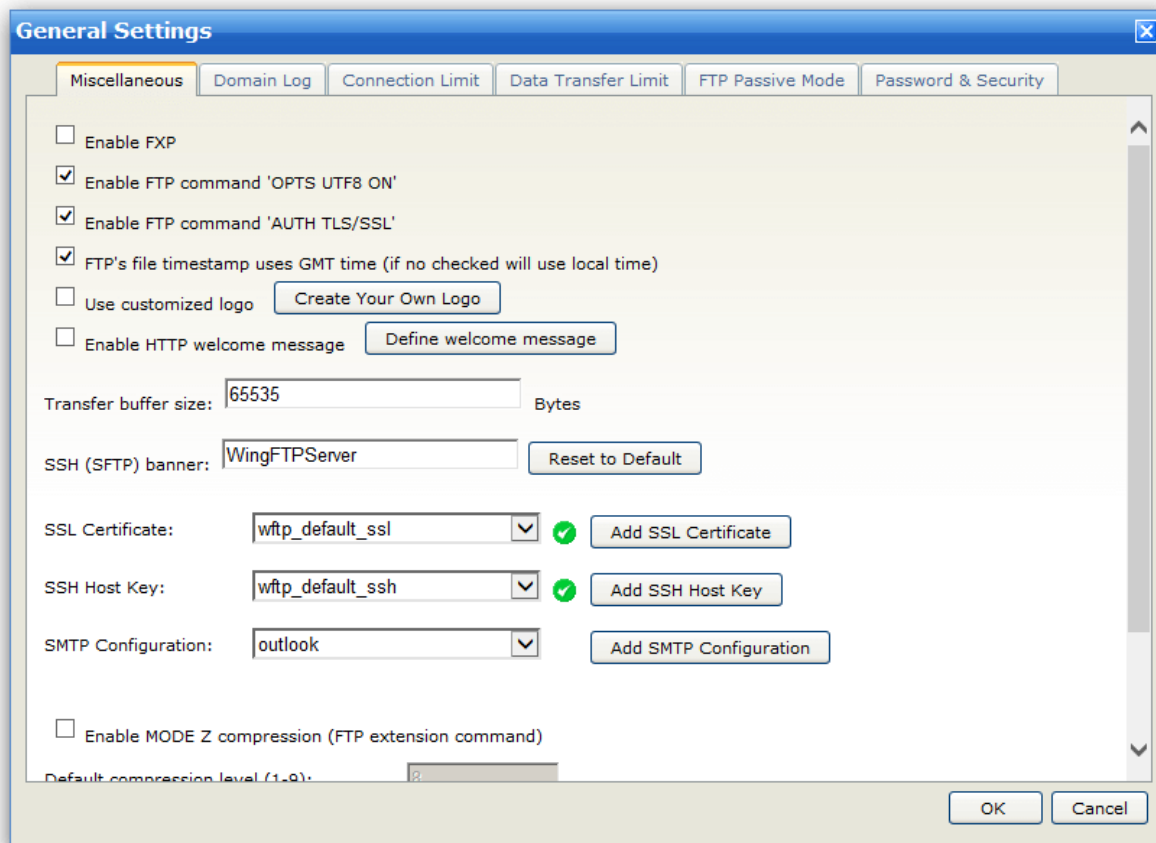
This graph chart reports domain traffics of the last 5 minutes, 5 hours or ten days.



## Domain Settings

### General Settings

#### Miscellaneous



#### Enable FXP

Activate or deactivate FTP transfer from server to server.

#### Enable FTP command 'OPTS UTF8 ON'

You can disable it to improve Unicode compatibility when using Windows Explorer.

#### Enable FTP command 'AUTH TLS/SSL.'

Allow explicit FTP over TLS/SSL.

#### File timestamp using GMT time

When enabled, file listing uses GMT timestamp. If not checked, the local time will be used instead.

#### Use customized logo

You can specify a customized logo to be displayed on the Web Client interface, only for this domain.

#### Enable HTTP welcome message

You can enable this option and define the welcome message, then the logged in HTTP(S) session will see the message.

**Transfer buffer size**

Transfer buffer size used to transfer data (default is 65535). You can increase this value if you want the server to read/write less often from/to the hard drive when clients transfer.

**SSH (SFTP) banner**

You can define your own SSH (SFTP) banner here.

**SSL Certificate**

Select an SSL certificate for this domain. You can manage your SSL certificates in "Server -> Settings -> SSL Certificate Manager". SSL certificate is used for encrypting data exchanges between the client and the server. Without certificates, you can not establish TLS/SSL connections (FTPS or HTTPS).

**SSH Host Key**

Select an SSH host key for this domain. You can manage your SSH host keys at "Server -> Settings -> SSH Host Key Manager". An SSH host key is used for the SFTP connections.

**SMTP Configuration**

Select an SMTP configuration for this domain. You can manage your SMTP configurations at "Server -> Settings -> SMTP Configuration Manager". SMTP configuration is used for sending emails.

**Enable MODE Z support**

Activate or deactivate FTP transfer compression.

**Default compression level**

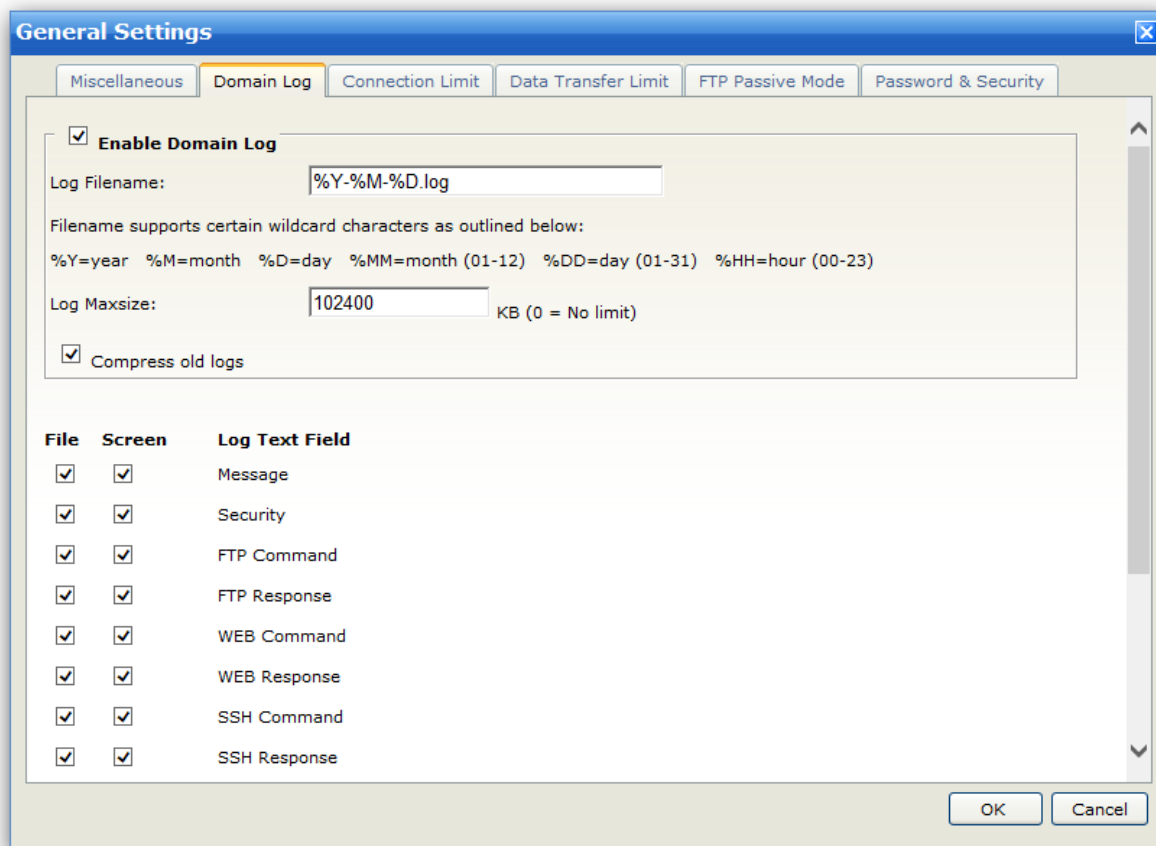
If FTP Client does not specify the compression level, the server will use this value as default.

**Maximum and Minimum allowed compression level**

The server only allows the compression level between the minimum value and maximum value.

## Domain Log

Here you can configure domain log settings.



### Log Filename

All the domain log files will be stored under the folder "Log/YourDomain", and you can define the custom filename for the log files, you can also use the following wildcard characters for filename:

*%Y - Current year*

*%M - Current month, 1-12*

*%D - Current day, 1-31*

*%MM - Current month, 01-12*

*%DD - Current day, 01-31*

*%HH - Current hour, 00-23*

### Log Maxsize

The log file will be ignored if it reaches the limit you defined.

### Compress old logs

The log of the previous day will be automatically compressed into a zip file for saving the disk space.

### File & Screen Logging Options

Choose types of logs to be stored in the log files or to be displayed under "Domain > Logs & Status > Domain Log".

**Connection Limit**

The screenshot shows the 'General Settings' dialog box with the 'Connection Limit' tab selected. The dialog has a title bar with a close button (X) and a tabbed interface with the following tabs: Miscellaneous, Domain Log, Connection Limit (selected), Data Transfer Limit, FTP Passive Mode, and Password & Security. The main content area is divided into two sections:

- Domain-specific options:(0 = No limit)**
  - Maximum number of sessions on domain: 0
  - Maximum sessions per IP address on domain: 0
- Options below can be overridden by user-level or group-level:(0 = No limit)**
  - Maximum number of sessions per user account: 0
  - Maximum sessions per IP address for user account: 0
  - Automatic idle connection timeout: 0 minutes
  - Automatic transfer connection timeout: 0 minutes

Below these sections, there is a checked checkbox for 'HTTP/HTTPS session keep-alive' and a text input field for 'TLS session timeout' set to '3600' seconds. At the bottom right, there are 'OK' and 'Cancel' buttons.

**Max number of sessions**

Limit the number of sessions logged on this domain simultaneously.

**Max session Per IP**

Limit the number of sessions from the same IP.

**Max sessions per user account**

Specifies the maximum number of concurrent sessions that may be opened from a single user account.

**Max sessions per IP per user account**

Specifies the maximum number of concurrent sessions that a user may open from a single IP address.

**Automatic idle connection timeout (for FTP/SFTP protocol)**

When a client has been idle for a specific time, it will be disconnected automatically.

**Automatic transfer connection timeout (for FTP/SFTP protocol)**

When no data transfer for a specific time, client will be disconnected automatically.

**HTTP/HTTPS session keep-alive**

The HTTP session will keep-alive automatically by default, when you disable this option, the HTTP session will expire after 10 minutes of inactivity.

## TLS session timeout

Default value is 3600 seconds, it means TLS session can be reused in 3600 seconds to avoid a full handshake.

## Data Transfer Limit

The screenshot shows the 'General Settings' dialog box with the 'Data Transfer Limit' tab selected. The dialog has several sections for configuring data transfer limits and Weblink options.

- Domain-specific options:(0 = No limit)**
  - Maximum download speed for domain: 0 KB/s
  - Maximum upload speed for domain: 0 KB/s
- Options below can be overridden by user-level or group-level:(0 = No limit)**
  - Maximum download speed per session: 0 KB/s
  - Maximum upload speed per session: 0 KB/s
  - Maximum download speed for user accounts: 0 KB/s
  - Maximum upload speed for user accounts: 0 KB/s
- Weblink options:**
  - Enable file download link
  - Enable file request link
  - Show subfolders for Weblink download
  - Keep Weblink when file is deleted or renamed
  - Overwrite the existed file when uploading file via file request link
- The domain URL:  e.g. http://YourDomain/

Buttons for 'OK' and 'Cancel' are located at the bottom right of the dialog.

### Max Domain Download Speed

Max download speed for this domain.

### Max Domain Upload Speed

Max upload speed for this domain.

### Max Session Download Speed

Max download speed for a single session.

### Max Session Upload Speed

Max upload speed for a single session.

### Max User Download Speed

Max download speed for a single user.

### Max User Upload Speed

Max upload speed for a single user.

**Enable file download link**

When enabled, the client is able to generate a weblink for file sharing (your client can download files via this link without login). You can also set the maximum download speed for file sharing.

**Enable file request link**

When enabled, the client is able to generate a upload link for requesting files (your client can upload files via this link without login).

**Show subfolders for Weblink download**

If you check on this option, the subfolders in the weblink download page will be shown and can be accessible.

**Keep Weblink when file is deleted or renamed**

When enabled, the weblink will be maintained after the file is overwritten.

**Overwrite the existed file when uploading file via file request link**

When enabled, it will overwrite the existed file for file request uploading.

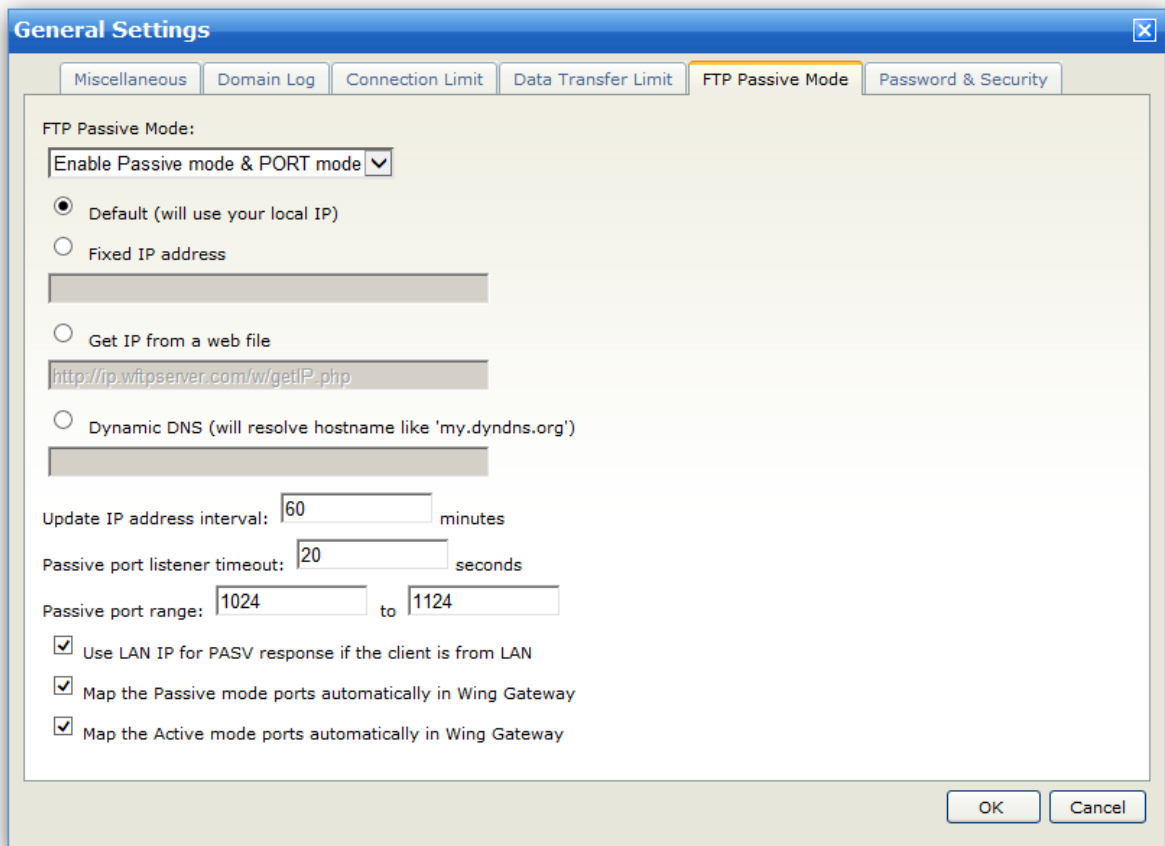
**The domain URL**

When you specify a fixed domain URL, the download/upload weblink will use this domain URL.

**FTP Passive Mode**

FTP passive mode configuration, you can also check out this KB article: <https://bbs.wftpserver.com/viewtopic.php?f=5&t=5>



**Default**

Use the local IP address.

**Fixed IP address**

The fixed IP address points to this server, and normally it is the fixed internet IP address.

**Get IP from a web file**

Get your internet IP from a web file, like: <http://ip.wftpserver.com/w/getIP.php>

**Dynamic DNS**

If you are assigned a dynamic IP and are using a service like dyndns.org (or other) to keep your domain name pointing to your IP.

**Update IP address interval**

If you get your IP from a web file or DNS, the server will update it in the specified time period.

**Passive port listener timeout**

Specify the timeout for closing the opened passive port listener.

**Passive port range**

The passive port range is a group of ports assigned for the passive mode, if the FTP server is behind a firewall or router, then you need to forward passive ports in the firewall/router manually.

If you have 100 concurrent FTP connections, you can open 100 passive ports, like 1024 - 1124.

**Use LAN IP for PASV response if the client is from LAN**

Always use the local IP address of the server machine if the client is from LAN.

### Map the Passive mode ports automatically in Wing Gateway

If you use Wing Gateway for the failover or load balancing cluster, and Wing Gateway is also in the DMZ zone, then checking on this option will be very helpful.

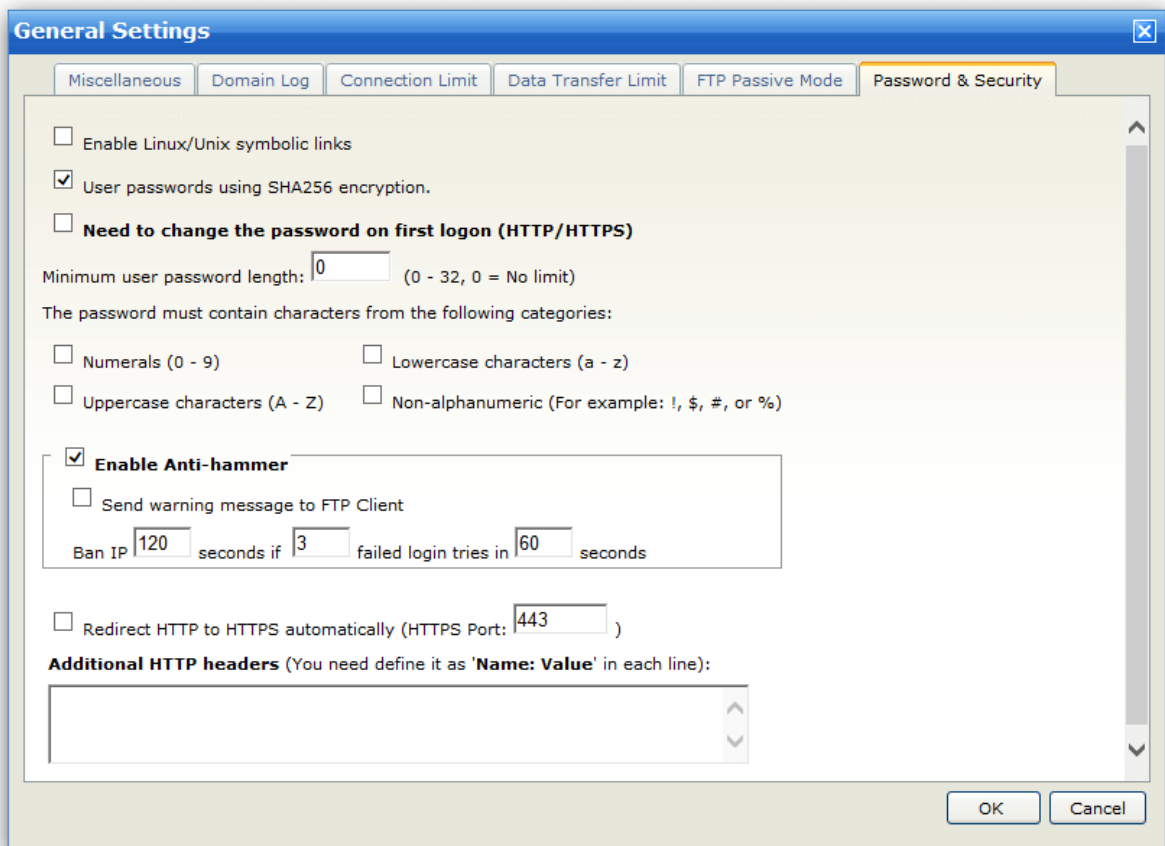
It will forward all the passive ports automatically in Wing Gateway. If you have two or more Wing FTP servers, you need to specify separate passive ports in each server, for example, Server A uses 1024 - 1124, Server B uses 2024 - 2124.

### Map the Active mode ports automatically in Wing Gateway

Like the above option, this option is disabled by default, which means Wing FTP server will connect to the FTP client directly for the Active mode transfer, when this option is enabled, then Wing FTP server will let Wing Gateway connect to the FTP client instead.

### Password & Security

Here you can configure settings for the password and security.



#### Enable Linux/Unix symbolic links

Allow or disallow file listing for symbolic links under Linux/Unix.

#### User passwords using SHA256 encryption

The user password will be stored with SHA256 hash by default, when this option is disabled, the user password will use MD5 hash.

### **Need to change the password on the first logon**

When enabled, the user needs to change his password on the first login. Only for web clients.

### **Minimum user password length**

It will restrict the user password length when adding a user account or changing the password.

### **The password must contain characters from the following categories**

You can specify the password complexity by choosing the following categories.

### **Enable Anti-hammer**

Activate or deactivate anti-hammering function.

### **Send hammering message to FTP client**

If a client tries to hammer this server, the server will send a message to him/her.

### **Ban IP xx seconds if xx failed login tries in xx seconds**

The counter will monitor the login tries and count them. If the number of failed login tries during the period is exceeded, the user IP will be banned for a period of time as defined.

### **Redirect HTTP to HTTPS automatically**

When enabled, all the HTTP will be redirected to the HTTPS protocol, and you can also specify the HTTPS listener port.

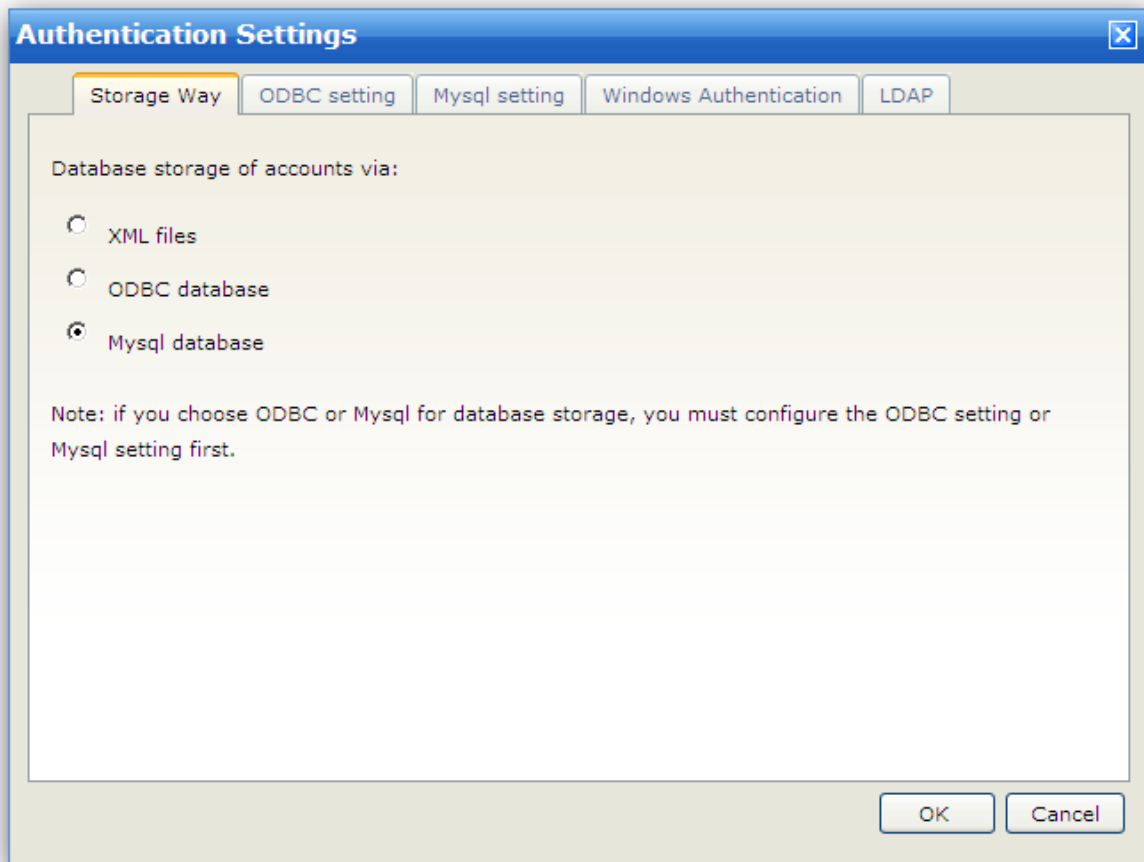
### **Additional HTTP headers**

You can define additional HTTP headers by yourself.

## Authentication Settings

Wing FTP Server supports several authenticating methods: XML files, ODBC database, MySQL database, Windows authentication (NTLM or Active Directory), and LDAP authentication.

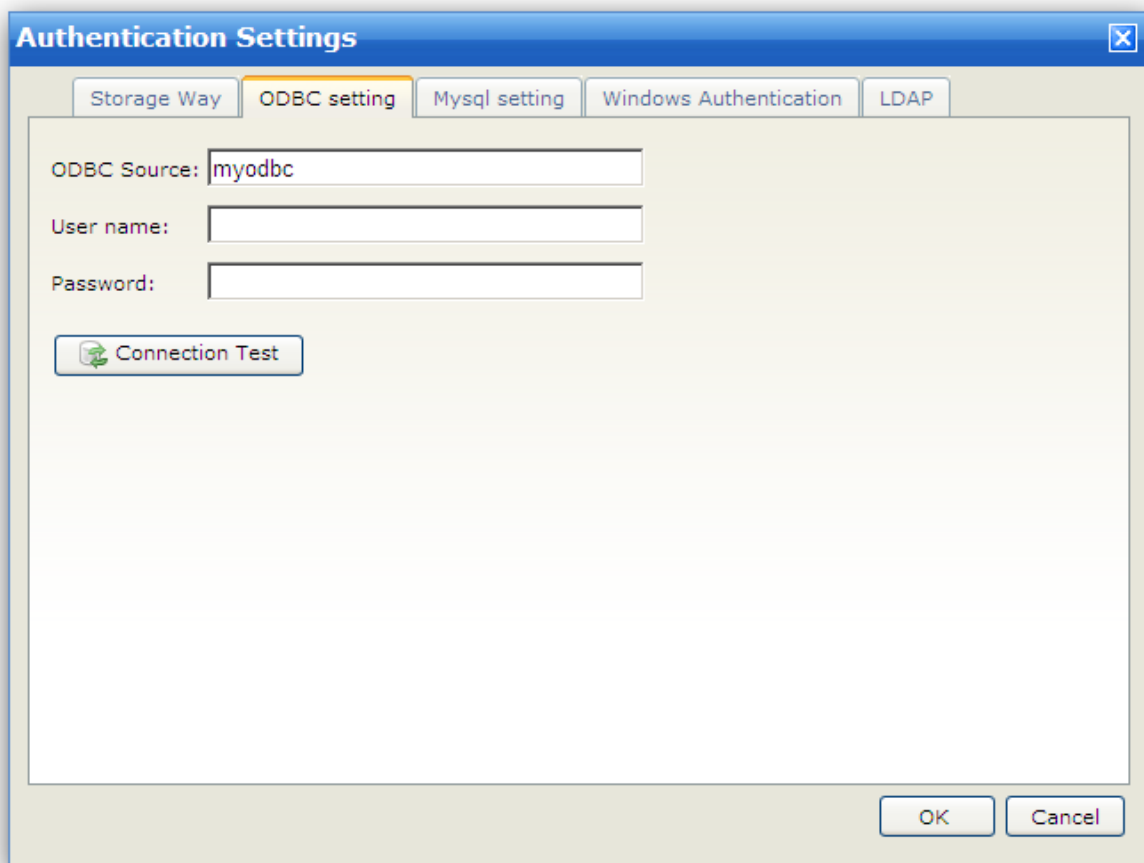
## Storage Way



XML: Store your user/group data in XML files.  
ODBC: Store your user/group data in ODBC database.  
MySQL: Store your user/group data in MySQL database.

**Note:** The server will automatically create all the database tables for the first time. If you use MySQL, please create a MySQL database first, the default database name is "wftp\_database".

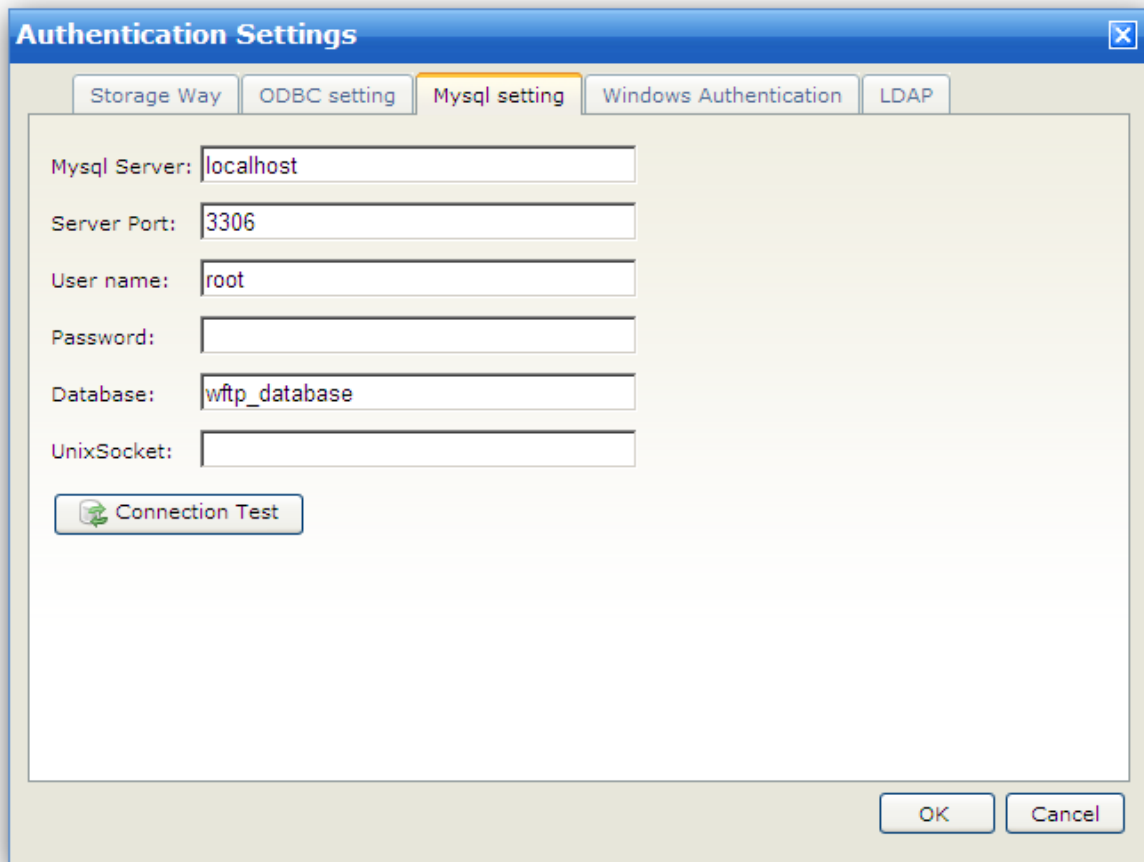
## ODBC setting



If you want to use ODBC database, just enter the ODBC source name (DSN), or username/password if you have.

When everything is fine, user/group data can be stored in this database through using an ODBC driver.

## Mysql setting

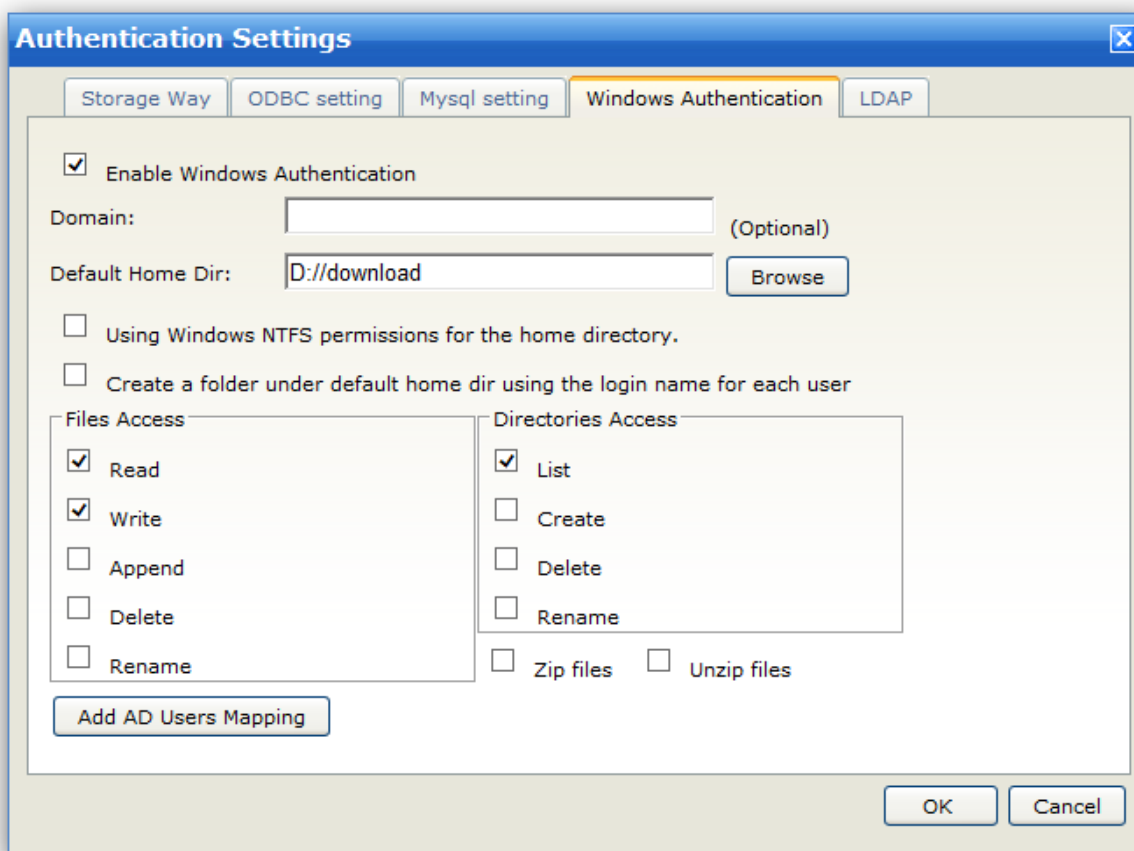


If your database system is MySQL, we suggest you use "MySQL" instead of "ODBC", because it can connect to MySQL database directly through C API, much faster than "ODBC".

You need to create a MySQL database "wftp\_database" for the first time.

You need to fill the field "UnixSocket" if you are using Linux/Unix system. Otherwise, just keep it empty.

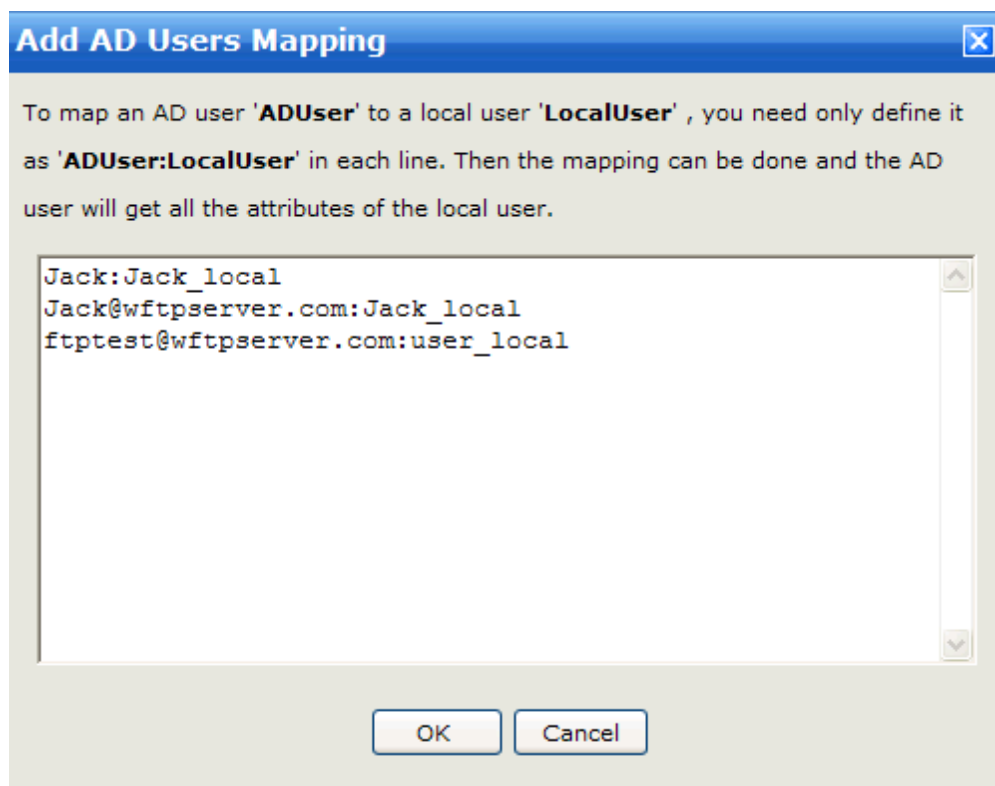
## Windows Authentication



Wing FTP Server can handle Windows NTLM or Windows Active Directory authentication, without the need to import users or sync user data manually.

If you enable "Windows Authentication", you need to specify a default home directory for the AD account. And if you want each AD account to have its own home directory, please enable the checkbox "Create a folder under default home dir using the login name for each user".

If you want to have more control to the AD user like a local user of WingFTP, you can map the AD user to a local user. First, create a local user at "Domains -> Users -> Add User", then click the button "Add AD Users Mapping", input a new line in the format of "**ADUser:LocalUser**" (AD username and Local username is separated by a colon). Since the version 6.3.2, the wildcard "\*" is supported for the Windows AD username, so you can write the mapping rule as: **Tester\*:LocalTester**

**NOTE:**

## 1) Domain Name

It's a string that specifies the name of the domain or server whose account database contains the AD account. If this field is empty, the user name must be specified in user principal name (UPN) format, user@DNS\_domain\_name. Otherwise, you can log in just with the username.

## 2) Map the AD account to local

For example:

AD user name: Jack@domain.com  
password: ad\_password

Local user name: Jack\_Local  
password: local\_password

After the mapping, you can log in the server with Jack@domain.com/ad\_password or Jack\_Local/local\_password. If you log in with Jack@domain.com/ad\_password, Jack@domain.com will have all the functions belong to the Jack\_Local like virtual directories, group memberships, permissions, and other settings.

Wing FTP server user authenticates sequence:

1) User "BOB" login with password "BOB2010".

2) Check the local user account to see if there is an account called "BOB".

2-1) if it exists, further checks the local password. If password is correct, your login is successful.

Otherwise, login has failed.

2-2) if the account does not exist, do AD authentication.

2-2-1) After completing the AD authentication, check if BOB has been mapped to a local user.

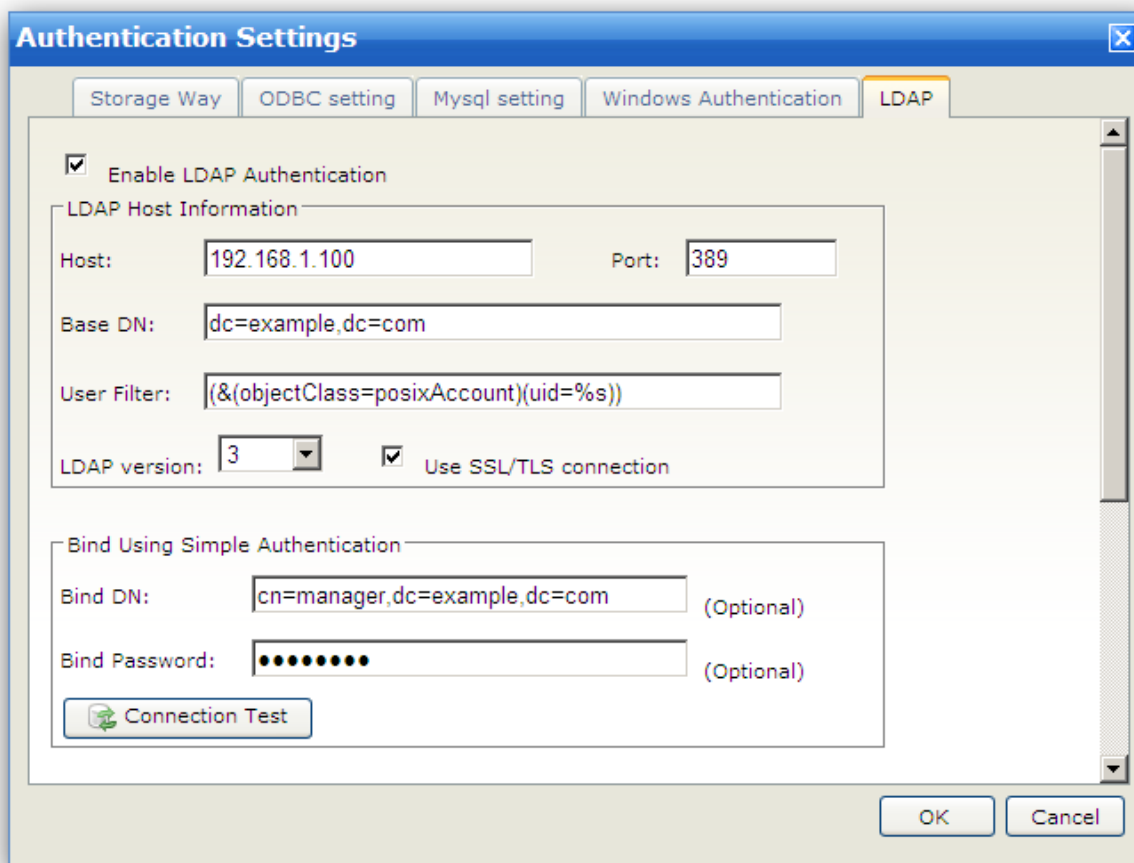
2-2-1-1) If "BOB" is mapped to a local user named "Local\_BOB", then it will get all the attributes of "Local\_BOB".



2-2-1-2) If there is no mapping for "BOB", take the AD authentication "Default Home Dir" as its home directory.

2-2-2) If the AD authentication fails, the login fails too.

## LDAP Authentication



Wing FTP Server supports using an LDAP(or LDAPS) database for authenticating users, and we have tested it successfully with OpenLDAP and Microsoft's Active Directory.

When you enable the option "LDAP Authentication", you need to provide the following information:

**Host:** the IP address or domain name of the LDAP server.

**Port:** the port number of the LDAP server, the default value is 389 (if you use Microsoft's Active Directory, please use the port 3268).

**Base DN:** the base domain name of search starting point. The DN string would usually be "**dc=xxxx,dc=com**".

**User Filter:** the filter to find the object for authenticating users. The special characters '%' in the filter string will be replaced with the real username.

Wing FTP Server uses default filter "*(&(objectClass=posixAccount)(uid=%s))*" for OpenLDAP. Or if you use

Microsoft's Active Directory, the filter string would usually be "`(&(objectClass=user)(sAMAccountName=%s))`".

**LDAP version:** the version of LDAP server will be 3 in normal cases.

**Use SSL/TLS connection:** you can enable it if the LDAP server supports SSL/TLS connection.

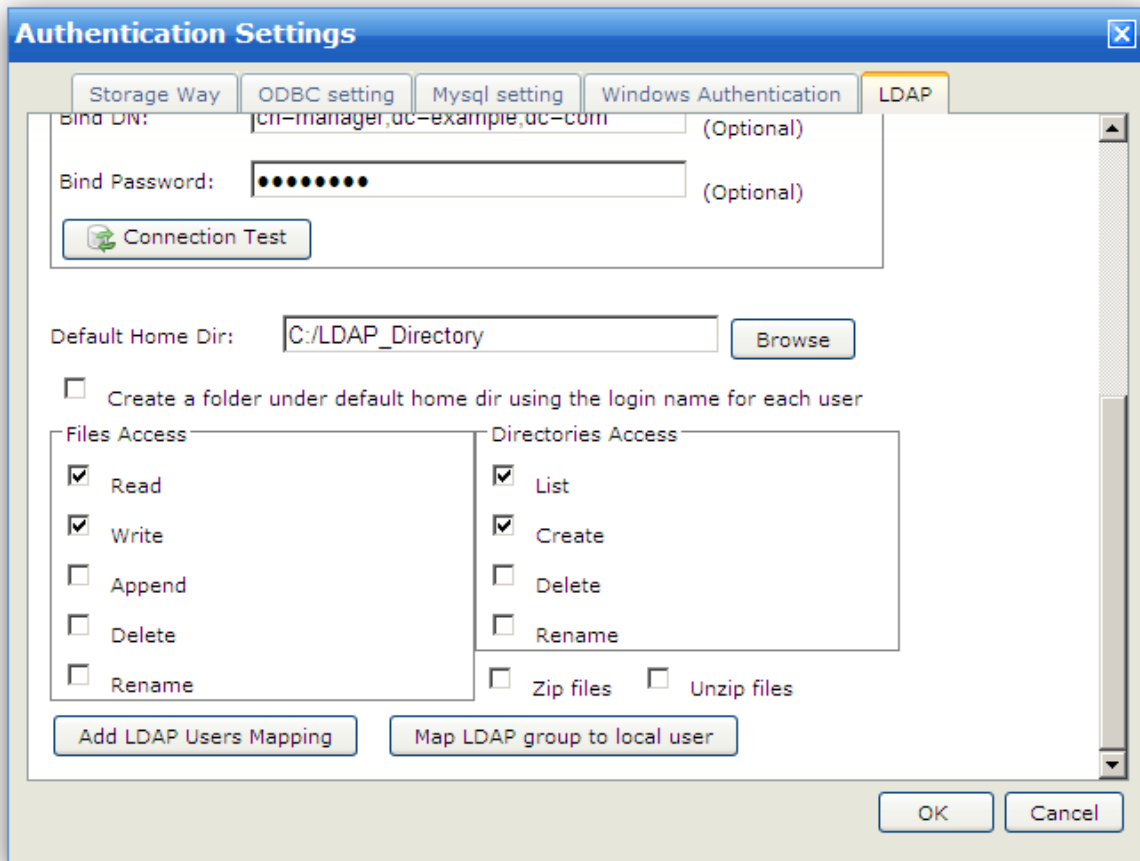
**Bind DN:** the LDAP distinguished name string for simple authentication, e.g. "`cn=manager,dc=example,dc=com`".

**Bind Password:** the bind password is the LDAP user password normally.

**Note1:** The LDAP database (except Windows Active Directory) must have an attribute 'userPassword' (RFC 3112), and the attribute 'userPassword' must be accessible. The following password format is supported: plain text, {crypt}, {md5}, {sha}, {smd5}, {ssh}

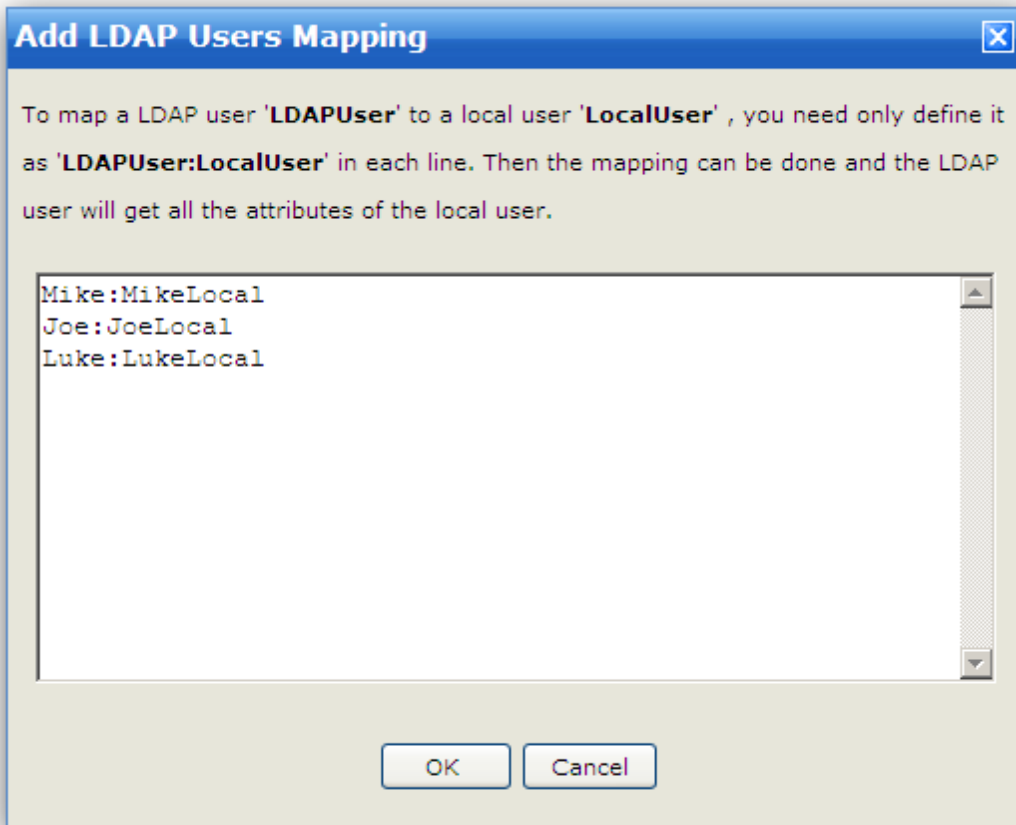
**Note2:** The username in the LDAP database cannot contain some special characters: '%', '\*', '?', '!', '"', '\'. For security purposes.

You need to specify a default home directory for the LDAP account. And if you want each LDAP account to have its own home directory, please enable check box "Create a folder under default home dir using the login name for each user":



If you want to have more control to LDAP user like a local user of WingFTP, you can map the LDAP user to a local user. First create a local user at "Domains -> Users -> Add User", then click the button "Add LDAP Users Mapping", input a new line in the format of "`LDAPUser:LocalUser`" (LDAP username and

Local username is separated by colon). Since the version 6.3.2, the wildcard "\*" is supported for the LDAP username, so you can write the mapping rule as: **Tester\*:LocalTester**



Map the LDAP account to the local account

For example:

LDAP username: Jack

password: ldap\_password

Local username: Jack\_local

password: local\_password

After the mapping, you can login the server with Jack/ldap\_password or Jack\_Local/local\_password. Then LDAP user "Jack" will have all the functions belong to the "Jack\_Local" like virtual directories, group memberships, permissions and other settings.

WingFTP user authenticates sequence:

1) User "JOE" login with password "JOE2010".

2) Check the local user account to see if there is a account called "JOE".

2-1) if it exists, further check the local password. If password is correct, your login is successful.

Otherwise, login has failed.

2-2) if the account does not exist, do LDAP authentication.

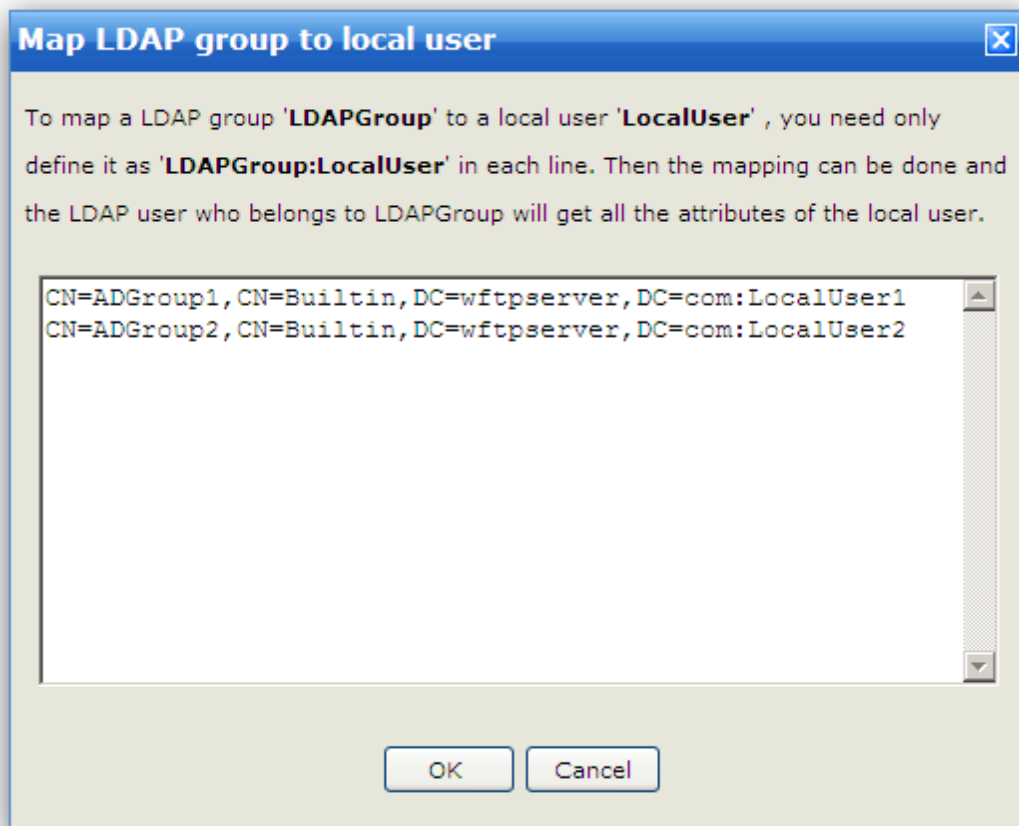
2-2-1) After completing the LDAP authentication, check if "JOE" has been mapped to a local user.

2-2-1-1) If "JOE" is mapped to a local user named "Local\_JOE", then it will get all the attributes of "Local\_JOE".

2-2-1-2) If there is no mapping for "JOE", take the LDAP authentication "Default Home Dir" as its home directory.

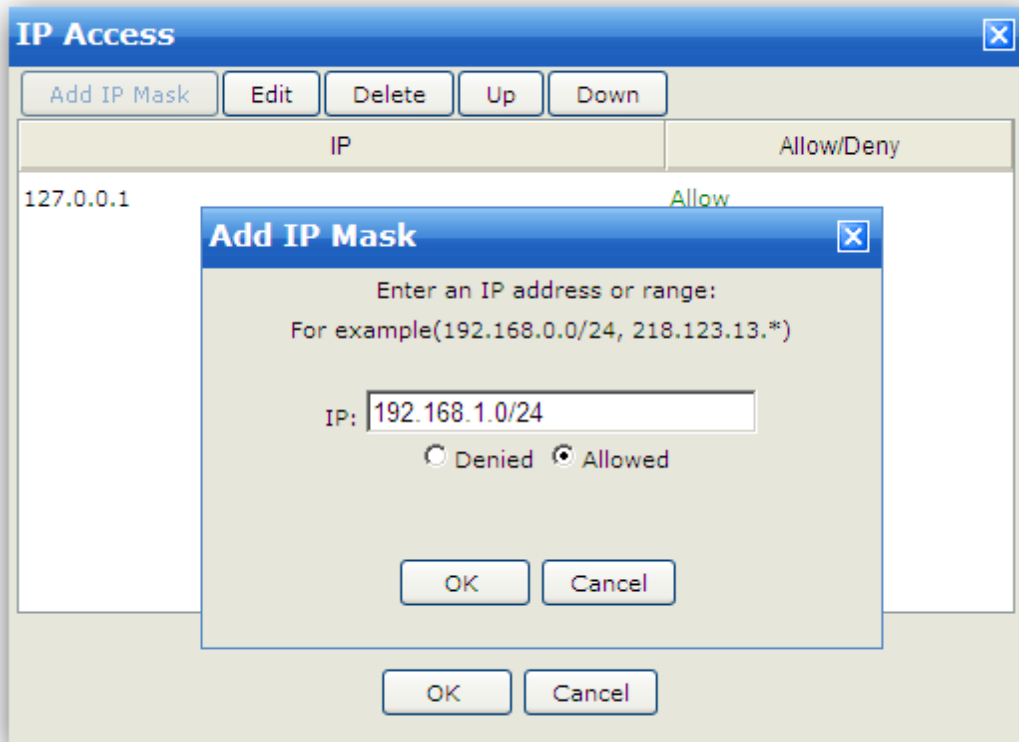
2-2-2) If the LDAP authentication fails, the login fails too.

You can also map LDAP group to local user by clicking the button "Map LDAP group to local user", input a new line in the format of "LDAPGroup:LocalUser"(LDAP group name and Local username is separated by colon), then all the LDAP users belong to this LDAP group can have local user's permissions.



## IP Access

Configure IP access rules to allow or deny access for this domain



If you do not specify an IP address, all the domain users can logon to the server from any IP address. But if you set allowed rules, the domain users can only obtain access by IP addresses specified in the allowed rules. If you set denied rules, the domain users can have access to the server by any IP address except those specified in the denied rules.

For example:

### **Allow 127.0.0.1**

Refuse all the domain users' connection from any IP except 127.0.0.1.

### **Supported wildcards**

IP address ranges and wildcards are supported by Wing FTP Server, as below:

xxx.xxx.xxx.xxx

IP address must be exactly matched(e.g. 192.168.1.1).

xxx.xxx.xxx.xxx-yyy

A specified range of IP addresses, e.g. 192.168.1.10-25.

xxx.xxx.xxx.xxx-yyy.yyy.yyy.yyy

A specified range of IP addresses, e.g. 192.168.1.0-192.168.5.255.

\*.\*.\* or xxx.\*.\* or xxx.xxx.\*.\* or xxx.xxx.xxx.\*

Any valid IP address value (For example, 192.168.\*.\* represents any IP between 192.168.0.0 and 192.168.255.255).

CIDR convention is also supported :

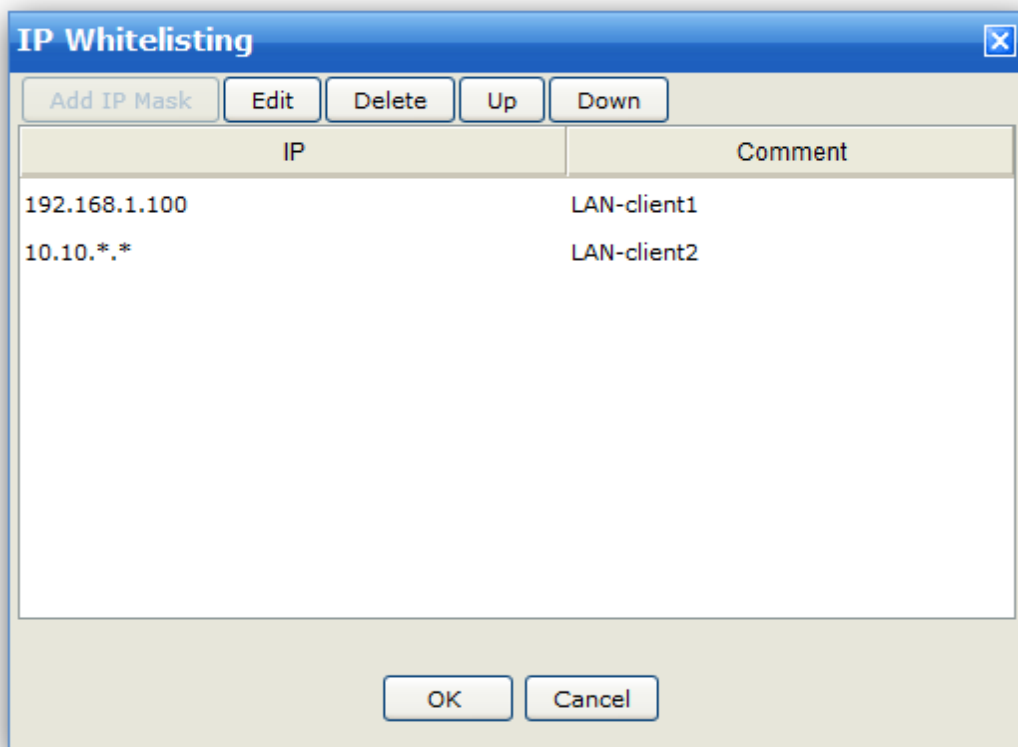
192.168.0.0/24 (represents any IP between 192.168.0.0 and 192.168.0.255)

## Priority

The Domain IP Access rules has priority over User IP Access rules.

## IP Whitelisting

IP whitelisting allows specific IPs to access server without any limitation.



## Supported wildcards

IP address ranges and wildcards are supported by Wing FTP Server, as below:

xxx.xxx.xxx.xxx

IP address must be exactly matched(e.g. 192.168.1.1).

xxx.xxx.xxx.xxx-yyy

A specified range of IP addresses, e.g. 192.168.1.10-25.

xxx.xxx.xxx.xxx-yyy.yyy.yyy.yyy

A specified range of IP addresses, e.g. 192.168.1.0-192.168.5.255.

\*.\*.\* or xxx.\*.\* or xxx.xxx.\*.\* or xxx.xxx.xxx.\*

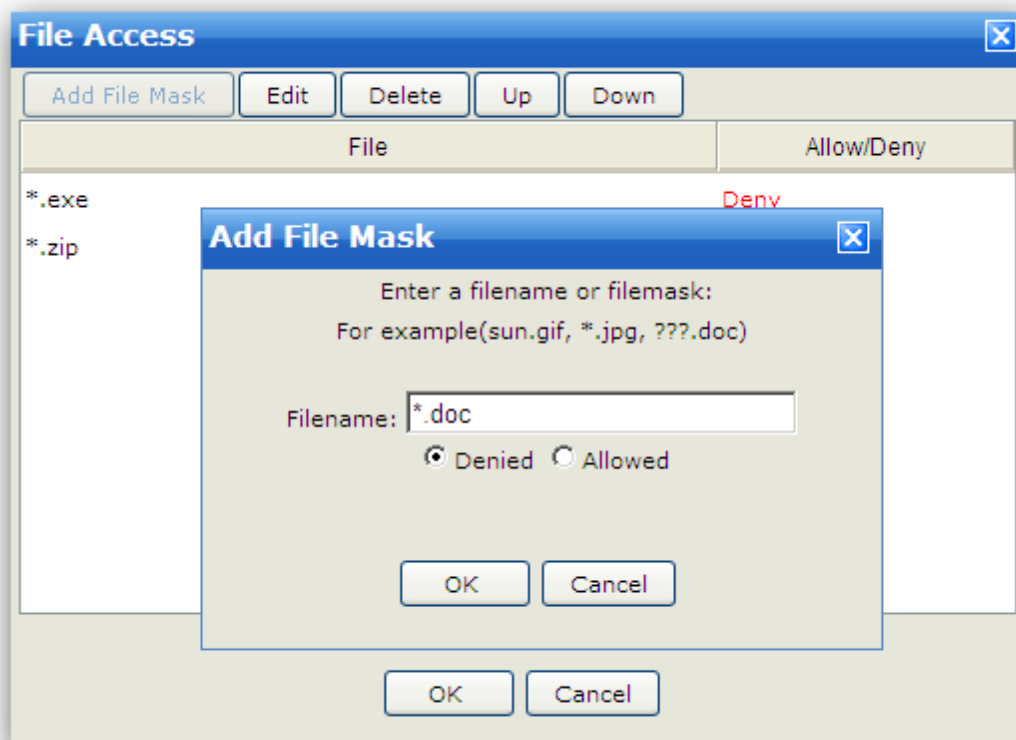
Any valid IP address value (For example, 192.168.\*.\* represents any IP between 192.168.0.0 and 192.168.255.255).

CIDR convention is also supported :

192.168.0.0/24 (represents any IP between 192.168.0.0 and 192.168.0.255)

## File Access

Configure file access rules to allow or deny access to this domain.



Banned files are files that can't be accessed on the server. You can specify file/path mask (?, \* supported) : \*.\*.jpg, c:\path\images\_200?\

Using this form, you can define deny/allow access based on Filename for this domain.

If you do not specify any file/path, all users can access all the files on the server. But if you set an allow list, all users can only access the files allowed in the list. If you set a deny list, all users can access the files except those in the deny list. The order of the rules is very important too.

For example

## Allow \*.rar

Can not access/store any file except \*.rar.

## Deny \*.zip

Can access/store any file except \*.zip.

## Rule list

The rule list shows the current list and order of file access rules. Rules can be added and removed from the list using the Add and Delete buttons.

Also, the order of the rules may be altered using the Up and Down buttons on the right of the rule list.

## Supported wildcards

To define access rules, you can use the wildcards ? and \*.

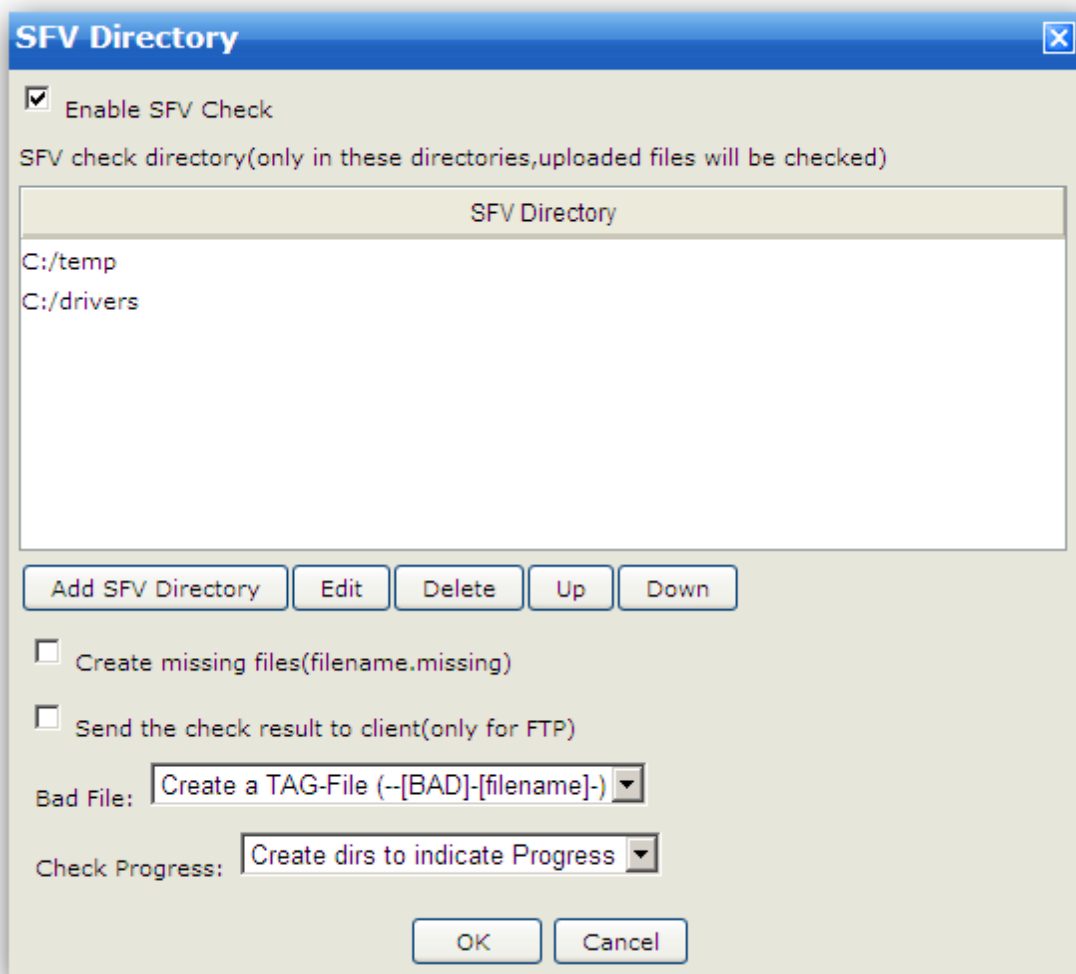
## Priority

The domain's File Access list has priority over the user account's File Access list.

## SFV Directory

Simple file verification (SFV) is a file format for storing CRC32 checksums of files in order to verify the integrity of files. SFV can be used to detect random corruptions in a file, but cannot be used for checking the authenticity in any meaningful way. Typically, the .sfv extension is used on SFV files. SFV Check is used to check the files the user uploaded. If you enable this option, when a client uploads an SFV file, the server will start to check the files listed in the SFV file and send the result to the client. The client can create SFV files by [QuickSFV](#).



**Add SFV Directory**

The server only checks SFV files in the SFV Directory. So if you want to enable SFV Check, you need to add SFV directory first.

**Edit**

Edit the selected directory.

**Delete**

Delete the selected directory.

**Create missing files(filename.missing)**

If the files listed in the SFV file can not be found in the same directory with the SFV file, missing files (filename.missing) will be created.

**Send the check result to client(only for FTP)**

FTP clients can receive the checker's response message.

**Bad File**

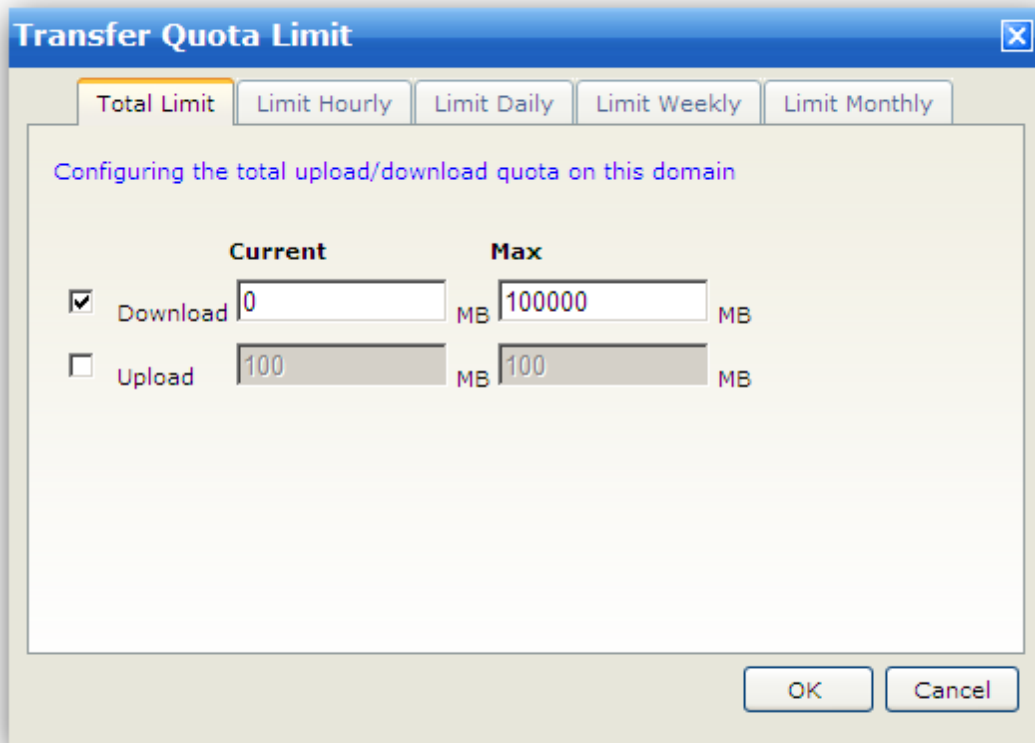
What to do when the server encounters a bad file. You can choose "Create a Tag File," "Rename the bad

file," or "Delete the bad file."

### Check Progress

How to indicate the check progress. You can choose "Create dirs to indicate progress," "Create files to indicate progress," or "Do not indicate progress."

### Transfer Quota Limit

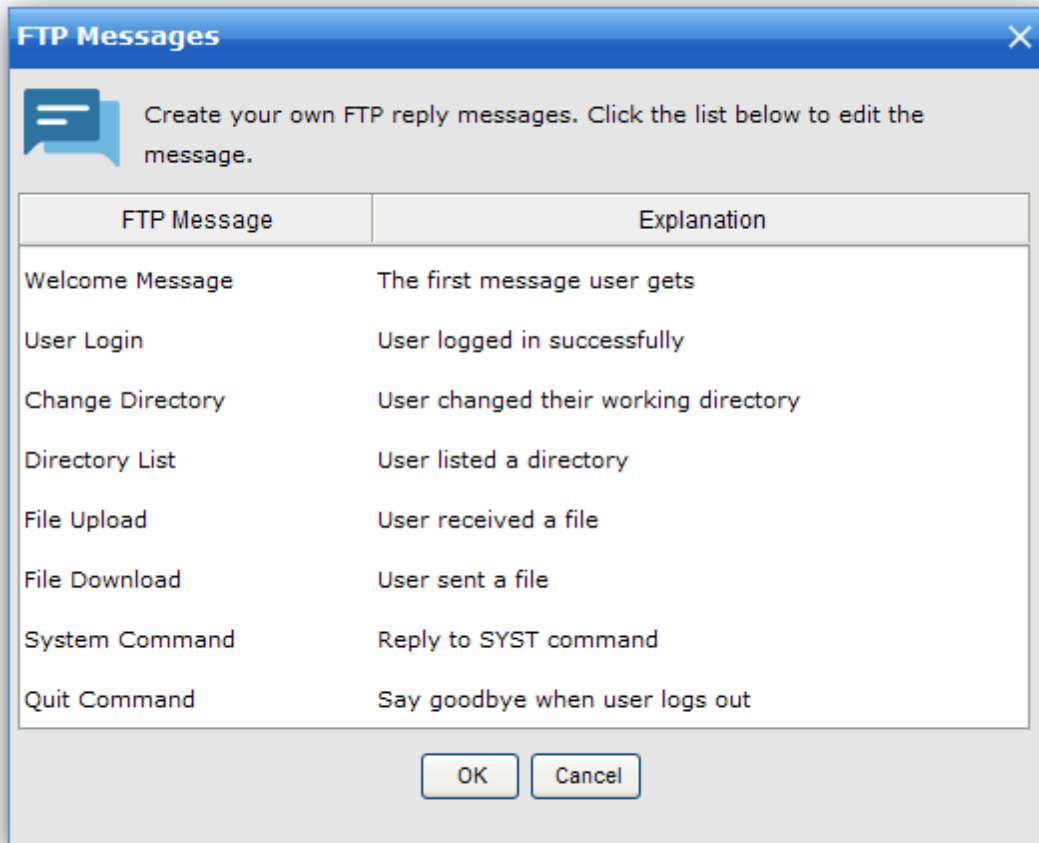


**Current :** MBytes transferred currently.  
**Max :** maximum MBytes that can be transferred.

**Total Limit:** the limit will be never reset.  
**Limit Hourly:** reset the limitation every hour.  
**Limit Daily:** reset the limitation every day.  
**Limit Weekly:** reset the limitation every week.  
**Limit Monthly:** reset the limitation every month.

## FTP Messages

You can define the custom FTP reply messages here.



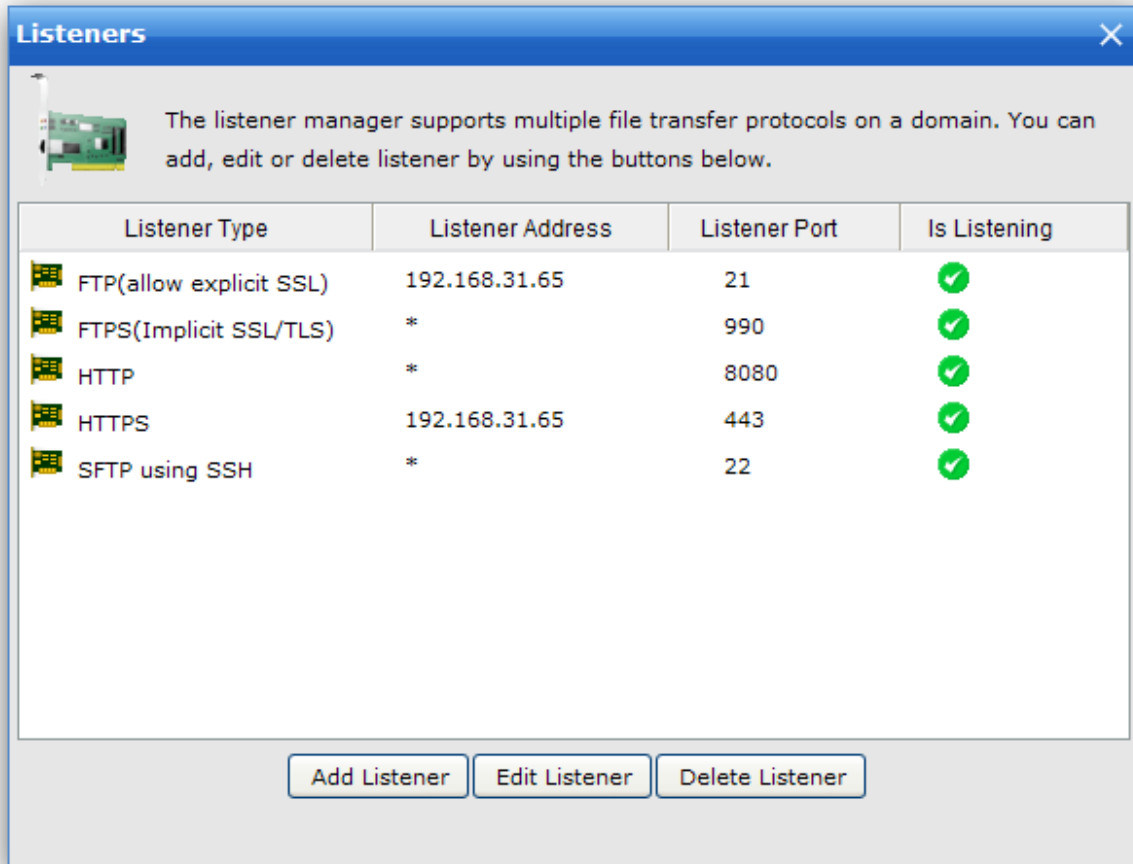
If you think the built-in messages are too cryptic for your clients or want to translate them, you can define them by yourself.

The above window lists all the FTP reply messages which can be customized, just double click on a message to edit it.

You can use server variables in the reply messages, please have a look at the chapter "Advanced Features -> Server Variables".

## Listeners

Wing FTP Server can listen on multiple ports (or multiple IP addresses) for FTP/SFTP/HTTP protocols.



### Listener Type

The supported transfer protocols: FTP, FTPS, HTTP, HTTPS and SFTP.

### IP Address

This domain listener address, default "\*" means the listener port listens on all the available IPs. If you change it into a single IP address, then the listener port will listen on that IP address only.

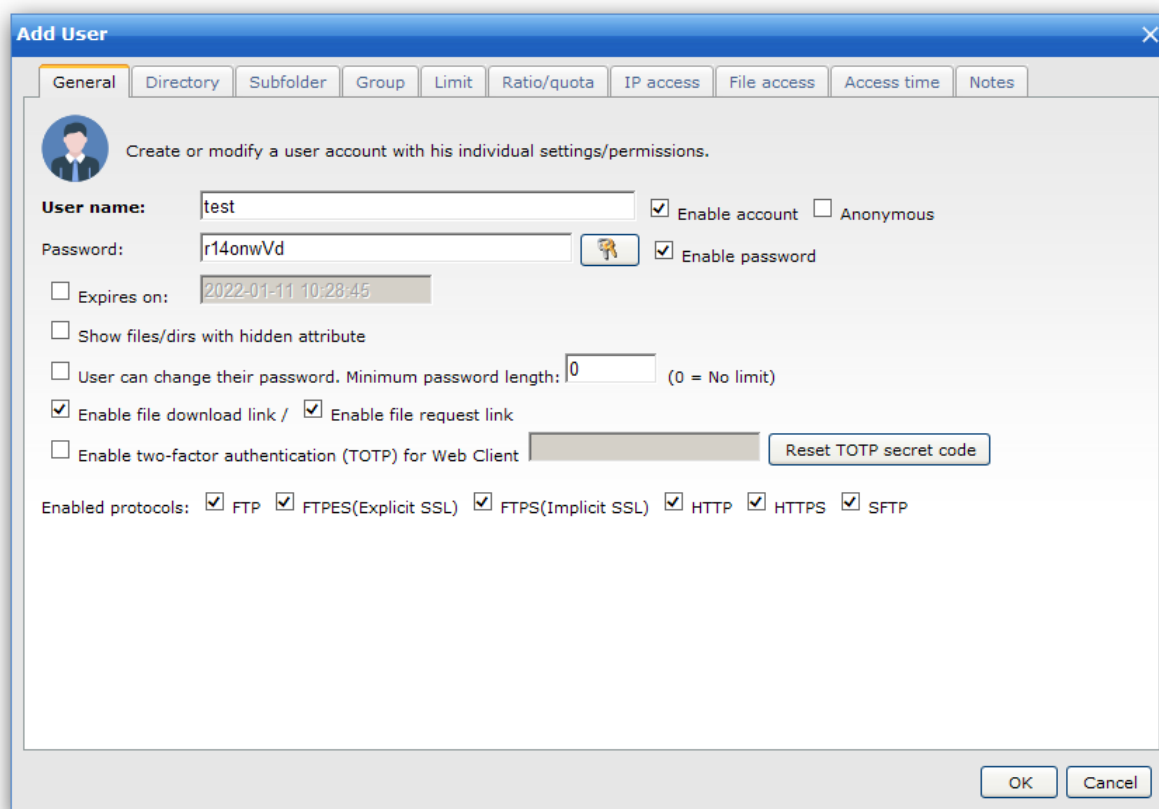
### Port

The available port is 1 - 65535.

## Users

### User > General

A user account is required in order to obtain access to the server. Click "Add User" or "Quick Add" button to add a user account.



### User Name

User name must be unique, and can not contain any of the following special characters: \ / < > | : ? \*.

### Enable account

Uncheck this option to disable the user account, disabled accounts remain on the server but cannot log in.

### Anonymous

When this option is checked, no password authentication needed.

### Password

The user password should be complex to prevent brute force attacks.

### Expire on

This account would be automatically disabled at the specified time.

### Show files/dirs with hidden attribute

With this option, files/folders with hidden attributes will be shown in the file listing.

### User can change his password

When this option is enabled, Web client can change the password in the main interface, and FTP client can send a command "SITE PSWD oldpass newpass" for changing password.

### Enable file download/request link

Enable this option if you want to generate file download/request link for this user.

**Enable two-factor authentication (TOTP) for Web Client**

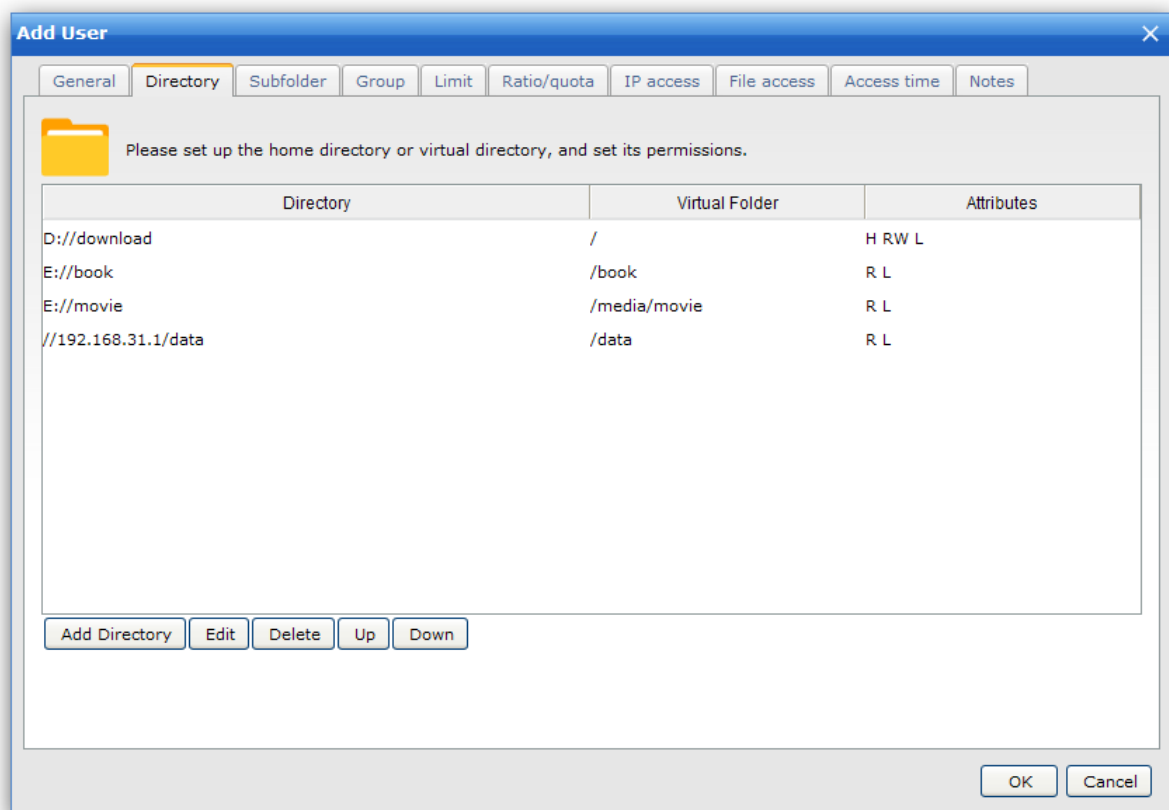
With this option, when you try to login the web client for the first time, you will be asked to install a TOTP app and scan a QR code, then you should enter a dynamic TOTP code which is generated by TOTP app. And the dynamic TOTP code will be required on any subsequent login, if client's 2FA device/data is lost, you can reset TOTP secret code by clicking the button "Reset TOTP secret code".

**Enabled protocols**

Select the transfer protocols you want to enable for this user account.

**User > Directory**

Setup a home directory or virtual directories with proper folder permissions, then client side can download files from these directories (or upload files into these directories).

**Add Directory**

It can be a home directory or virtual directories. If added as a virtual directory, the physical directory will be shown to the user as a virtual directory. For example, "E:/book" will be shown as subdirectory "/book" and "E:/movie" will be shown as subdirectory "/media/movie" (suppose you have a subfolder "/media" under the home directory).

**Edit**

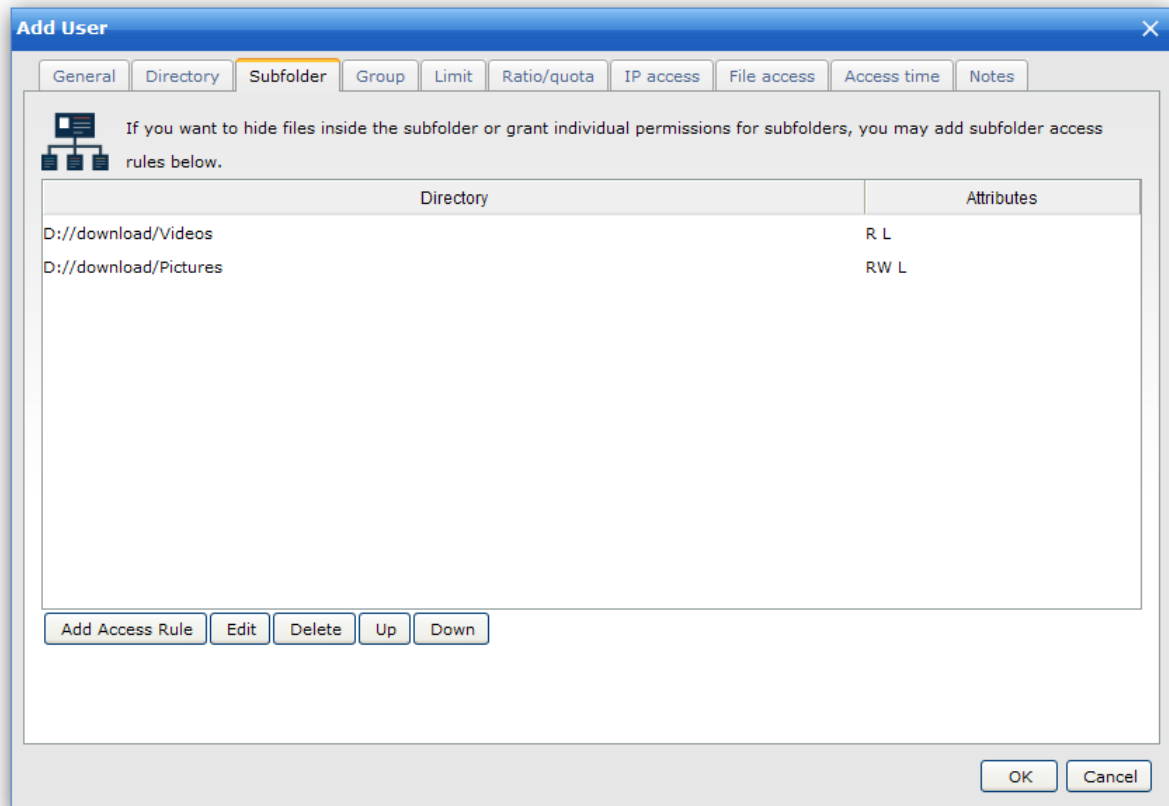
Edit the selected directory.

**Delete**

Delete the selected directory.

## User > Subfolder

If you want to hide some subfolders under the home directory or grant individual permissions for subfolders, you may add some subfolder access rules.



### Add Access Rule

Add an access rule for a subfolder, and then the subfolder will have individual permissions from the home directory.

### Edit

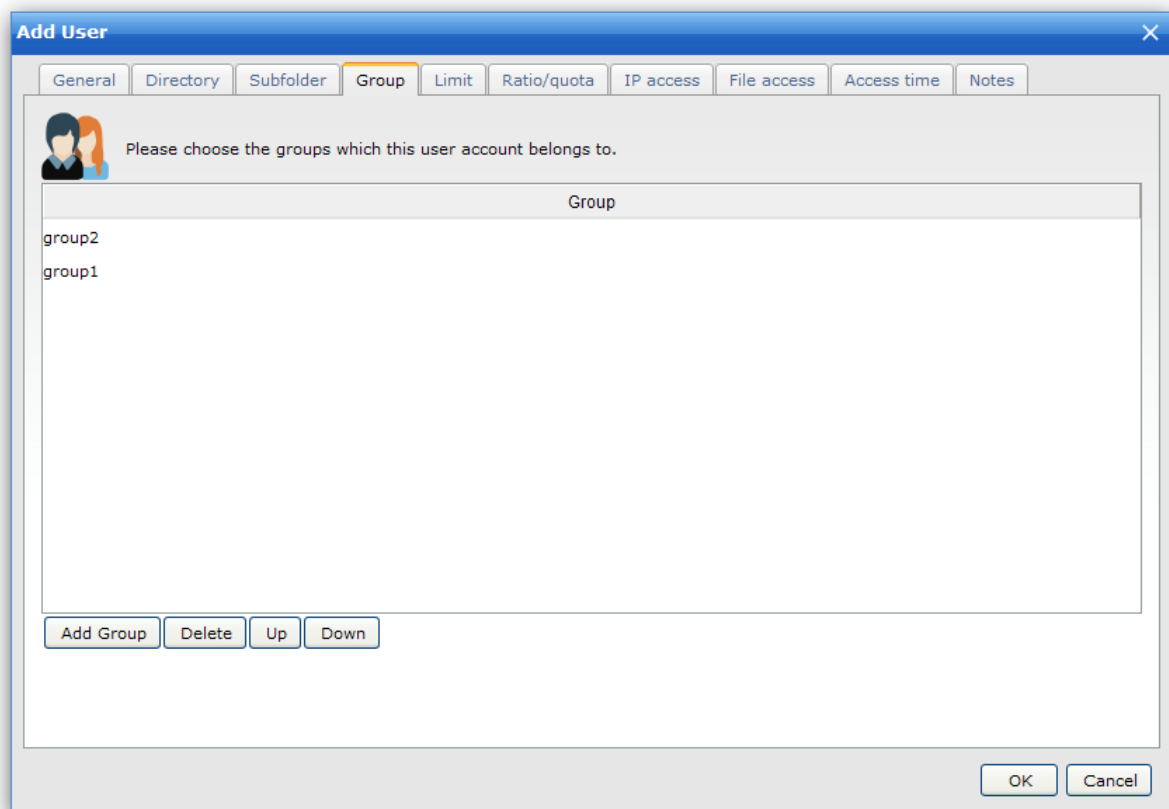
Edit the selected directory.

### Delete

Delete the selected directory.

## User > Group

If you want several users to have the same access rights to your server, it is a good idea to set up a group account for all of them rather than having to define the same options in each account. This is a handy method to handle large numbers of users/user rights without too much work. Permissions and attributes inherited by a user through group membership can be overridden at the user level. User can be a member of multiple groups in order to acquire various collections of permissions, such as File or IP access rules.

**Add Group**

Assign the user a group.

**Delete**

Remove the selected group.

**Up/Down**

Change the order of selected groups. Because a user can be a member of multiple groups, the order in which group memberships are presented is critical.



## User > Limit

You can configure Connection Limitation and Data Transfer Limitation here.

**Add User**

General | Directory | Subfolder | Group | **Limit** | Ratio/quota | IP access | File access | Access time | Notes

**Connection Limit(0 = No limit)**

Maximum number of sessions per user account:

Maximum sessions per IP address for user account:

Automatic idle connection timeout:  minutes

**Data Transfer Limit(0 = No limit)**

Maximum download speed per session:  KB/s

Maximum upload speed per session:  KB/s

Maximum download speed for user accounts:  KB/s

Maximum upload speed for user accounts:  KB/s

Automatic transfer connection timeout:  minutes

**Enable SSH Public Key Authentication**

SSH public key path:

SSH authentication method:

Only use public key authentication.

Use both public key and password authentication.

### Max sessions per user account

Specifies the maximum number of concurrent sessions that may be opened from a single user account.

### Max sessions per IP

Specifies the maximum number of concurrent sessions that a user may open from a single IP address.

### Automatic idle connection timeout (for FTP/SFTP protocol)

When a client has been idle for a specific time, client will be disconnected automatically.

### Max download speed per session

It limits the maximum download bandwidth for each individual session. Setting a limit of 0 KB/s means unlimited bandwidth.

### Max upload speed per session

It limits the maximum upload bandwidth for each individual session. Setting a limit of 0 KB/s means unlimited bandwidth.

### Max download speed for user accounts

Limits the maximum download bandwidth shared between all sessions associated with an individual user account. Setting a limit of 0 KB/s means unlimited bandwidth.

### Max upload speed for user accounts

Limits the maximum upload bandwidth shared between all sessions associated with an individual User account. Setting a limit of 0 KB/s means unlimited bandwidth.

### Automatic transfer connection timeout (for FTP/SFTP protocol)

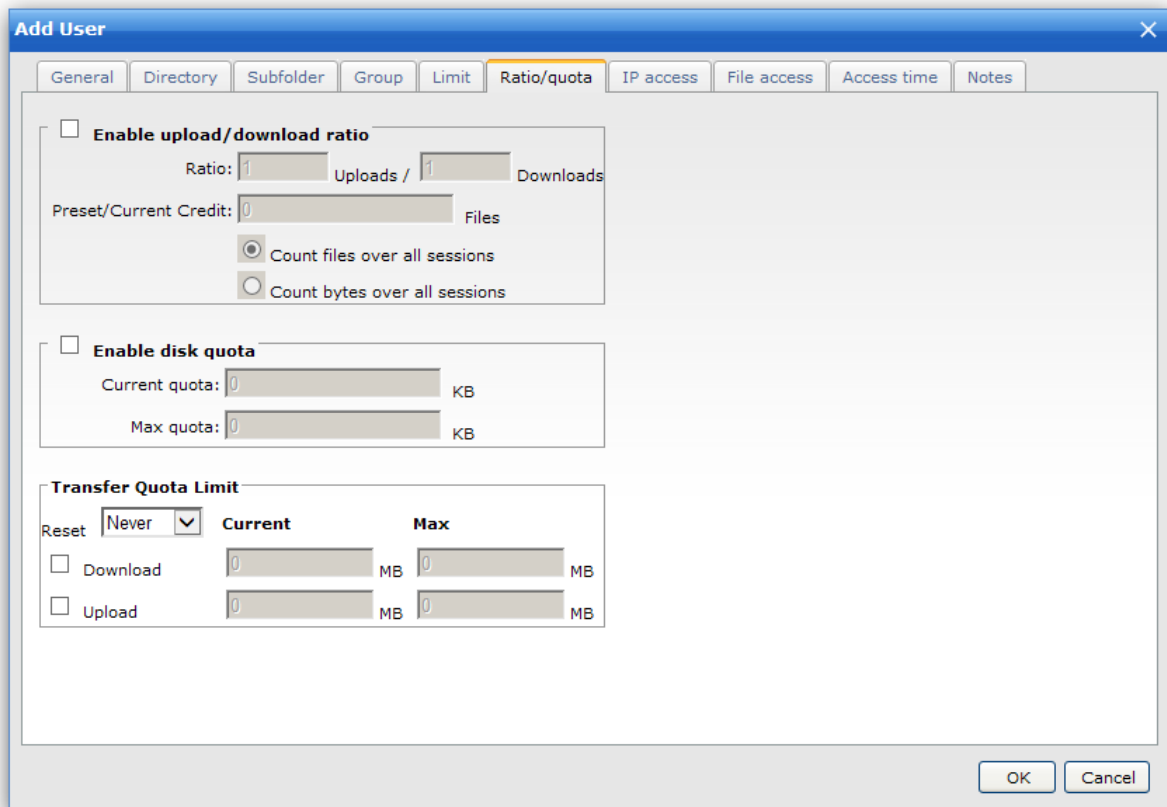
When no data transfer for a specific time, client will be disconnected automatically.

### Enable SSH Public Key Authentication

WingFTP uses the password for the SFTP authentication by default. If you want more secure, you can use public-key authentication.

### User > Ratio & Quota

Ratios are implemented to encourage people to give while they receive. For example, if you want a user to get 1 byte of download for every byte they upload, the ratio is set to 1/1; 1 byte upload for 2 bytes to download makes a ratio of 1/2. If they upload a file of 100 bytes at a ratio set to 1/2, they will have 200 bytes for credit but only 50 bytes if the ratio is set to 2/1.



### Ratio uploads

$\text{New credit} = \text{current credit} + (\text{ratio download} / \text{Ratio upload}) * \text{file size}$

### Ratio downloads

$\text{New credit} = \text{current credit} - \text{file size}$

### Count files overall sessions

The ratios apply to files uploaded or downloaded across sessions. The ratio will be calculated using the number of files uploaded or downloaded. The Credit amount applies to all users of all sessions using this account, and it is remembered by the server between sessions.

### Count bytes overall sessions

The ratios apply to files uploaded or downloaded across sessions. The ratio will be calculated using the number of bytes uploaded or downloaded. The Credit amount applies to all users of all sessions using this account, and it is remembered by the server between sessions.

### Quota

This is feature allows you to set the maximum space each user can use on your server. A user cannot upload if he exceeds his quota. Then he needs to delete some files. If a user's quota is zero and he deletes a file, his quota remains at zero.

### Reset every (hour, day, week or month)

The limit will be reset every selected period. As a hosting company, you could allocate your user a maximum traffic per month using this option.

### Transfer Limit Current

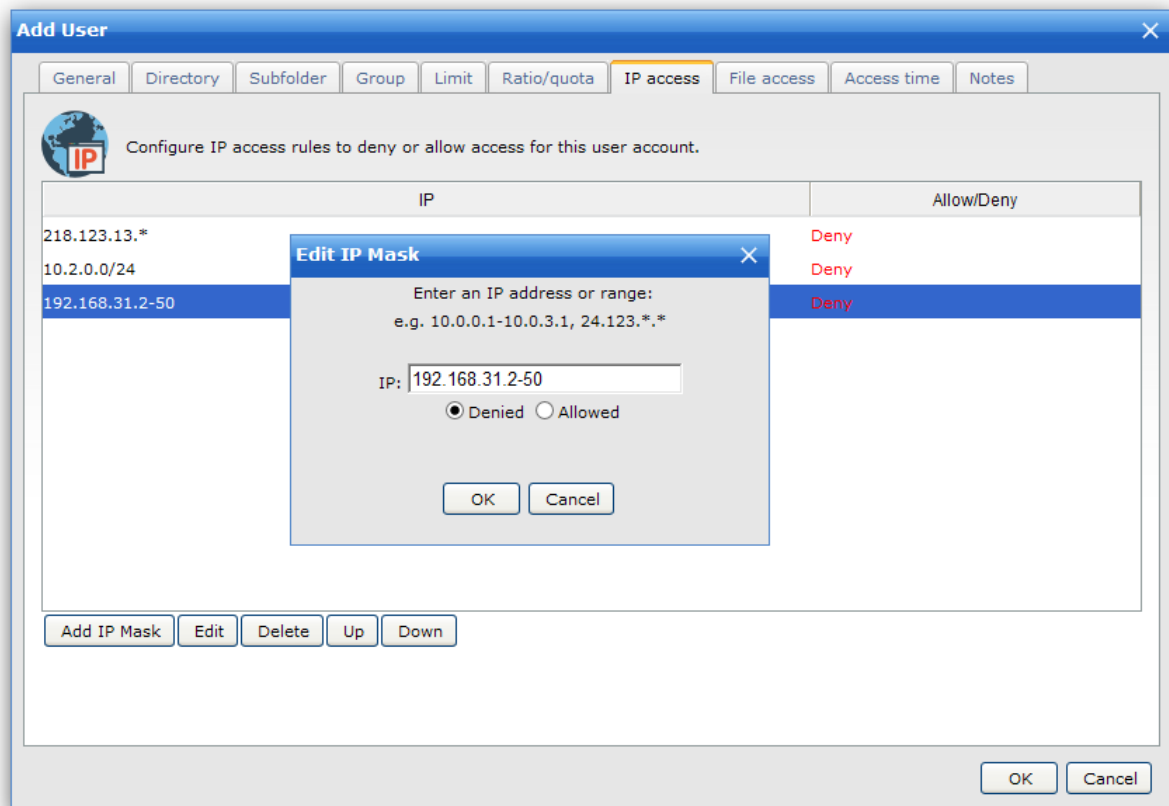
Actual MBytes transferred.

### Transfer Limit Max

Maximum MBytes can be transferred.

### User > IP Access

Configure IP access rules to allow or deny access for this user.



You can define IP access rules to allow/deny users' access based on IP address for this user.

If you do not specify any IP address, this user can log in from any IP address. But if you set allowed rules, the user can only access from the IP addresses in the allowed rules. If you set denied rules, the user can obtain access by all IP addresses except those in the denied rules.

For example:

**Allow 127.0.0.1**

Refuse connection from any IP except 127.0.0.1.

## Supported wildcards

IP address ranges and wildcards are supported by Wing FTP Server, as below:

`xxx.xxx.xxx.xxx`

IP address must be exactly matched(e.g., 192.168.1.1).

`xxx.xxx.xxx.xxx-yyy`

A specified range of IP addresses, e.g. 192.168.1.10-25.

`xxx.xxx.xxx.xxx-yyy.yyy.yyy.yyy`

A specified range of IP addresses, e.g. 192.168.1.0-192.168.5.255.

`*.*.*` or `xxx.*.*` or `xxx.xxx.*.*` or `xxx.xxx.xxx.*`

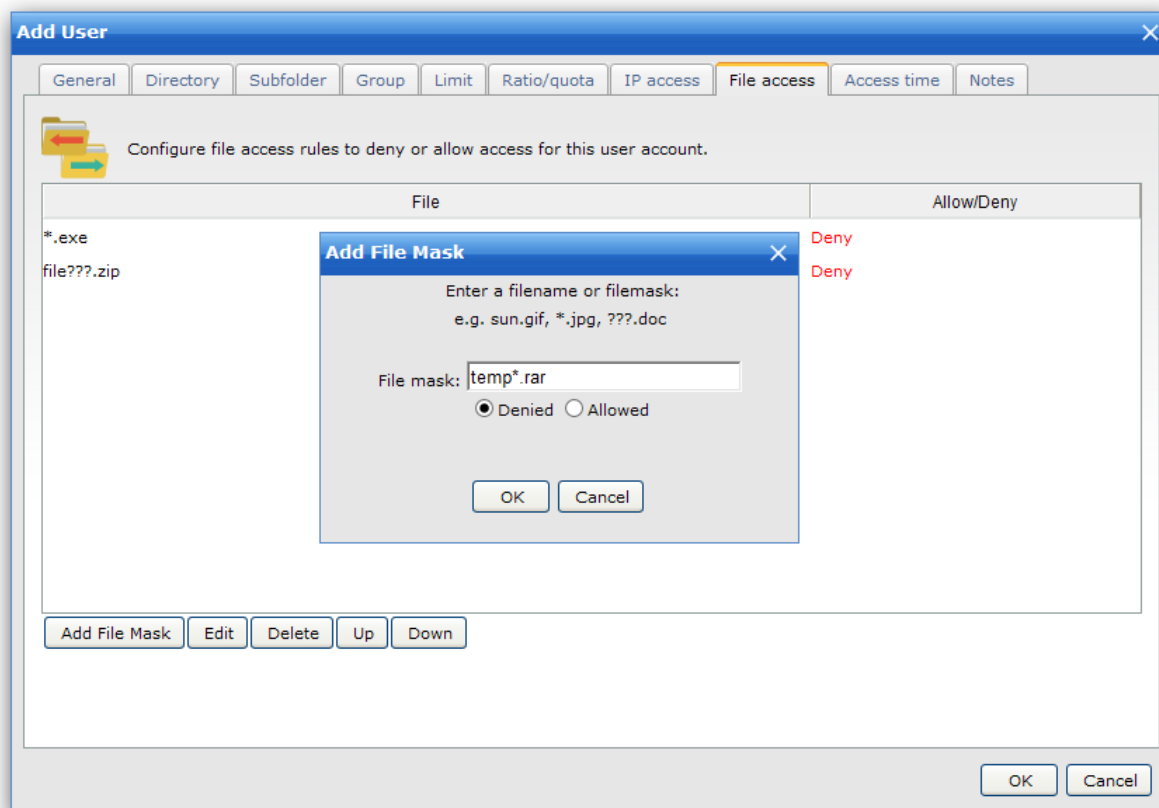
Any valid IP address value (For example, 192.168.\*.\* represents any IP between 192.168.0.0 and 192.168.255.255).

CIDR convention is also supported :

192.168.0.0/24 (represents any IP between 192.168.0.0 and 192.168.0.255)

## User > File Access

Configure File access rules to allow or deny access for this user.



Banned files are files that can't be accessed on the server. You can specify file/path mask (?, \* supported) : \*.jpg, c:\path\images\_200?\

Using this form, you can define allow/deny access based on Filename for this user account.

If you do not specify any file/path, this user can access all the files on the server. But if you set an allow list, the user can only access the files allowed in the list. If you set a deny list, the user can access all the files except those in the deny list. The order of the rules is critical too.

For example

### Allow \*.rar

Can not access/store any file except \*.rar.

### Deny \*.zip

Can access/store any file except \*.zip.

### Rule list

The rule list shows the current list and order of file access rules. Rules can be added and removed from the list using the Add and Delete buttons.

Also, the order of the rules may be altered using the Up and Down buttons on the right of the rule list.

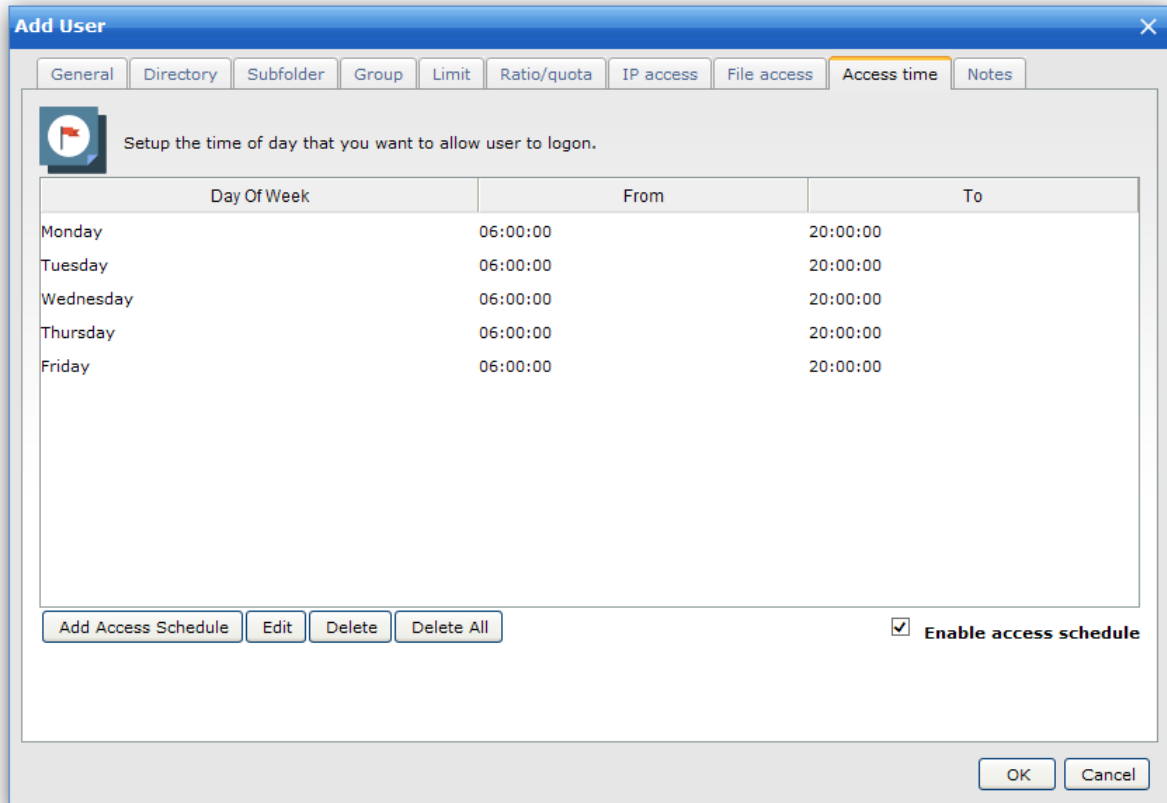
### Supported wildcards

To define access rules, you can use the wildcards ? and \*.

## User > Access Time

This option defines the access time restriction for the user account.

When you enable this option, you must add some access time rules, otherwise, this user can't login any more.



### Enable access schedule

Activate or deactivate the restriction.

### Add Access Schedule

Add "Access Time Schedule" for this user. This user can log in at your specified time.

### Edit

Edit your "Time Schedule."

### Delete

Delete the selected "Time Schedule."

### Delete All

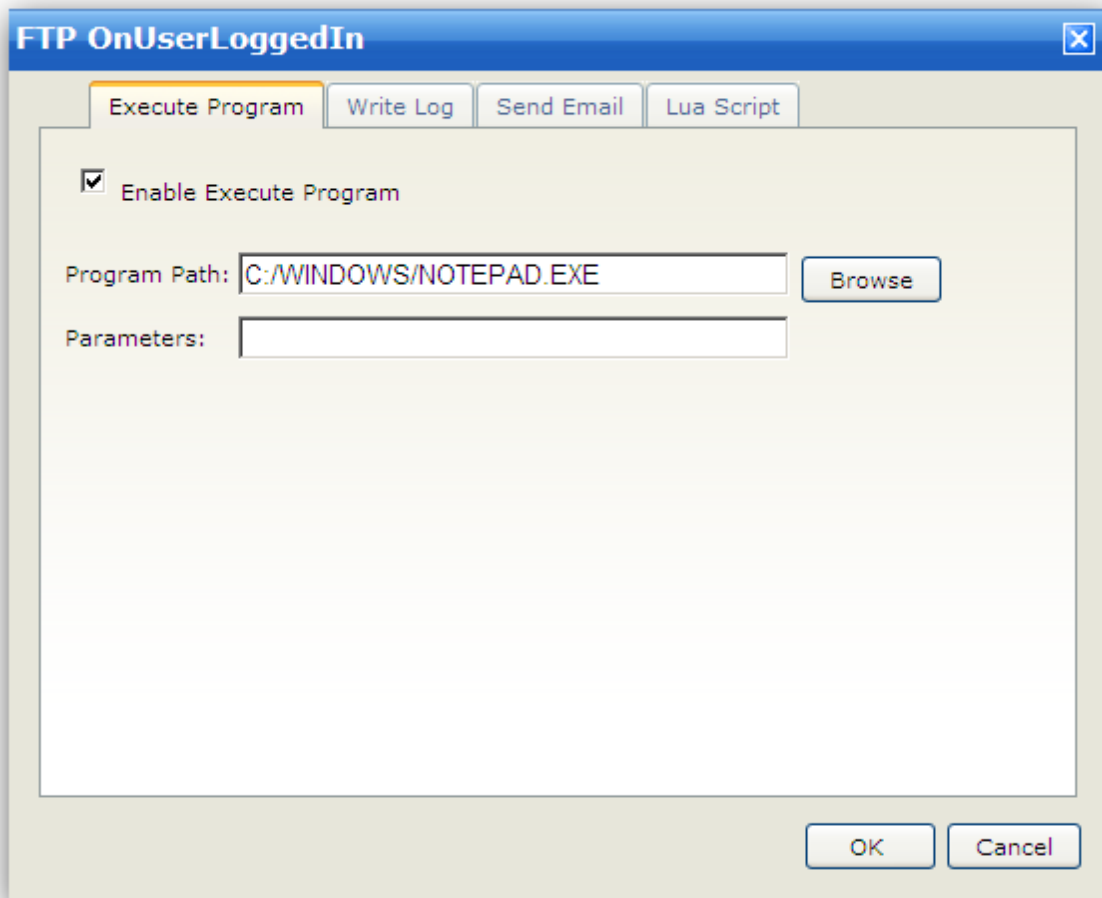
Delete all "Time Schedule." This user can never log in again unless you disable "Access Time Schedule."

## Event Manager

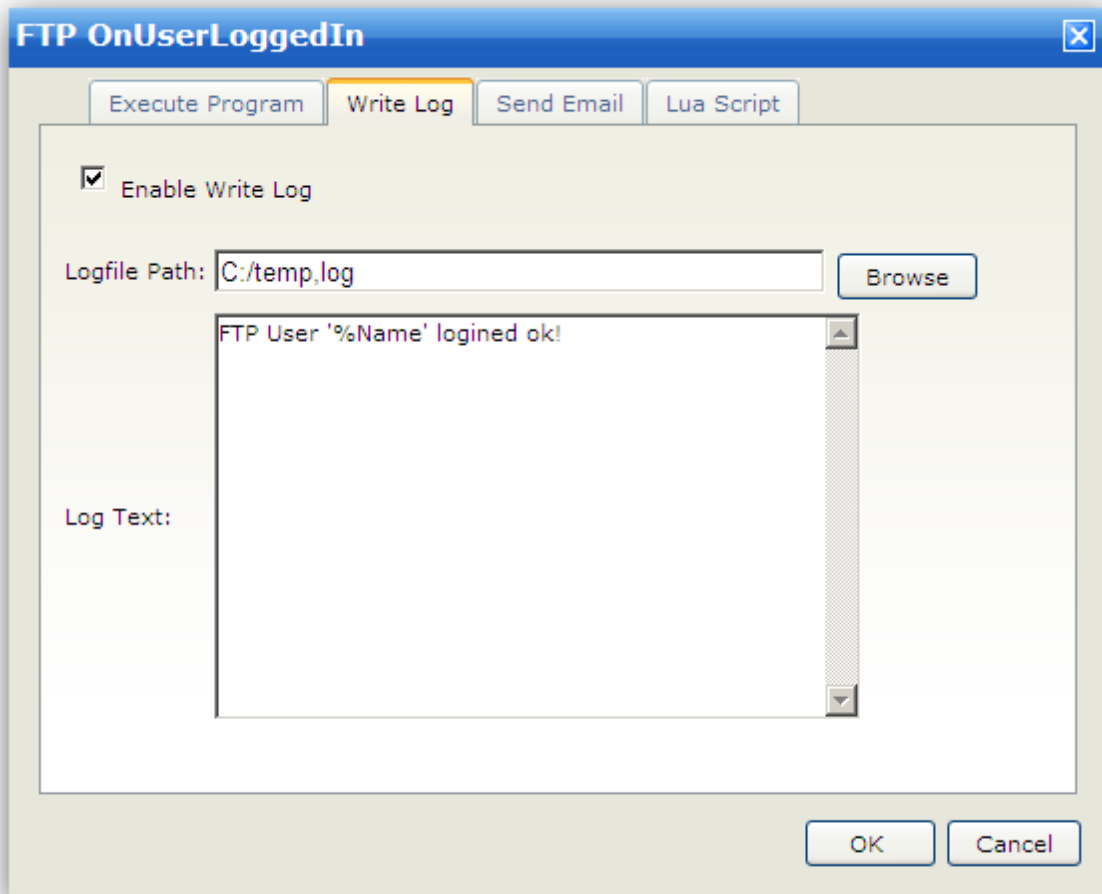
Use Event Manager to execute Lua script, send an email notification, or run 3rd party program when triggering an event. For example, when you finish uploading a file, a notification email can be sent to the administrator automatically.

## Event Actions

### 1. Execute Command

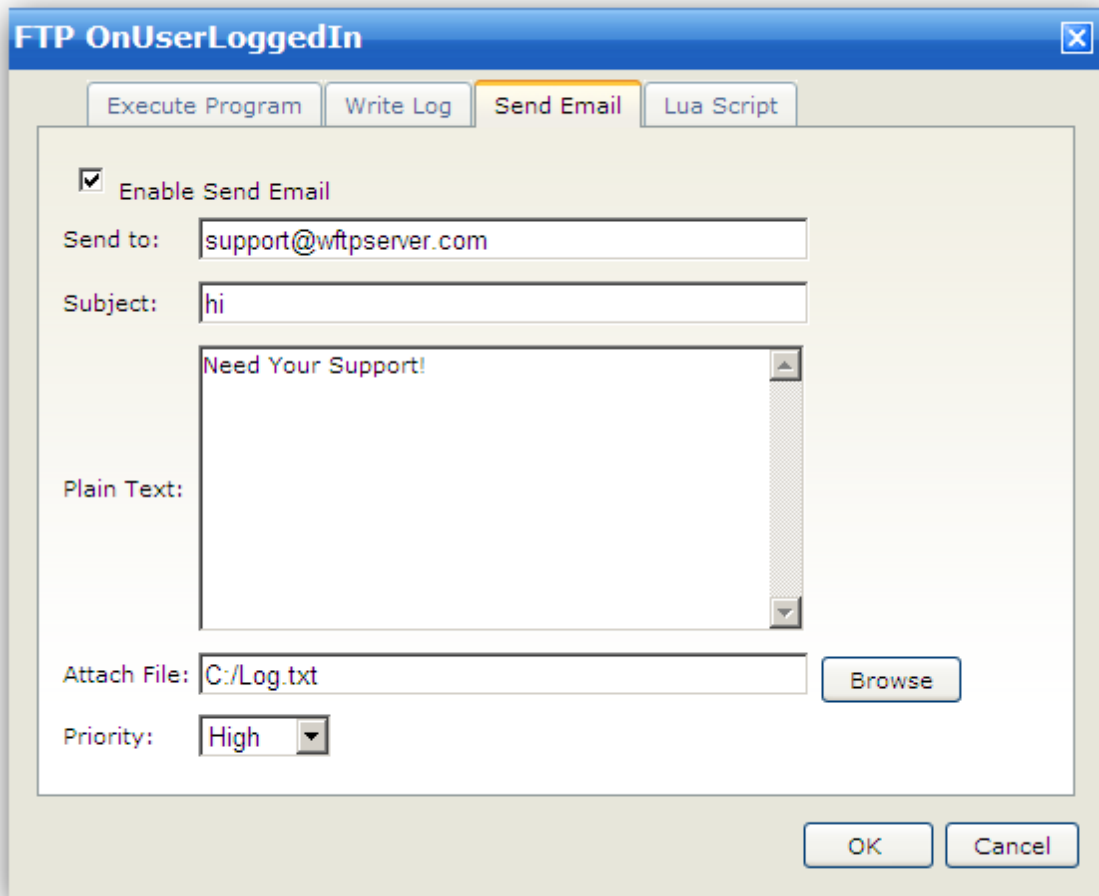


### 2. Write Log

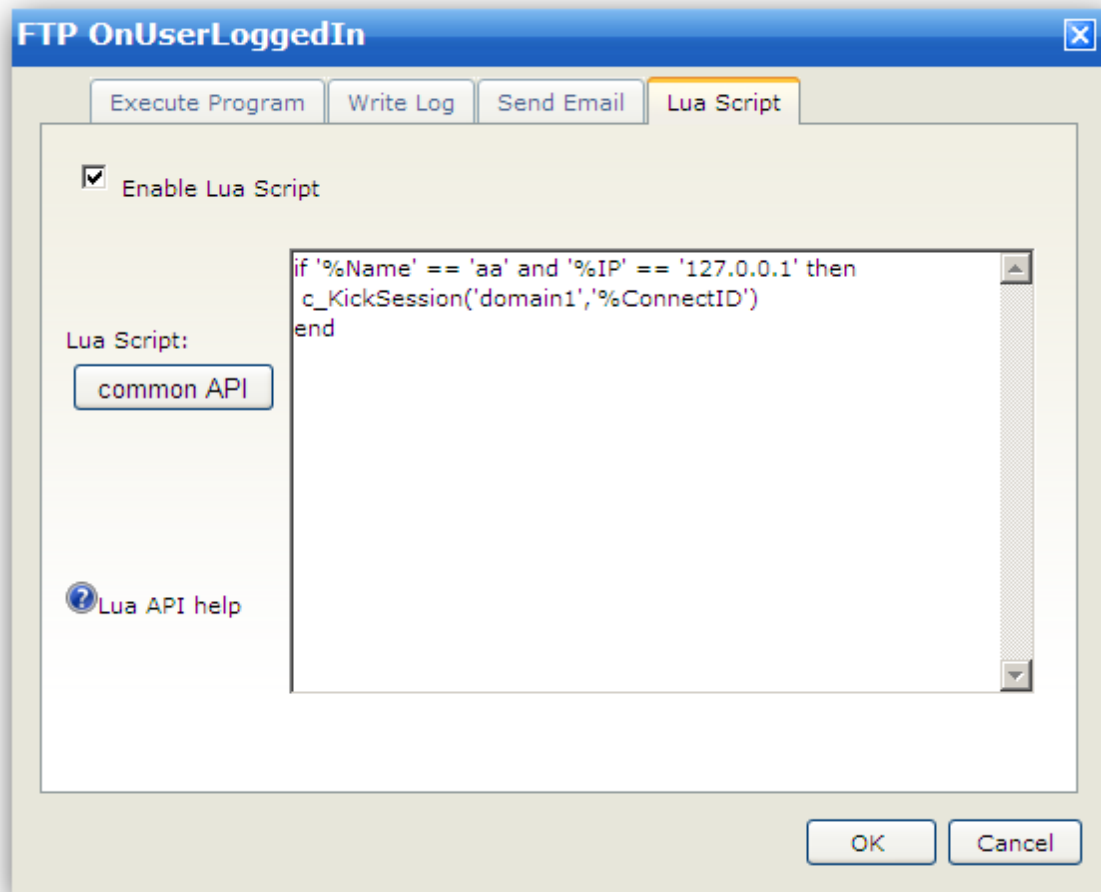


3. Send Email





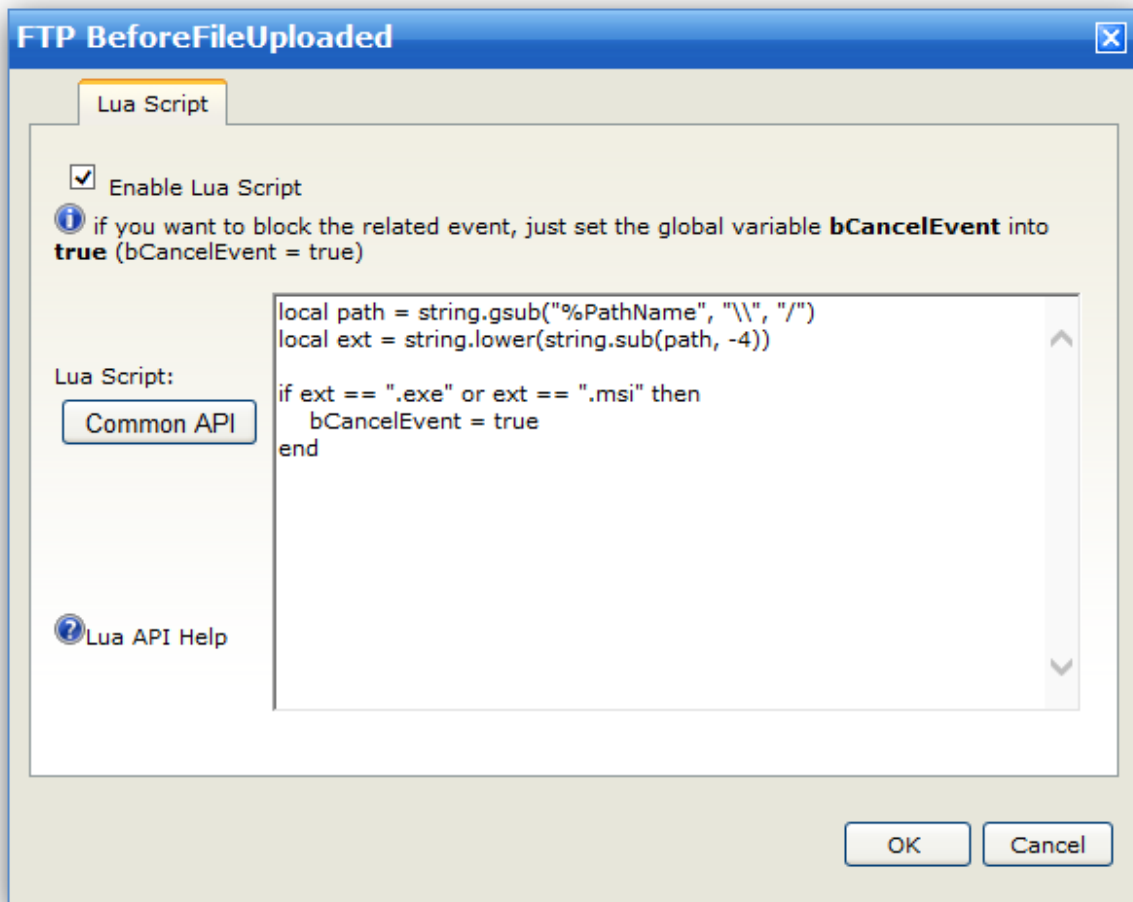
4. Lua Script



All the above actions can use many server variables, like "%Name" for the logged-in in username, "%FileName" for the uploaded file name. You can see a list of the server variables under "[Advanced Features -> Server Variables](#)".

## Special Events

Some special events will be triggered before handling related activity: BeforeUserLoggedIn, BeforeFileDownloaded, BeforeFileUploaded, BeforeFileDeleted, BeforeDirDeleted. You can use the Lua script to handle such event only. For example, event "**BeforeFileUploaded**" will be triggered before uploading a file, if you want to block the uploading activity, just set the global variable **bCancelEvent** into **true**.



The above Lua script will detect the file extension name, and if the user attempts to upload a .exe or .msi file, then the event "**BeforeFileUploaded**" will block the uploading activity.

In the event "**BeforeUserLoggedIn**", the global variable **bSelfAuthenticated** can be used for customized authentication, when you set this variable into **true**, the customized authentication is passed.

## Lua Language

### About Lua

[Lua](#) is a powerful, fast, lightweight, embeddable scripting language. Lua combines simple procedural syntax with powerful data description constructs based on associative arrays and extensible semantics. Lua is dynamically typed, runs by interpreting bytecode for a register-based virtual machine, and has automatic memory management with incremental garbage collection, making it ideal for configuration, scripting, and rapid prototyping.

Wing FTP Server includes support for the Lua scripting language. Lua scripts can be used in several cases, such as task scheduler, event manager, and administration console. It will help you to complete a complex scheduler task or a useful FTP LOGIN event or just a simple console command.

You can write Lua scripts simply by using standard Lua libraries and [Wing FTP Server's Lua API](#)<sup>[108]</sup>. For example, if you want to add a user to the server, you can call `c_AddUser(...)` API to implement it, then if you want to delete the user, just call `c_DeleteUser(...)`. It is simple yet powerful.

**Simple example** (a daily scheduler task for removing inactive users in domain "default" that have not logged in within 15 days)

```
do
    local strUserlist = c_GetUserList("default")
    local userlist = Split(strUserlist, "\n")
    for _, username in pairs(userlist) do
        local user = c_GetUser("default", username)
        local logintime = user.last_logintime
        local logintime_t = c_TranslateTime(logintime)
        if (os.time() - logintime_t) >= 3600*24*15 then
            c_DeleteUser("default", username)
        end
    end
end
end
```

### RESTful web service

Wing FTP Server also provides a [RESTful](#) web service for administration, and you can call it to execute lua script in any programming language.

The RESTful web service URL may look like this: **`http://127.0.0.1:5466/admin_webservice.html?admin=demo&pass=demo123&cmd=xxxxx`**

There are three URL parameters in the above URL, the first parameter "admin" means the administrator's username, the second parameter "pass" means the administrator's password, and the third parameter "cmd" means the lua script with URL encoded.

If the web service call fails, it will return a string starting with "[ERROR RESULT]."

Here we will present an example in some programming languages. This example is very simple, just for calculating the number of all the domains' online sessions.

#### PHP example:

```

$strUrl = "http://127.0.0.1:5466/admin_webservice.html";
$strUrlParam = "?admin=demo&pass=demo123&cmd=";
$strLuaScript = <<<EOT
    local nSessionCnt = 0
    for _,domain in pairs(c_GetDomainList()) do
        nSessionCnt = nSessionCnt + c_GetSessionCount(domain)
    end
    print(nSessionCnt)
EOT;

$strResult = file_get_contents($strUrl.$strUrlParam.rawurlencode($strLuaScript));

```

**VB script/ASP example:**

```

Set xmlHttp = CreateObject("MSXML2.ServerXMLHTTP")
strUrl = "http://127.0.0.1:5466/admin_webservice.html"
strUrlParam = "?admin=demo&pass=demo123&cmd="
strLuaScript = "local nSessionCnt = 0 "_
                &"for _,domain in pairs(c_GetDomainList()) do "_
                &" nSessionCnt = nSessionCnt + c_GetSessionCount(domain) "_
                &"end "_
                &"print(nSessionCnt)"
xmlHttp.open "GET", strUrl&strUrlParam&URLEncode(strLuaScript), False
xmlHttp.send
strResult = xmlHttp.responseText

```

```

Function URLEncode(strInput)
    For i = 1 To Len(strInput)
        intAscii = Asc(Mid(strInput, i, 1))
        If ((intAscii < 58) And (intAscii > 47)) Or ((intAscii < 91) And (intAscii > 64)) Or ((intAs
            strOutput = strOutput & Chr(intAscii)
        Else
            If intAscii < 16 Then
                strOutput = strOutput & "%0" & Trim(Hex(intAscii))
            Else
                strOutput = strOutput & "%" & Trim(Hex(intAscii))
            End If
        End If
    Next
    URLEncode = strOutput
End Function

```

**JAVA example:**

```

import java.io.*;
import java.net.*;

class GetUrlContent {
    public static void main(String[] args) throws IOException {
        String strUrl = "http://127.0.0.1:5466/admin_webservice.html";
        String strUrlParam = "admin=demo&pass=demo123&cmd=";

        String strLuaScript = "local nSessionCnt = 0 "+
            "for _,domain in pairs(c_GetDomainList()) do "+
            " nSessionCnt = nSessionCnt + c_GetSessionCount(domain) "+
            "end "+
            "print(nSessionCnt)";
    }
}

```

```

        String strResult = SendGetRequest(strUrl, strUrlParam+java.net.URLEncoder.encode(strLuaScript));
    }

    public static String SendGetRequest(String url, String param) {
        String result = "";
        try {
            String urlName = url + "?" + param;
            URL U = new URL(urlName);
            URLConnection connection = U.openConnection();
            connection.connect();
            BufferedReader in = new BufferedReader(new InputStreamReader(connection.getInputStream()));
            String line;

            while ((line = in.readLine()) != null) {
                result += line;
            }
            in.close();
        } catch (Exception e) {
            result = "";
        }
        return result;
    }
}

```

## Server Variables

The server variables can be used in Administration Console, Event Manager, Task Scheduler, and even for the FTP response message and HTTP welcome message.

### Server Variables

- %CurrentTime - The current time in local time format
- %YYYY - Current year
- %MM - Current month, 01-12
- %DD - Current day, 01-31
- %HH - Current hour, 00-23
- %mm - Current minute, 00-59
- %ss - Current second, 00-59
- %ServOnlineSessions - The total online sessions under this server
- %ServMaxOnlineSessions - The highest concurrent sessions record
- %ServLast24HSessions - The total number of sessions that connected in the past 24 hours
- %ServAverageSessionLife - The average staying time under this server
- %ServMaxSessionLife - The longest staying time for a session under this server
- %ServTotalSessions - The total number of sessions that have connected since being placed online.
- %ServRealtimeDownloadSpeedKBS - The current download transfer rate in KB/s
- %ServAverageDownloadSpeedKBS - The average download rate in KB/s
- %ServDownloadBytes - The total amount of data downloaded since being placed online
- %ServDownloadFiles - The total number of files downloaded since being placed online
- %ServRealtimeUploadSpeedKBS - The current upload transfer rate in KB/s
- %ServAverageUploadSpeedKBS - The average upload rate in KB/s
- %ServUploadBytes - The total amount of data uploaded since being placed online
- %ServUploadFiles - The total number of files uploaded since being placed online

### Domain Variables

- %DomOnlineSessions - The total online sessions under this domain
- %DomMaxOnlineSessions - The highest concurrent sessions record
- %DomLast24HSessions - The total number of sessions that connected in the past 24 hours
- %DomAverageSessionLife - The average staying time under this domain
- %DomMaxSessionLife - The longest staying time for a session under this domain

%DomTotalSessions - The total number of sessions that connected since being placed online  
%DomRealtimeDownloadSpeedKBS - The current download transfer rate in KB/s  
%DomAverageDownloadSpeedKBS - The average download rate in KB/s  
%DomDownloadBytes - The total amount of data downloaded since being placed online  
%DomDownloadFiles - The total number of files downloaded since being placed online  
%DomRealtimeUploadSpeedKBS - The current upload transfer rate in KB/s  
%DomAverageUploadSpeedKBS - The average upload rate in KB/s  
%DomUploadBytes - The total amount of data uploaded since being placed online  
%DomUploadFiles - The total number of files uploaded since being placed online

## Session Variables - Applies to the current session

%Domain - The domain name  
%IP - The client IP address  
%Name - The user name of the current session  
%Dir - The current directory, like "/download"  
%LastDir - The last created or removed directory (physical path), like "c:/upload/tempfolder"  
%PathName - The last accessed file's path, like "c:/download/test.zip"  
%FileName - The last accessed file's name, like "test.zip"  
%FileSize - The last accessed file's size  
%OldFilePath - The old file path before renaming  
%NewFilePath - The new file path after renaming  
%SessionID - The session ID  
%Protocol - The protocol type of current session (FTP, FTPS, FTPES, SFTP, HTTP, HTTPS).  
%Password - The password string sent by the client, this variable can be used in the event "**BeforeUserLoggedIn**" only.  
%WebLink - the ID of Weblink or Upload link, like "d197da4875af58340c98fe196b5195f3".  
%ClientVersion - The FTP client name  
%UserMaxQuota - The max disk quota for this user  
%UserCurrentQuota - The current disk quota for this user  
%UserName - The user real name, which is defined under "User -> Notes -> Name"  
%UserEmail - The email address, which is defined under "User -> Notes -> Email address"  
%UserNote - The email address, which is defined under "User -> Notes -> Other notes"  
%ConRealtimeDownloadSpeedKBS - The current download transfer rate in KB/s  
%ConAverageDownloadSpeedKBS - The average download rate in KB/s  
%ConDownloadFiles - The total number of files downloaded for this session  
%ConRealtimeUploadSpeedKBS - The current upload transfer rate in KB/s  
%ConAverageUploadSpeedKBS - The average upload rate in KB/s  
%ConUploadFiles - The total number of files uploaded for this session  
%ConTransferBytes - The total amount of data transferred last time by this session(include download and upload)  
%ConTransferSpeedKBS - The average transfer rate transferred last time by this session in KB/s

## Server Lua API

The Lua API is a set of administrator interface functions that allow you to use it in task scheduler, event manager, and administration console, and more information about Lua may be found here: <http://www.lua.org/>.

- [User & Group Functions](#) 

- [Domain Functions](#) <sup>[116]</sup>
- [System Functions](#) <sup>[130]</sup>
- [Get/Set Option Functions](#) <sup>[140]</sup>
- [Miscellaneous Functions](#) <sup>[143]</sup>
- [Administrator Functions](#) <sup>[147]</sup>

## User & Group

**c\_AddUser**(string strDomain,string strUsername,string strPassword,int nProtocolType,int bEnablePassword,int bEnableAccount,int nMaxDownloadSpeed,int nMaxUploadSpeed,int nMaxConnection,int nConnectTimeout,int nIdleTimeout,int nConnectPerIp,int nPassLength,int bShowHiddenFile,int bChangePass,int bSendMessage,int nRatioCredit, int nRatioDownload, int nRatioUpload, int nRatioCountMethod,int bEnableRatio,int nCurrentQuota,int nMaxQuota,int bEnableQuota, string strNoteName, string strNoteAddress,string strNoteZip,string strNotePhone,string strNoteFax,string strNoteEmail, string strNoteMemo, table tabUserDirectory, table tabUserIpMasks, table tabUserFileMasks, table tabUserUserGroups,int bEnableSchedule, table tabUserSchedules, int nLimitResetType, int bLimitEnableUpload, int nCurUploadSize,int nMaxUploadSize, int bLimitEnableDownload, int nCurDownloadSize, int nMaxDownloadSize,int bEnableExpire, string strExpireTime, int nMaxDownloadSpeedPerUser, int nMaxUploadSpeedPerUser,string strSSHPubKey, table tabSubFolderPerm, int bEnableSSHPubKey, int nSSHAuthMethod, int bEnableWeblink, int bEnableUplink, int bEnable2FA, string str2FACode, string strExtraInfo)

### Parameters

- [1]string the domain name
- [2]string the user name
- [3]string password with MD5 encryption
- [4]int enabled protocols mask, FTP=1, FTPES(explicit SSL)=2, FTPS(implicit SSL)=4, HTTP=8, HTTP (SSL)=16, SSH=32, if only FTP and HTTP are allowed, the mask number will be 1+8=9.
- [5]int enable password, 1=yes, 0=no
- [6]int enable account, 1=yes, 0=no
- [7]int max download speed for one session
- [8]int max upload speed for one session
- [9]int max number of connections
- [10]int ftp connection timeout value
- [11]int ftp idle timeout value
- [12]int max number of connections per Ip
- [13]int max password length
- [14]int show the hidden file?, 1=show,0=hide
- [15]int user can change password?, 1=allow,0=deny
- [16]int send chat message?, 1=allow,0=deny
- [17]int credit ratio
- [18]int download ratio
- [19]int upload ratio
- [20]int ratio count method, 0=count for files,1=count for bytes
- [21]int enable ratio, 1=yes, 0=no
- [22]int current quota size
- [23]int max quota size
- [24]int enable quota, 1=yes,0=no



[25]string note name  
 [26]string note address  
 [27]string note zipcode  
 [28]string note phone number  
 [29]string note fax number  
 [30]string note email  
 [31]string note memo  
 [32]table a table list of user directories (user directory also is a table, its structure is formatted as {m\_strDir,m\_strAlias,m\_bIsHomeDir,m\_bFileRead,m\_bFileWrite,m\_bFileAppend,m\_bFileDelete,m\_bDirectoryList,m\_bDirectoryMake,m\_bDirectoryDelete,m\_bDirectoryRename,m\_bFileRename,m\_bZipFile,m\_bUnzipFile}).  
 [33]table a table list of user ipmasks (user ipmask also is a table, its structure is formatted as {m\_strIp,m\_bRefuse}).  
 [34]table a table list of user filemasks list (user filemask also is a table, its structure is formatted as {m\_strMask,m\_bRefuse}).  
 [35]table a table list of user groups (user group also is a table, its structure is formatted as {m\_strGroupname}).  
 [36]int enable schedule, 1=yes,0=no  
 [37]table a table list of access schedulers (access scheduler also is a table, its structure is formatted as {m\_nWeekday,m\_strTimefrom, m\_strTimeto}).  
 [38]int transfer limit reset type, 0=reset never,1=reset hourly,2=reset daily,3=reset weekly,4=reset monthly  
 [39]int enable upload limit, 1=yes, 0=no  
 [40]int current upload size  
 [41]int max upload size  
 [42]int enable download limit, 1=yes, 0=no  
 [43]int current download size  
 [44]int max download size  
 [45]int enable account expire, 1=yes, 0=no  
 [46]string account expire time string, e.g. "2010-04-01 13:30:01"  
 [47]int max download speed for user  
 [48]int max upload speed for user  
 [49]string ssh public key path  
 [50]table subfolder access rules, its structure looks like parameter 32.  
 [51]int enable ssh public key authentication, 1=yes,0=no  
 [52]int ssh authentication method, 1=Use both public key and password authentication,0=Only use public key authentication  
 [53]int enable web download link, 1=yes,0=no  
 [54]int enable web request (upload) link, 1=yes,0=no  
 [55]int enable two-factor authentication (TOTP) for Web Client, 1=yes,0=no  
 [56]string TOTP secret code  
 [57]string extra information for the user account

**Return Values**

nil

**Remarks**

Add or modify a user account.

---

**c\_UserExist**(string strDomain, string strUsername)

**Parameters**

[1]string the domain name  
 [2]string the user name

**Return Values**

[1]bool return true if the specified user exists in the specified domain, otherwise return false

**Remarks**

Check whether the specified user exists.

---

**c\_GetUser**(string strDomain,string strUsername)

**Parameters**

[1]string the domain name  
[2]string the user name

**Return Values**

[1]table a user data table formatted as {username,password,max\_download,max\_upload,max\_download\_account,max\_upload\_account,max\_connection,connect\_timeout,idle\_timeout,connect\_per\_ip,pass\_length,show\_hidden\_file,change\_pass,send\_message,ratio\_credit,ratio\_download,ratio\_upload,ratio\_count\_method,enable\_ratio,current\_quota,max\_quota,enable\_quota,note\_name,note\_address,note\_zip,note\_phone,note\_fax,note\_email,note\_memo,{Directories table},{Ipmarks table},{Filemasks table},{Usergroups table},enable\_group,enable\_schedule,{Scheduler table},limit\_reset\_time,limit\_reset\_type,limit\_enable\_upload,cur\_upload\_size,max\_upload\_size,limit\_enable\_download,cur\_download\_size,max\_download\_size,enable\_expire,expiretime,total\_received,total\_sent,login\_count,file\_download,file\_upload,failed\_download,failed\_upload,last\_loginip,last\_logintime,protocol\_type,enable\_password,enable\_account,ssh\_pubkey\_path,{Subfolder Rules table}}.

**Remarks**

Return a table of user data if the specified user exists, otherwise return nil.

---

**c\_DeleteUser**(string strDomain, string strUsername)

**Parameters**

[1]string the domain name  
[2]string the user name

**Return Values**

nil

**Remarks**

Delete a user.

---

**c\_CopyUser**(string strDomain, string strUsername, string strNewUsername)

**Parameters**

[1]string the domain name  
[2]string the source user name  
[3]string the destination user name

**Return Values**

[1]int the result code, 1=success, -1=source user not exists, -2=destination user already exists.

**Remarks**

Copy a user.

---

**c\_AddUserDirectory**(string strDomain, string strUsername, string strDirectory, string strAlias, bool blsHomeDir, bool bFileRead, bool bFileWrite, bool bFileAppend, bool bFileDelete, bool bDirectoryList, bool bDirectoryMake, bool bDirectoryDelete, bool bDirectoryRename, bool bFileRename, bool bZipFile, bool bUnzipFile)

### Parameters

[1]string the domain name  
[2]string the user name  
[3]string the directory path  
[4]string alias of the directory, set "/" for home directory.  
[5]bool is home directory?  
[6]bool can user read files?  
[7]bool can user write files?  
[8]bool can user resume files uploading?  
[9]bool can user delete files?  
[10]bool can user see files listing?  
[11]bool can user make a directory?  
[12]bool can user remove a directory?  
[13]bool can user rename a directory?  
[14]bool can user rename a file?  
[15]bool can user zip file or folder?  
[16]bool can user unzip a zip file?

### Return Values

nil

### Remarks

Add a home directory or virtual directory for user account.

---

**c\_ResetUserStatistic**(string strDomain, string strUsername)

### Parameters

[1]string the domain name  
[2]string the user name

### Return Values

nil

### Remarks

Reset user's statistics.

---

**c\_GetUserList**(string strDomain)

### Parameters

[1]string the domain name

### Return Values

[1]string username list string

### Remarks

Returns user list, username in the list is separated with a carriage return sign.

**c\_GetUserListPage**(string strDomain, int nPageNum)

**Parameters**

[1]string the domain name  
[2]int the page number

**Return Values**

[1]string username list string

**Remarks**

This function is similar to c\_GetUserList(), the difference is that it will be used when storing user data via database, and the 2nd parameter tells records offset (20 records/page, 0=1~20, 1=21~40...).

**c\_GetUserPageCount**(string strDomain)

**Parameters**

[1]string the domain name

**Return Values**

[1]int the total records of users  
[2]int the total pages (20 records per page)

**Remarks**

Returns the total records of users in database, and the total pages (20 records per page).

**c\_AddGroup**(string strDomain, string strGroupname, int nMaxDownloadSpeed, int nMaxUploadSpeed, int nMaxConnection, int nConnectTimeout, int nIdleTimeout, int nConnectPerIp, int nConnectPerUser, int nConnectPerIpForUser, int bShowHiddenFile, int bSendMessage, table tabUserDirectory, table tabUserIpMasks, table tabUserFileMasks, int nMaxDownloadSpeedPerUser, int nMaxUploadSpeedPerUser, int nMaxDownloadSpeedPerGroup, int nMaxUploadSpeedPerGroup, table tabUserList, table tabSubfolderPerm)

**Parameters**

[1]string the domain name  
[2]string the group name  
[3]int max download speed for one session  
[4]int max upload speed for one session  
[5]int max number of connections  
[6]int ftp connect timeout value  
[7]int ftp idle timeout value  
[8]int max number of connections per IP for group  
[9]int max number of connections per user account  
[10]int max number of connections per IP for user account  
[11]int whether to show the hidden file, 1=show, 0=hide  
[12]int whether to send message, 1=allow, 0=deny  
[13]table a table list of group directories (group directory also is a table, its structure is formatted as {m\_strDir, m\_strAlias, m\_bIsHomeDir, m\_bFileRead, m\_bFileWrite, m\_bFileAppend, m\_bFileDelete, m\_bDirectoryList, m\_bDirectoryMake, m\_bDirectoryDelete, m\_bDirectoryRename, m\_bFileRename, m\_bZipFile, m\_bUnzipFile}).  
[14]table a table list of group ipmasks (group ipmask also is a table, its structure is formatted as {m\_strIp, m\_bRefuse}).

- [15]table a table list of group filemasks (group filemask also is a table, its structure is formatted as {m\_strMask,m\_bRefuse}).
- [16]int max download speed for specified user
- [17]int max upload speed for specified user
- [18]int max download speed for this group
- [19]int max upload speed for this group
- [20]table userlist belong to this group, like { {"aa"}, {"bb"}, {"cc"} }
- [21]table subfolder access rules, its structure looks like parameter 13

## Return Values

nil

## Remarks

Add or modify a user group.

---

**c\_GroupExist**(string strDomain, string strGroupname)

## Parameters

- [1]string the domain name
- [2]string the group name

## Return Values

- [1]bool return true if the specified group exists in the specified domain, otherwise return false

## Remarks

Check whether the specified group exists.

---

**c\_GetGroup**(string strDomain, string strGroupname)

## Parameters

- [1]string the domain name
- [2]string the group name

## Return Values

- [1]table a table of group, its structure is formatted as {groupname,max\_download,max\_upload,max\_download\_account,max\_upload\_account,max\_connection,connect\_timeout, idle\_timeout, connect\_per\_ip, max\_session\_per\_user, max\_ipsession\_for\_user, show\_hidden\_file,send\_message, {directory table},{lpmask table},{Filemask table}, total\_received, total\_sent, login\_count, file\_download, file\_upload, failed\_download, failed\_upload,{Userlist table} }.

## Remarks

Return a table of group data if the specified group exists, otherwise return nil.

---

**c\_DeleteGroup**(string strDomain, string strGroupname)

## Parameters

- [1]string the domain name
- [2]string the group name

## Return Values

nil

## Remarks

Delete a group.

---

**c\_CopyGroup**(string strDomain, string strGroupname, string strNewGroupname)

## Parameters

[1]string the domain name  
[2]string the source group name  
[3]string the destination group name

## Return Values

[1]int the result code, 1=success, -1=source group not exists, -2=destination group already exists.

## Remarks

Copy a group.

---

**c\_AddGroupDirectory**(string strDomain, string strGroupname, string strDirectory, string strAlias, bool bIsHomeDir, bool bFileRead, bool bFileWrite, bool bFileAppend, bool bFileDelete, bool bDirectoryList, bool bDirectoryMake, bool bDirectoryDelete, bool bDirectoryRename, bool bFileRename, bool bZipFile, bool bUnzipFile)

## Parameters

[1]string the domain name  
[2]string the group name  
[3]string the directory path  
[4]string alias of the directory, set "/" for home directory.  
[5]bool is home directory?  
[6]bool can user read files?  
[7]bool can user write files?  
[8]bool can user resume files uploading?  
[9]bool can user delete files?  
[10]bool can user see files listing?  
[11]bool can user make a directory?  
[12]bool can user remove a directory?  
[13]bool can user rename a directory?  
[14]bool can user rename a file?  
[15]bool can user zip file or folder?  
[16]bool can user unzip a zip file?

## Return Values

nil

## Remarks

Add virtual directory for a group.

---

**c\_ResetGroupStatistic**(string strDomain, string strGroupname)

## Parameters

[1]string the domain name  
[2]string the group name

## Return Values

nil

## Remarks

Reset group's statistics.

---

**c\_GetGroupList**(string strDomain)

## Parameters

[1]string the domain name

## Return Values

[1]string groupname list string

## Remarks

Returns the group list, and group name in the list is separated with a carriage return sign.

---

**c\_GetGroupListPage**(string strDomain, int nPageNum)

## Parameters

[1]string the domain name

[2]int the page number

## Return Values

[1]string groupname list string

## Remarks

This function is similar to c\_GetGroupList(), the difference is that it will be used when storing group data via database, and the 2nd parameter tells records offset (20 records/page, 0=1~20, 1=21~40...).

---

**c\_GetGroupPageCount**(string strDomain)

## Parameters

[1]string the domain name

## Return Values

[1]int the total records of groups

[2]int the total pages (20 records per page)

## Remarks

Returns the total records of groups in database, and the total pages (20 records per page).

Domain Function

**c\_AccessDataWay**(string strDomain)

## Parameters

[1]string the domain name

## Return Values

[1]int the method for database storage of accounts information, 1=xml file, 2=mysql, 3=ODBC

## Remarks

Returns method of database storage, 1=xml file, 2=mysql, 3=ODBC

---

**c\_ResetUserData**(string strDomain)

**Parameters**

[1]string the domain name

**Return Values**

nil

**Remarks**

Reset all the user/group data in the specified domain.

---

**c\_ReconnectDB**(string strDomain)

**Parameters**

[1]string the domain name

**Return Values**

nil

**Remarks**

Reconnects the database.

---

**c\_TestMysql**(string strDomain)

**Parameters**

[1]string the domain name

**Return Values**

[1]bool return true if mysql connection is successful, otherwise return false

**Remarks**

Test mysql connection.

---

**c\_TestODBC**(string strDomain)

**Parameters**

[1]string the domain name

**Return Values**

[1]bool return true if ODBC connection is successful, otherwise return false

**Remarks**

Test ODBC connection.

---

**c\_KickSession**(string strDomain,int nId,int nKickWay,int nBlockMin,int nAddToUser,int nAddToGroup,  
int nAddToDomain,int nDisableAccount,string strSendMsg)

**Parameters**



[1]string the domain name  
[2]int the session ID  
[3]int the kick way,0=disconnect,1=disconnect and ban IP for minutes, 2=disconnect and block IP permanently  
[4]int the minutes number for banning IP.  
[5]int whether to add this IP to user's IP-block list, 1=yes, 0=no.  
[6]int whether to add this IP to group's IP-block list, 1=yes, 0=no.  
[7]int whether to add this IP to domain's IP-block list, 1=yes, 0=no.  
[8]int whether to disable user account, 1=yes, 0=no.  
[9]string the message string will be sent to ftp client.

### Return Values

[1]int the result code,1=success, 0=failure

### Remarks

Kill the specified session with full options.

---

**c\_KickSessionByName**(string strDomain, string strName)

### Parameters

[1]string the domain name  
[2]string the user name

### Return Values

[1]int the result code,1=success, 0=failure

### Remarks

Kill the online sessions of same names as parameter "strName".

---

**c\_KickAllSessions**(string strDomain)

### Parameters

[1]string the domain name

### Return Values

[1]int the result code,1=success, 0=failure

### Remarks

Kill all the online sessions.

---

**c\_GetConnectionsList**(string strDomain)

### Parameters

[1]string the domain name

### Return Values

[1]table an online session(formatted as {id, username, protocol, ip, lastcommand,directory}) list table.

### Remarks

Return a table list of online sessions.

---

**c\_GetDomainLog**(string strDomain)

### Parameters

[1]string the domain name

### Return Values

[1]string domain log text

### Remarks

Get the log text of a specified domain.

---

**c\_SetFTPEvent**(string strDomain,int nEventType,int bExecute,string strExeFilePath,string strParameter,int bLogFile,string strLogFilePath,string strLogText,int bSendMail,string strMailTo,string strSubject,string strPlainText,string strAttachFile,int nPriority,int bLuaScript,string strLuaScriptText)

### Parameters

[1]string the domain name  
[2]int the ftp event type, you can check them at the bottom.  
[3]int whether to execute a program, 1=yes, 0=no.  
[4]string the physical path of third-party program.  
[5]string the parameters of third-party program.  
[6]int whether to write something to a log file, 1=yes, 0=no.  
[7]string the path of log file.  
[8]string the log text.  
[9]int whether to send an email, 1=yes, 0=no.  
[10]string the receiver's email address.  
[11]string the email's subject.  
[12]string the email's plain text.  
[13]string the email's attachment file path.  
[14]int the priority for sending email, 1=normal, 2=high, 3=low  
[15]int whether to execute lua script, 1=yes, 0=no.  
[16]string the lua script content.

### Return Values

nil

### Remarks

This function is used for configuring the ftp event's settings.

---

**c\_GetFTPEvent**(string strDomain,int nEventType)

### Parameters

[1]string the domain name  
[2]int the ftp event type, you can check them at the bottom.

### Return Values

[1]int whether to execute a program, 1=yes, 0=no.  
[2]string the physical path of third-party program.  
[3]string the parameters of third-party program.  
[4]int whether to write something to a log file, 1=yes, 0=no.  
[5]string the path of log file.  
[6]string the log text.  
[7]int whether to send an email, 1=yes, 0=no.

[8]string the receiver's email address.  
[9]string the email's subject.  
[10]string the email's plain text.  
[11]string the email's attachment file path.  
[12]int the priority for sending email, 1=normal, 2=high, 3=low  
[13]int whether to execute lua script, 1=yes, 0=no.  
[14]string the lua script content.

## Remarks

Get specified ftp event's settings.

---

**c\_SetSSEvent**(string strDomain,int nEventType,int bExecute,string strExeFilePath,string strParameter,int bLogFile,string strLogFilePath,string strLogText,int bSendMail,string strMailTo,string strSubject,string strPlainText,string strAttachFile,int nPriority,int bLuaScript,string strLuaScriptText)

## Parameters

same as c\_SetFTPEvent()

## Return Values

nil

## Remarks

This function is used for configuring the ssh event's settings.

---

**c\_GetSSEvent**(string strDomain, int nEventType)

## Parameters

[1]string the domain name  
[2]int the ssh event type, you can check them at the bottom.

## Return Values

same as c\_SetFTPEvent()

## Remarks

Get specified ssh event's settings.

---

**c\_SetHTTPEvent**(string strDomain,int nEventType,int bExecute,string strExeFilePath,string strParameter,int bLogFile,string strLogFilePath,string strLogText,int bSendMail,string strMailTo,string strSubject,string strPlainText,string strAttachFile,int nPriority,int bLuaScript,string strLuaScriptText)

## Parameters

same as c\_SetFTPEvent()

## Return Values

nil

## Remarks

This function is used for configuring http event's settings.

---

**c\_GetHTTPEvent**(string strDomain,int nEventType)

## Parameters

[1]string the domain name  
[2]int the http event type, you can check them at the bottom.

## Return Values

same as c\_SetFTPEvent()

## Remarks

Get specified http event's settings.

---

## [c\\_GetListenerList](#)(string strDomain)

### Parameters

[1]string the domain name

### Return Values

[1]table a domain listener(formatted as {ID,Type,Ip\_Address,Port,Gateway\_Host,Gateway\_Port,Is\_Listening}) list table.

### Remarks

Return a table list of domain listeners.

---

## [c\\_AddListener](#)(string strDomain,int nListenerType,string strListenerIp,int nListenerPort, strGatewayHost,nGatewayPort)

### Parameters

[1]string the domain name  
[2]int listener type,1=FTP, 2=FTPS, 3=HTTP, 4=HTTPS, 5=SFTP.  
[3]string listener's IP address, \* for all local addresses.  
[4]int listener port  
[5]string gateway host  
[6]int gateway port

### Return Values

[1]int the result code, 0=success, 1=invalid IP, 2=invalid port, 3=invalid index, 4=listener exists, 99=unknown error.

### Remarks

Add a domain listener.

---

## [c\\_DeleteListener](#)(string strDomain,int nIndex)

### Parameters

[1]string the domain name  
[2]int listener index

### Return Values

[1]int the result code, 0=success, 1=invalid IP, 2=invalid port, 3=invalid index, 4=listener exists, 99=unknown error.

### Remarks

Delete a domain listener.

---

**c\_UpdateListener**(string strDomain,int nIndex,int nListenerType,string strListenerIp,int nListenerPort, strGatewayHost,nGatewayPort)

**Parameters**

[1]string the domain name  
[2]int listener index  
[3]int listener type, 1=FTP, 2=FTPS, 3=HTTP, 4=HTTPS, 5=SFTP.  
[4]string listener's IP address, \* for all local addresses.  
[5]int listener port  
[6]string gateway host  
[7]int gateway port

**Return Values**

[1]int the result code, 0=success, 1=invalid IP, 2=invalid port, 3=invalid index, 4=listener exists, 99=unknown error.

**Remarks**

Modify a domain listener by specified index.

---

**c\_GetSfvDirList**(string strDomain)

**Parameters**

[1]string the domain name

**Return Values**

[1]table the SFV directory list table.

**Remarks**

Return SFV directory list.

---

**c\_GetIPMaskList**(string strDomain)

**Parameters**

[1]string the domain name

**Return Values**

[1]table the domain IPmask(formatted as {ip,refuse}) list table.

**Remarks**

Return domain's IPmask list.

---

**c\_GetFileMaskList**(string strDomain)

**Parameters**

[1]string the domain name

**Return Values**

[1]table the domain Filemask(formatted as {filename,refuse}) list table.

**Remarks**

Return domain's filemask list.

---

**c\_SetSfvDirList**(string strDomain,table tabSfvDir)

**Parameters**

[1]string the domain name  
[2]table the SFV directory list table

**Return Values**

nil

**Remarks**

Set SFV directory list.

---

**c\_SetIPMaskList**(string strDomain,table tabIpMask)

**Parameters**

[1]string the domain name  
[2]table the IPmask list table (formatted as {ip,refuse})

**Return Values**

nil

**Remarks**

Set domain's IPmask list.

---

**c\_SetFileMaskList**(string strDomain,table tabFileMask)

**Parameters**

[1]string the domain name  
[2]table the filemask list table (formatted as {filename,refuse})

**Return Values**

nil

**Remarks**

Set domain's filemask list.

---

**c\_GetTransferLimit**(string strDomain)

**Parameters**

[1]string the domain name

**Return Values**

[1]table the transfer limit table formatted as {enable\_upload\_never,cur\_upload\_size\_never,max\_upload\_size\_never,enable\_download\_never,cur\_download\_size\_never,max\_download\_size\_never,enable\_upload\_hourly,cur\_upload\_size\_hourly,max\_upload\_size\_hourly,enable\_download\_hourly,cur\_download\_size\_hourly,max\_download\_size\_hourly,enable\_upload\_daily,cur\_upload\_size\_daily,max\_upload\_size\_daily,enable\_download\_daily,cur\_download\_size\_daily,max\_download\_size\_daily,enable\_upload\_weekly,cur\_upload\_size\_weekly,max\_upload\_size\_weekly,enable\_download\_weekly,cur\_download\_size\_weekly,

max\_download\_size\_weekly,enable\_upload\_monthly,  
cur\_upload\_size\_monthly,max\_upload\_size\_monthly,enable\_download\_monthly,  
cur\_download\_size\_monthly,max\_download\_size\_monthly}.

## Remarks

Return a table of transfer limit settings.

---

**c\_SetTransferLimit**(string strDomain,table tabTransferLimit)

## Parameters

[1]string the domain name  
[2]table the transfer limit table (like the return value format of c\_GetTransferLimit() ).

## Return Values

nil

## Remarks

Set domain's transfer limit.

---

**c\_StartDomain**(string strDomain)

## Parameters

[1]string the domain name

## Return Values

nil

## Remarks

Put the domain online.

---

**c\_StopDomain**(string strDomain,int nStopMode)

## Parameters

[1]string the domain name  
[2]int the options, 0=disconnect all clients, 1=wait for clients to disconnect, 2=wait for clients to terminate their transfer.

## Return Values

nil

## Remarks

Put the domain offline.

---

**c\_IsDomainOnline**(string strDomain)

## Parameters

[1]string the domain name

## Return Values

[1]bool return true if the domain is online, otherwise return false

**Remarks**

Check whether the domain is online.

---

**c\_GetSessionCount**(string strDomain)**Parameters**

[1]string the domain name

**Return Values**

[1]int the current session number

**Remarks**

Return the current session number in the specified domain.

---

**c\_GetSessionLiveTime**(string strDomain,int nSessionID)**Parameters**

[1]string the domain name

[2]int the client session ID

**Return Values**

[1]int the live time of this session

**Remarks**

Return session's living time.

---

**c\_GetSessionStatistic**(string strDomain,int nSessionID)**Parameters**

[1]string the domain name

[2]int the client session ID

**Return Values**

[1]table session statistics table formatted as {down\_speed,down\_averagespeed,down\_bytes,down\_files,up\_speed,up\_averagespeed,up\_bytes,up\_files}.

**Remarks**

Get session statistics.

---

**c\_GetDomainStatistic**(string strDomain)**Parameters**

[1]string the domain name

**Return Values**

[1]table domain statistics table formatted as {running\_time,current\_session,max\_session,last24hour\_session,session\_averagesec,session\_maxsec,total\_session,down\_speed,down\_averagespeed,down\_bytes,down\_files,up\_speed,up\_averagespeed,up\_bytes,up\_files}.

**Remarks**

Get domain statistics.



---

**c\_SendSiteMsg**(string strDomain,int nSessionID,string strMessage)

**Parameters**

[1]string the domain name  
[2]int the client session ID  
[3]string the message text

**Return Values**

nil

**Remarks**

Send a message to the client(with FTP/FTPS protocol).

---

**c\_BroadcastSiteMsg**(string strDomain,string strMessage)

**Parameters**

[1]string the domain name  
[2]string the message text

**Return Values**

nil

**Remarks**

Broadcast a message to all the clients(with FTP/FTPS protocol).

---

**c\_GetSiteMsg**(string strDomain,int nSessionID)

**Parameters**

[1]string the domain name  
[2]int the client session ID

**Return Values**

[1]string the message text

**Remarks**

Return all the chatting message of the specified client.

---

**c\_CreateDomainLogo**(string strDomain,string strLogoPath)

**Parameters**

[1]string the domain name  
[2]string the image path of your domain logo.

**Return Values**

nil

**Remarks**

Create a customized logo for the specified domain.

---

**c\_GetTempBlockIpList**(string strDomain)

**Parameters**

[1]string the domain name

**Return Values**

[1]table temporary banned IP(formatted as {ip,sec}) list table

**Remarks**

Get the temporary banned IP list.

---

**c\_AddTempBlockIp**(string strDomain,string strBanIp,int nBanSec)

**Parameters**

[1]string the domain name  
[2]string the banned IP address  
[3]int value of banned seconds

**Return Values**

[1]bool return true if added the temporary banned IP successfully, otherwise return false

**Remarks**

Add a temporary banned IP.

---

**c\_DeTempBlockIp**(string strDomain,string strBanIp)

**Parameters**

[1]string the domain name  
[2]string the banned IP address

**Return Values**

[1]bool return true if removed the temporary banned IP successfully, otherwise return false

**Remarks**

Remove a temporary banned IP.

---

**c\_GetUserAbsolutePath**(string strDomain,string strUsername,string strNowDir)

**Parameters**

[1]string the domain name  
[2]string the user name  
[3]string the current user directory

**Return Values**

[1]string return the physical path of current user directory

**Remarks**

Get the physical path of current user directory.

---

**c\_GetChartData**(string strDomain,int nTimeType,int nTrafficType)

## Parameters

[1]string the domain name  
 [2]int time period type 0: last 5minutes, 1: last 5hours, 2: last 10days  
 [3]int traffic type 0: download speed, 1: upload speed, 2: session number

## Return Values

[1]string return the input data for graphs chart of real-time server traffics.

## Remarks

Get the input data for graphs chart of real-time server traffics.

**c\_SetDomainPasvOption**(string strDomain,int nPasvModeWay,string strPasvFixIp,string strPasvWebIp,string strPasvDnsIp,int nPasvRefIntval,int nEnableUPnP,int nMinPasvPort,int nMaxPasvPort)

## Parameters

[1]string the domain name  
 [2]int get server's IP way in ftp pasv mode transfer  
 [3]string your server's IP on internet  
 [4]string the specified web url that will resolve your internet IP.  
 [5]string the specified dynamic DNS, will resolve the hostname as my.dnsdns.org.  
 [6]int the refreshed interval value for getting server's IP.  
 [7]int enable auto-forwarding pasv ports to router, 1=yes, 0=no.  
 [8]int the pasv ports range from.  
 [9]int the pasv ports range to.

## Return Values

nil

## Remarks

This function is used to change domain's FTP PASV options.

## //FTP Event Type(used in c\_SetFTPEvent, c\_GetFTPEvent)

```
FTP_USER_LOGIN_EVENT = 0           //When user has successfully logged in
FTP_USER_DISCONNECT_EVENT = 1      //When user leaves the server
FTP_USER_CONNECT_TIMEOUT_EVENT = 2 //When user has been disconnected for no command
sending
FTP_DIR_CREATE_EVENT = 3           //When a directory has been created
FTP_DIR_DELETE_EVENT = 4           //When a directory has been deleted
FTP_EXCEED_USERPASS_EVENT = 5     //When a user failed to authenticate(exceed limit USER/
PASS)
FTP_FILE_BANNED_EVENT = 6          //When users attempt to upload a banned file
FTP_FILE_DELETE_EVENT = 7          //When a file has just been deleted
FTP_FILE_DOWNLOAD_EVENT = 8        //When a file has just been downloaded
FTP_FILE_RENAME_EVENT = 9          //When a file has been renamed
FTP_FILE_UPLOAD_EVENT = 10         //When a file was just uploaded
FTP_QUOTA_EXCEED_EVENT = 11        //When quota has reached maximum quota allowed
FTP_TOOMANY_CONNECTION_PERIP_EVENT = 12 //When too many connections have
logged in with same IP
FTP_TOOMANY_CONNECTION_PERACCOU_EVENT = 13 //When too many connections have
logged in with same account
```

```

FTP_IP_BANNED_EVENT = 14 //When a banned IP try to login on this server
FTP_SITE_COMMAND_MESSAGE = 15 //When user use site command to send a message to
the server
FTP_SITE_COMMAND_CHANGE_PASSWORD = 16 //When user use site command to
change password
FTP_EXCEED_LIMIT = 17 //When user exceeds transfer limit
FTP_EXCEED_MAX_SESSION_ON_GROUP = 18 //When exceed Maximum number of
sessions for group
FTP_EXCEED_MAX_SESSION_PER_IP_ON_GROUP = 19 //When exceed Maximum sessions per
IP address for group
FTP_EXCEED_MAX_SESSION_ON_DOMAIN = 20 //When exceed Maximum number of
sessions on domain
FTP_EXCEED_MAX_SESSION_PER_IP_ON_DOMAIN = 21 //When exceed Maximum sessions per
IP address on domain

```

**//SSH Event Type(used in c\_SetSSHEvent,c\_GetSSHEvent)**

```

SSH_USER_LOGIN_EVENT = 0 //When user has successfully logged in
SSH_USER_DISCONNECT_EVENT = 1 //When user leaves the server
SSH_USER_CONNECT_TIMEOUT_EVENT = 2 //When user has been disconnected for no command
sending
SSH_DIR_CREATE_EVENT = 3 //When a directory has been created
SSH_DIR_DELETE_EVENT = 4 //When a directory has been deleted
SSH_EXCEED_USERPASS_EVENT = 5 //When a user failed to authenticate(exceed limit USER/
PASS)
SSH_FILE_BANNED_EVENT = 6 //When users attempt to upload a banned file
SSH_FILE_DELETE_EVENT = 7 //When a file has just been deleted
SSH_FILE_DOWNLOAD_EVENT = 8 //When a file has just been downloaded
SSH_FILE_RENAME_EVENT = 9 //When a file has been renamed
SSH_FILE_UPLOAD_EVENT = 10 //When a file was just uploaded
SSH_QUOTA_EXCEED_EVENT = 11 //When quota has reached maximum quota allowed
SSH_TOOMANY_CONNECTION_PERIP_EVENT = 12 //When too many connections have
logged in with same IP
SSH_TOOMANY_CONNECTION_PERACCOU_EVENT = 13 //When too many connections have
logged in with same account
SSH_IP_BANNED_EVENT = 14 //When a banned IP try to login on this server
SSH_EXCEED_LIMIT = 15 //When user exceed transfer limit
SSH_EXCEED_MAX_SESSION_ON_GROUP = 16 //When exceed Maximum number of
sessions for group
SSH_EXCEED_MAX_SESSION_PER_IP_ON_GROUP = 17 //When exceed Maximum sessions per
IP address for group
SSH_EXCEED_MAX_SESSION_ON_DOMAIN = 18 //When exceed Maximum number of
sessions on domain
SSH_EXCEED_MAX_SESSION_PER_IP_ON_DOMAIN = 19 //When exceed Maximum sessions per
IP address on domain

```

**//HTTP Event Type(used in c\_SetHTTPEvent,c\_GetHTTPEvent)**

```

WEB_USER_LOGIN_EVENT = 0 //When user has successfully logged in
WEB_DIR_CREATE_EVENT = 1 //When a directory has been created
WEB_DIR_DELETE_EVENT = 2 //When a directory has been deleted
WEB_EXCEED_USERPASS_EVENT = 3 //When a user failed to authenticate(exceed limit USER/
PASS)
WEB_FILE_BANNED_EVENT = 4 //When users attempt to upload a banned file
WEB_FILE_DELETE_EVENT = 5 //When a file has just been deleted
WEB_FILE_DOWNLOAD_EVENT = 6 //When a file has just been downloaded
WEB_FILE_RENAME_EVENT = 7 //When a file has been renamed

```

```
WEB_FILE_UPLOAD_EVENT = 8           //When a file was just uploaded
WEB_QUOTA_EXCEED_EVENT = 9          //When quota has reached maximum quota allowed
WEB_TOOMANY_CONNECTION_PERIP_EVENT = 10 //When too many connections have
logged in with same IP
WEB_TOOMANY_CONNECTION_PERACCOU_EVENT = 11 //When too many connections have
logged in with same account
WEB_IP_BANNED_EVENT = 12            //When a banned IP try to login on this server
WEB_CHANGE_PASSWORD = 13           //When user changes his password
WEB_EXCEED_LIMIT = 14               //When user exceeds transfer limit
WEB_EXCEED_MAX_SESSION_ON_GROUP = 15 //When exceed Maximum number of sessions
for group
WEB_EXCEED_MAX_SESSION_PER_IP_ON_GROUP = 16 //When exceed Maximum sessions per
IP address for group
WEB_EXCEED_MAX_SESSION_ON_DOMAIN = 17 //When exceed Maximum number of sessions
on domain
WEB_EXCEED_MAX_SESSION_PER_IP_ON_DOMAIN = 18 //When exceed Maximum sessions per
IP address on domain
WEB_USER_DISCONNECT_EVENT = 19      //When user leaves the server
```

## System Function

### **c\_ExitServer()**

#### **Parameters**

nil

#### **Return Values**

nil

#### **Remarks**

Safely shut down the server process.

---

### **c\_StartServer()**

#### **Parameters**

nil

#### **Return Values**

nil

#### **Remarks**

Put all the domains online.

---

### **c\_StopServer()**

#### **Parameters**

nil

#### **Return Values**

nil

#### **Remarks**

Put all the domains offline.

---

**c\_ReplaceGlobalVar**(string strSource)

**Parameters**

[1]string the source string

**Return Values**

[1]string the string with replaced values.

**Remarks**

Replace all the global variables of the source string and return it.

---

**c\_SendMail**(string strTo,string strSubject,string strPlainText,string strAttach,string strSmtpName,bool bHTML)

**Parameters**

[1]string the receiver's email addresses, multiple email addresses can be separated by a comma

[2]string mail subject

[3]string mail content

[4]string attach file path

[5]string the smtp configuration name

[6]bool mail content is HTML?, true=yes, false=no.

**Return Values**

[1]bool return true if email is sent successfully, otherwise return false

**Remarks**

Send an email with a few options.

---

**c\_SendMailComplete**(string strTo,string strSubject,string strPlainText,string strAttach,string strSenderEmail,string strSenderName,string strSmtpServer,int nSmtpPort,string strSmtpUserName,string strSmtpPassword,bool bSmtpNeedAuth,bool bUseSsl,bool bHTML)

**Parameters**

[1]string the receiver's email addresses, multiple email addresses can be separated by a comma

[2]string mail subject

[3]string mail content

[4]string attach file path

[5]string the sender's email address

[6]string the sender's name

[7]string the smtp server's address

[8]int the smtp server's port

[9]string the smtp server's username

[10]string the smtp server's password

[11]bool need auth?, true=yes, false=no.

[12]bool need SSL?, true=yes, false=no.

[13]bool mail content is HTML?, true=yes, false=no.

**Return Values**

[1]bool return true if email is sent successfully, otherwise return false

**Remarks**

Send an email with full options.

---

**c\_GetVersion()****Parameters**

nil

**Return Values**

[1]string the installed version of Wing FTP Server, such as "3.0.0"

**Remarks**

Get the installed version of Wing FTP Server.

---

**c\_GetLastVersion()****Parameters**

nil

**Return Values**

[1]string the latest version of Wing FTP Server, such as "3.0.0"

**Remarks**

Get the latest version of Wing FTP Server.

---

**c\_GetOsType()****Parameters**

nil

**Return Values**

[1]string the current operating system type, such as "Windows"

**Remarks**

Get the current operating system type.

---

**c\_ClearThumbCache()****Parameters**

nil

**Return Values**

nil

**Remarks**

Clear all the thumbnail images cache.

---

**c\_CreateCustomLogo()**

**Parameters**

[1]string the image path of your company logo.

**Return Values**

nil

**Remarks**

Create a custom logo for all domains.

---

**[c\\_ResetDefaultLogo\(\)](#)****Parameters**

nil

**Return Values**

nil

**Remarks**

Reset to the default logo for all domains.

---

**[c\\_GetGlobalIPMaskList\(\)](#)****Parameters**

nil

**Return Values**

[1]table the domain IPmask(formatted as {ip,refuse}) list table.

**Remarks**

Return entire server's ipmask list.

---

**[c\\_GetGlobalFileMaskList\(\)](#)****Parameters**

nil

**Return Values**

[1]table the domain Filemask(formatted as {filename,refuse}) list table.

**Remarks**

Return entire server's filemask list.

---

**[c\\_SetGlobalIPMaskList](#)(table tblIpMask)****Parameters**

[1]table the IPmask list table (formatted as {ip,refuse})

**Return Values**

nil

**Remarks**



Set entire server's ipmask list.

---

**c\_SetGlobalFileMaskList**(table tabFileMask)

**Parameters**

[1]table the Filemask list table (formatted as {filename,refuse})

**Return Values**

nil

**Remarks**

Set entire server's filemask list.

---

**c\_GetSystemLog**()

**Parameters**

nil

**Return Values**

[1]string system log text

**Remarks**

Get all the system log text.

---

**c\_AddSystemLog**(string strLog,int nType)

**Parameters**

[1]string the system log text

[2]int the system log type, you can check them at the bottom.

**Return Values**

nil

**Remarks**

Add a system log.

---

**c\_GetServerStatistic**()

**Parameters**

nil

**Return Values**

[1]table server statistics table formatted as {running\_time,current\_session,max\_session,last24hour\_session,session\_averagesec,session\_maxsec,total\_session, down\_speed,down\_averagespeed,down\_bytes,down\_files,up\_speed,up\_averagespeed,up\_bytes,up\_files}.

**Remarks**

Get server statistics.

---

**c\_CreateSSLCertificate**(string strFileName,string strFilePath,int nKeySize,string strCountry,string

strState,string strLocality,string strOrganization,string strUnit,string strCertName,string strEmail,string strPassword)

## Parameters

[1]string the certificate file name  
[2]string the output directory  
[3]int key size  
[4]string 2-digit country code  
[5]string full state or province  
[6]string locality(city)  
[7]string organization  
[8]string organization unit  
[9]string common name(your name or your server's hostname)  
[10]string contact e-mail  
[11]string SSL key password

## Return Values

[1]bool return true if created SSL certificate file successfully, otherwise return false

## Remarks

This function is used for creating an SSL certificate file .

---

## [c\\_GetSSLCertList\(\)](#)

### Parameters

nil

### Return Values

[1]table a table list of all the SSL configuration (formatted as {name,state})

### Remarks

Return all the SSL certificate configurations.

---

## [c\\_GetSSLCertificate](#)(string strName)

### Parameters

[1]string the certificate configuration name

### Return Values

[1]table a table formatted as {ssl\_certpath,ssl\_keypath,ssl\_password}.

### Remarks

Get an SSL certificate configuration table on success, or nil on error.

---

## [c\\_CheckSSLCertificate](#)(string strName)

### Parameters

[1]string the certificate configuration name

### Return Values

[1]bool return true if SSL certificate is valid, otherwise return false

## Remarks

Check whether the specified SSL certificate is valid.

---

**c\_AddSSLCertificate**(string strName,string strCertPath,string strKeyPath,string strPassword)

## Parameters

[1]string the certificate configuration name  
[2]string the certificate file path  
[3]string the key file path  
[4]string the key password

## Return Values

[1]bool return true if successfully added an SSL certificate configuration, otherwise return false

## Remarks

Add or modify an SSL certificate configuration.

---

**c\_DeleteSSLCertificate**(string strName)

## Parameters

[1]string the certificate configuration name

## Return Values

[1]bool return true if successfully deleted an SSL certificate configuration, otherwise return false

## Remarks

Delete an SSL certificate configuration.

---

**c\_CreateSSHKey**(string strFileName,string strFilePath,string strKeyPassword,int nKeySize,int nKeyType)

## Parameters

[1]string the certificate file name  
[2]string the output directory  
[3]int the key password  
[4]int key size  
[5]int key type, 0=RSA, 1=DSA

## Return Values

[1]bool return true if created SSH key file successfully, otherwise return false

## Remarks

This function is used for creating an SSH key file .

---

**c\_GetSSHKeyList**()

## Parameters

nil

## Return Values

[1]table the list of all SSH configuration (formatted as {name,state})

**Remarks**

Return all the SSH key configurations.

---

**c\_GetSSHKey**(string strName)

**Parameters**

[1]string the SSH key configuration name

**Return Values**

[1]table a table formatted as {ssh\_keypath,ssh\_password}.

**Remarks**

Get an SSH key configuration table on success, or nil on error.

---

**c\_CheckSSHKey**(string strName)

**Parameters**

[1]string the SSH key configuration name

**Return Values**

[1]bool return true if SSH key is valid, otherwise return false

**Remarks**

Check whether the specified SSH key is valid.

---

**c\_AddSSHKey**(string strName,string strKeyPath,string strPassword)

**Parameters**

[1]string the certificate configuration name

[2]string the key file path

[3]string the ssh password

**Return Values**

[1]bool return true if added an SSH key configuration successfully, otherwise return false

**Remarks**

Add or modify an SSH key configuration.

---

**c\_DeleteSSHKey**(string strName)

**Parameters**

[1]string the SSH key configuration name

**Return Values**

[1]bool return true if deleted an SSH key configuration successfully, otherwise return false

**Remarks**

Delete an SSH key configuration.

---

## **c\_GetSMTPList()**

### **Parameters**

nil

### **Return Values**

[1]string smtp configuration list.

### **Remarks**

Return all the smtp configurations, and configuration name in string is separated with a carriage return sign.

---

## **c\_GetSMTP**(string strName)

### **Parameters**

[1]string the smtp configuration name

### **Return Values**

[1]table a table formatted as {sender\_name,sender\_email,need\_auth,smtp\_server,smtp\_port,smtp\_username,smtp\_password,use\_ssl}.

### **Remarks**

Return an smtp configuration table on success, or nil on error.

---

## **c\_AddSMTP**(string strName,string strSenderName,string strSenderEmail,bool bSmtplibNeedAuth,string strSmtpServer,int nSmtpPort,string strSmtpUserName,string strSmtpPassword,bool bUseSsl)

### **Parameters**

[1]string the smtp configure name  
[2]string the sender's name  
[3]string the sender's email address  
[4]bool need auth?, true=yes,false=no.  
[5]string the smtp server's address  
[6]int the smtp server's port  
[7]string the smtp server's username  
[8]string the smtp server's password  
[9]bool need SSL?, true=yes,false=no.

### **Return Values**

nil

### **Remarks**

Add or modify an smtp configuration.

---

## **c\_DeleteSMTP**(string strName)

### **Parameters**

[1]string the smtp configuration name

### **Return Values**

[1]bool return true if deleted an smtp configuration successfully, otherwise return false

### **Remarks**

Delete an smtp configuration.

---

## **c\_GetTaskList()**

### **Parameters**

nil

### **Return Values**

[1]table the system task(formatted as {taskname,type,day,executed,datefrom,timefrom}) list table

### **Remarks**

Return all the system tasks.

---

## **c\_GetTask**(string strTaskName)

### **Parameters**

[1]string the task scheduler name

### **Return Values**

[1]table a table formatted as {taskname,type,day,executed,datefrom,timefrom,script}.

### **Remarks**

Get a task scheduler table on success, or nil on error.

---

## **c\_AddTask**(string strTaskName,int nType,int nDay,string strDatefrom,string strTimefrom,string strScript,bool bReset)

### **Parameters**

[1]string the task scheduler name

[2]int the task scheduler type,0=one time task,1=hourly task,2=daily task,3=weekly task,4=monthly task

[3]int means the day number when you select weekly task or monthly task.

[4]string the date string of first starting task, such as 2009-9-9

[5]string the time string of first starting task, such as 09:09:09

[6]string the lua script text for the task.

[7]bool whether to reset the task, true=yes, false=no.

### **Return Values**

nil

### **Remarks**

Add or modify a system task scheduler.

---

## **c\_DeleteTask**(string strTaskName)

### **Parameters**

[1]string the task scheduler name

### **Return Values**

[1]bool return true if deleted system task scheduler successfully, otherwise return false

### **Remarks**

Delete a system task scheduler.

---

**c\_TaskExist**(string strTaskName)

**Parameters**

[1]string the task scheduler name

**Return Values**

[1]bool return true if the specified task scheduler exists, otherwise return false

**Remarks**

Check whether the specified task scheduler exists.

Get/Set Options

**c\_SetOptionInt**(string strDomain,int nKey,int nValue)

**Parameters**

[1]string the domain name

[2]int the key of domain option

[3]int the integer value of domain option

**Return Values**

nil

**Remarks**

Set the integer value of a domain option.

for example:

```
c_SetOptionInt("domain", DOPTION_DATA_ACCESS_INTERFACE_INT, 1)
```

DOPTION\_DATA\_ACCESS\_INTERFACE\_INT is a macro definition of domain options, you can check them at the bottom.

---

**c\_SetOptionStr**(string strDomain,int nKey,string strValue)

**Parameters**

[1]string the domain name

[2]int the key of domain option

[3]string the string value of domain option

**Return Values**

nil

**Remarks**

This function is similar to c\_SetOptionInt(), it is used for setting the string value of a domain option.

---

**c\_GetOptionInt**(string strDomain,int nKey)

**Parameters**

[1]string the domain name

[2]int the key of domain option

## Return Values

[1]int the integer value of domain option

## Remarks

Get the integer value of a domain option.

for example:

```
local interface = c_GetOptionInt("domain", DOPTION_DATA_ACCESS_INTERFACE_INT)
```

---

**c\_GetOptionStr**(string strDomain,int nKey)

## Parameters

[1]string the domain name

[2]int the key of domain option

## Return Values

[1]string the string value of domain option

## Remarks

Get the string value of a domain option.

---

**c\_SetGlobalOptionInt**(int nKey,int nValue)

## Parameters

[1]int the key of global option

[2]int the integer value of global option

## Return Values

nil

## Remarks

Set the integer value of a global option.

for example:

```
c_SetGlobalOptionInt(GOOPTION_SYSTEM_LOGFILE_ENABLE_INT, 1)
```

GOPTION\_SYSTEM\_LOGFILE\_ENABLE\_INT is a macro definition of global options, you can check them on the bottom..

---

**c\_SetGlobalOptionStr**(int nKey,string strValue)

## Parameters

[1]int the key of global option

[2]string the string value of global option

## Return Values

nil

## Remarks

This function is similar to c\_SetGlobalOptionInt(), it is used for setting the string value of a global option.

---

**c\_GetGlobalOptionInt**(int nKey)

## Parameters



[1]int the key of global option

## Return Values

[1]int the integer value of global option

## Remarks

Get the integer value of a global option.  
for example:

```
local enable_systemlog = c_GetGlobalOptionInt(GOPTION_SYSTEM_LOGFILE_ENABLE_INT)
```

---

**c\_GetGlobalOptionStr**(int nKey)

## Parameters

[1]int the key of global option

## Return Values

[1]string the string value of global option

## Remarks

Get the string value of a global option.

---

**c\_SetAdminOptionInt**(int nKey,int nValue)

## Parameters

[1]int the key of admin option

[2]int the integer value of admin option

## Return Values

nil

## Remarks

Set the integer value of admin option.

for example:

```
c_SetAdminOptionInt(ADMIN_OPTION_LOGFILE_ENABLE_INT, 1)
```

ADMIN\_OPTION\_LOGFILE\_ENABLE\_INT is a macro definition of admin options, you can check them on the bottom..

---

**c\_SetAdminOptionStr**(int nKey,string strValue)

## Parameters

[1]int the key of admin option

[2]string the string value of admin option

## Return Values

nil

## Remarks

This function is similar to c\_SetAdminOptionInt(), it is used for setting the string value of an admin option.

---

**c\_GetAdminOptionInt**(int nKey)

**Parameters**

[1]int the key of admin option

**Return Values**

[1]int the integer value of admin option

**Remarks**

Get the integer value of an admin option.

for example:

```
local enable_adminlog = c_GetAdminOptionInt(ADMIN_OPTION_LOGFILE_ENABLE_INT)
```

---

**c\_GetAdminOptionStr**(int nKey)**Parameters**

[1]int the key of admin option

**Return Values**

[1]string the string value of admin option

**Remarks**

Get the string value of an admin option.

**Miscellaneous****md5**(string strText)**Parameters**

[1]string the original string

**Return Values**

[1]string the md5 hash of a string

**Remarks**

Calculate the md5 hash of a string.

---

**c\_GetAppPath**()**Parameters**

nil

**Return Values**

[1]string the application installation path

**Remarks**

Returns the absolute path where Wing FTP Server is installed.

---

**c\_IsDir**(string strDirPath)**Parameters**

[1]string the directory path

**Return Values**

[1]bool return true if strDirPath is an existing directory, otherwise return false

**Remarks**

Tells whether the path is a real directory.

---

**c\_Mkdir**(string strDirPath)

**Parameters**

[1]string the directory path

**Return Values**

[1]bool return true if created a directory successfully, otherwise return false

**Remarks**

Attempts to make a directory specified by strPath.

---

**c\_RemoveFileDir**(string strPath)

**Parameters**

[1]string the directory path

**Return Values**

[1]bool return true if removed a file or a directory successfully, otherwise return false

**Remarks**

Attempts to remove a file or directory specified by strPath.

---

**c\_FileExist**(string strFilePath)

**Parameters**

[1]string the file path

**Return Values**

[1]bool return true if strFilePath is a regular file, otherwise return false

**Remarks**

Check whether a regular file exists.

---

**c\_GetFileTime**(string strFilePath)

**Parameters**

[1]string the file path

**Return Values**

[1]int the time when file content was changed

**Remarks**

Get the time when file content was changed.

---

## `c_GetTimeMS()`

### Parameters

nil

### Return Values

[1]int the milliseconds of current time

### Remarks

Get the milliseconds of current time.

---

## `c_GetTimeUS()`

### Parameters

nil

### Return Values

[1]int the microseconds of current time

### Remarks

Get the microseconds of current time.

---

## `c_GetRandom()`

### Parameters

nil

### Return Values

[1]int the random number

### Remarks

Generate a system random number.

---

## `c_GetDir(string strDirPath)`

### Parameters

[1]string the directory path

### Return Values

[1]string return an iterative directory name when no exception occurred, otherwise return **nil**.

### Remarks

Returns an iterative directory name inside the specified path `strDirPath`. So you can use this function for directory traversal,

for example:

```
for subdir in c_GetDir(rootpath) do
  print(subdir)
end
```

## `c_GetFileDir(string strDirPath)`

---

## Parameters

[1]string the directory path

## Return Values

[1]string return an iterative directory/file name when no exception occurred, otherwise return **nil**.

## Remarks

Returns an iterative directory/file name inside the specified path strDirPath. So you can use this function for directory/file traversal,

for example:

```
for subdirfile in c_GetFileDir(rootpath) do
  print(subdirfile)
end
```

---

## c\_GetRootDir()

### Parameters

nil

### Return Values

[1]table a table list of local drivers in windows, and in other OS its subdirectory table under "/"

### Remarks

Return local drivers string table in windows, subdirectory table under "/" in other OS.

for example:

```
for _,root_sub in pairs(c_GetRootDir()) do
  if root_sub ~= nil and root_sub["isdir"] == true then
    print(root_sub["dir"])
  end
end
```

---

## c\_GetLocalIpList()

### Parameters

nil

### Return Values

[1]string the local IP list string

### Remarks

Returns the local IP list, and IP in string is separated with a comma sign.

---

## c\_TranslateTime(string strTime)

### Parameters

[1]string the time string, such as "2009-09-09 00:00:00"

### Return Values

[1]int time\_t value of the time

### Remarks

Translate a time string to a time\_t value .

Administrator

**c\_CheckAdmin**(string strUserName,string strPassword)

**Parameters**

[1]string admin username  
[2]string admin password

**Return Values**

[1]bool return true if check administrator login is successful, otherwise return false

**Remarks**

Administrator login ok?

---

**c\_CheckAdminIp**(string strUserName,string strIp)

**Parameters**

[1]string admin username  
[2]string the client IP

**Return Values**

[1]bool return false when IP is banned, otherwise return true

**Remarks**

Check administrator login IP.

---

**c\_AdminLogout**(string strSessionId)

**Parameters**

[1]string the session ID

**Return Values**

nil

**Remarks**

Administrator logout.

---

**c\_AdminExist**(string strUserName)

**Parameters**

[1]string admin username

**Return Values**

[1]bool return true if specified admin exists, otherwise return false

**Remarks**

Check whether the specified administrator exists.

---

**c\_AddAdmin**(string strUserName, string strPassword, bool bReadonly, table tabAdminIpMasks)

## Parameters

[1]string admin username  
[2]string admin password  
[3]bool is read-only admin?  
[4]table administrator's IPmask(formated as {ip,refuse}) list table.

## Return Values

nil

## Remarks

Add or modify an administrator .

---

**c\_DeleteAdmin**(string strUserName)

## Parameters

[1]string admin username

## Return Values

nil

## Remarks

Delete an administrator.

---

**c\_GetAdmin**(string strUsername)

## Parameters

[1]string admin username

## Return Values

[1]table administrator data table formatted as {username,password,{ipmask table}}.

## Remarks

Return an administrator data table if the specified admin exists, otherwise return nil.

---

**c\_GetAdminType**()

## Parameters

[1]string the administrator name

## Return Values

[1]bool return true if administrator is read-only admin, otherwise return false

## Remarks

Get the administrator type.

---

**c\_GetAdminList**()

## Parameters

nil

## Return Values

[1]table administrator(formated as {username,lastlogin,lastip}) list table.

**Remarks**

Return administrators list.

---

**c\_AddAdminFailedIp**(string strIp)

**Parameters**

[1]string client IP

**Return Values**

nil

**Remarks**

Add the client IP to a banned list when login failed, for system anti-hammer.

---

**c\_RemoveAdminFailedIp**(string strIp)

**Parameters**

[1]string the client IP

**Return Values**

nil

**Remarks**

Remove the specified IP from banned list when login is successful, for system anti-hammer.

---

**c\_ClearAdminSession**()

**Parameters**

nil

**Return Values**

nil

**Remarks**

Clear all the expired admin sessions.

---

**c\_ChangeAdminListener**(int nHttpPort,bool bHttpSecure)

**Parameters**

[1]int administration listener's port

[1]bool whether to use SSL, 1=yes, 0=no

**Return Values**

[1]bool return true if change listener successfully, otherwise return false

**Remarks**

Modify the administrator's listener.

---



## `c_GetAdminIPMaskList()`

### Parameters

nil

### Return Values

[1]table the administrator IPmask(formated as {ip,refuse}) list table

### Remarks

Return administrator IPmask list.

---

## `c_SetAdminIPMaskList(table tblIpMask)`

### Parameters

[1]table the administrator IPmask(formated as {ip,refuse}) list table

### Return Values

nil

### Remarks

Set administrator IPmask list.

---

## `c_GetAdminLog()`

### Parameters

nil

### Return Values

[1]string admin log text

### Remarks

Get all the admin log text.

---

## `c_AddAdminLog(string strLog,int nType)`

### Parameters

[1]string the admin log text

[2]int the admin log type, you can check them at the bottom.

### Return Values

nil

### Remarks

Add an admin log.

## Database Schema

**Note:** *You needn't to create database schemas or tables by yourself, Wing FTP Server will create them automatically when configuring ODBC/Mysql settings(if you use Mysql, please create a mysql database first, the default database name is "wftp\_database").*

## Schemas for ODBC database

1) table for mapping relations of user and group([wftp\\_table\\_user\\_group](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
User_Name	varchar (64)	NOT NULL	the user name
Group_Name	varchar (64)	NOT NULL	the group name

2) table for user information([wftp\\_table\\_user](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
User_Name	varchar (64)	NOT NULL	the user name
EnableAccount	int		enable account?, 1=yes,0=no
EnablePassword	int		enable password?, 1=yes,0=no
Password	varchar (64)	NOT NULL	the user password
ProtocolType	int		enabled protocols mask, FTP=1,FTP(TLS)=2,FTP(SSL)=4,HTTP=8,HTTP(SSL)=16,SSH=32,if only allow FTP and HTTP, the mask number is 1+8 = 9
EnableExpire	int		enable account expire?, 1=yes,0=no
ExpireTime	varchar (32)		the account expire time,such as "2009-09-09 00:00:00"
MaxDownloadSpeed	int		max session download speed
MaxUploadSpeed	int		max session upload speed
MaxConnection	int		max number of connections
SessionNoCommandTimeOut	int		ftp connection timeout value
SessionNoTransferTimeOut	int		ftp idle timeout value
ConnectionPerIp	int		max number of connections per Ip
PasswordLength	int		max password length
ShowHiddenFile	int		show the hidden file?, 1=show,0=hide
CanChangePassword	int		user can change password?, 1=allow,0=deny

Field Name	Type	NULL	Description
CanSendMessageToServer	int		send chat message?, 1=allow,0=deny
CurrentCredit	varchar (64)		current ratio credit
RatioDownload	int		ratio download
RatioUpload	int		ratio upload
RatioCountMethod	int		ratio count method,0=count for files,1=count for bytes
EnableRatio	int		enable ratio?, 1=yes,0=no
MaxQuota	varchar (64)		max quota size
CurrentQuota	varchar (64)		current quota size
EnableQuota	int		enable quota?, 1=yes,0=no
NotesName	varchar (64)		note name text
NotesAddress	varchar (255)		note address text
NotesZipCode	varchar (64)		note zipcode text
NotesPhone	varchar (64)		note phone number text
NotesFax	varchar (64)		note fax number text
NotesEmail	varchar (128)		note email text
NotesMemo	varchar (255)		note memo text
TotalReceivedBytes	varchar (64)		total received bytes
TotalSentBytes	varchar (64)		total sent bytes
LoginCount	int		login count
FileDownload	int		total downloaded files
FileUpload	int		total uploaded files
FailedDownload	int		total download failed files
FailedUpload	int		total upload failed files
LastLoginIp	varchar (64)		the last login IP
LastLoginTime	varchar (64)		the last login time
EnableGroup	int		enable group?, 1=yes,0=no
EnableSchedule	int		enable access schedule?, 1=yes,0=no
LimitResetTime	int		transfer limit reset time
LimitResetType	int		transfer limit reset type, 0=reset never,1=reset hourly,2=reset daily,3=reset weekly,4=reset monthly
EnableUploadLimit	int		enable transfer limit upload?, 1=yes,0=no
CurLimitUploadSize	int		current upload size
MaxLimitUploadSize	int		max upload size
EnableDownloadLimit	int		enable transfer limit download?, 1=yes,0=no
CurLimitDownloadSize	int		current download size
MaxLimitDownloadSize	int		max download size

Field Name	Type	NULL	Description
SSHPublicKeyPath	varchar (255)		ssh public key path (optional)
EnableSSHPubKeyAuth	int		enable ssh public key authentication, 1=yes,0=no
SSHAuthMethod	int		ssh authentication method, 1=use both public key and password authentication,0=only use public key authentication
EnableWeblink	int		enable web download link, 1=yes,0=no
EnableUplink	int		enable web request (upload) link, 1=yes,0=no
EnableTwoFactor	int		enable two-factor authentication (TOTP) for Web Client, 1=yes,0=no
TwoFactorCode	varchar (32)		TOTP secret code
ExtraInfo	varchar (255)		extra information for user account

### 3) table for user's directory([wftp\\_table\\_dir](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
User_Name	varchar (64)	NOT NULL	the user name
DirPath	varchar (255)	NOT NULL	the directory real path
DirAlias	varchar (128)		the directory virtual path
Home_Dir	int		is home directory?, 1=yes,0=no
File_Read	int		enable file read?, 1=yes,0=no
File_Write	int		enable file write?, 1=yes,0=no
File_Append	int		enable file append?, 1=yes,0=no
File_Delete	int		enable file delete?, 1=yes,0=no
Directory_List	int		enable directory list?, 1=yes,0=no
Directory_Rename	int		enable directory/file rename?, 1=yes,0=no
Directory_Make	int		enable directory create?, 1=yes,0=no
Directory_Delete	int		enable directory delete?, 1=yes,0=no
File_Rename	int		enable file rename?, 1=yes,0=no
Zip_File	int		enable zip files?, 1=yes,0=no
Unzip_File	int		enable unzip files?, 1=yes,0=no

### 4) table for user's filemask([wftp\\_table\\_filemask](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
User_Name	varchar (64)	NOT NULL	the user name
Filename	varchar (128)	NOT NULL	the file mask
Refuse	int		refuse access?, 1=yes,0=no

5) table for user's ipmask([wftp\\_table\\_ipmask](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
User_Name	varchar (64)	NOT NULL	the user name
Ip	varchar (128)	NOT NULL	the ip mask
Refuse	int		refuse access?, 1=yes,0=no

6) table for user's access schedule([wftp\\_table\\_schedule](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
User_Name	varchar (64)	NOT NULL	the user name
Weekday	int		the day for access schedule. 0=Sunday, 1=Monday, 2=Tuesday, 3=Wednesday, 4=Thursday, 5=Friday, 6=Saturday
TimeFrom	varchar (32)		start time for access
TimeTo	varchar (32)		end time for access

7) table for group information([wftp\\_table\\_group](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
Group_Name	varchar (64)	NOT NULL	the group name

Field Name	Type	NULL	Description
MaxDownloadSpeed	int		max session download speed
MaxUploadSpeed	int		max session upload speed
MaxConnection	int		max number of connections
SessionNoCommandTimeOut	int		ftp connection timeout value
SessionNoTransferTimeOut	int		ftp idle timeout value
ConnectionPerIp	int		max number of connections per Ip
ShowHiddenFile	int		show the hidden file?, 1=show,0=hide
CanSendMessageToServer	int		send chat message?, 1=allow,0=deny
TotalReceivedBytes	varchar (64)		total received bytes
TotalSentBytes	varchar (64)		total sent bytes
LoginCount	int		login count
FileDownload	int		total downloaded files
FileUpload	int		total uploaded files
FailedDownload	int		total download failed files
FailedUpload	int		total upload failed files

8) table for group's directory([wftp\\_table\\_dir2](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
Group_Name	varchar (64)	NOT NULL	the group name
DirPath	varchar (255)	NOT NULL	the directory real path
DirAlias	varchar (128)		the directory virtual path
Home_Dir	int		is home directory?, 1=yes,0=no
File_Read	int		enable file read?, 1=yes,0=no
File_Write	int		enable file write?, 1=yes,0=no
File_Append	int		enable file append?, 1=yes,0=no
File_Delete	int		enable file delete?, 1=yes,0=no
Directory_List	int		enable directory list?, 1=yes,0=no
Directory_Rename	int		enable directory/file rename?, 1=yes,0=no
Directory_Make	int		enable directory create?, 1=yes,0=no
Directory_Delete	int		enable directory delete?, 1=yes,0=no
File_Rename	int		enable file rename?, 1=yes,0=no

Field Name	Type	NULL	Description
Zip_File	int		enable zip files?, 1=yes,0=no
Unzip_File	int		enable unzip files?, 1=yes,0=no

9) table for group's filemask([wftp\\_table\\_filemask2](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
Group_Name	varchar (64)	NOT NULL	the group name
Filename	varchar (128)	NOT NULL	the file mask
Refuse	int		refuse access?, 1=yes,0=no

10) table for group's ipmask([wftp\\_table\\_ipmask2](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
Group_Name	varchar (64)	NOT NULL	the group name
Ip	varchar (128)	NOT NULL	the ip mask
Refuse	int		refuse access?, 1=yes,0=no

## Schemas for Mysql database

1) table for mapping relations of user and group([wftp\\_mysqltable\\_user\\_group](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
User_Name	varchar (64)	NOT	the user name

Field Name	Type	NULL	Description
		NULL	
Group_Name	varchar (64)	NOT NULL	the group name

2) table for user information([wftp\\_mysqltable\\_user](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
User_Name	varchar (64)	NOT NULL	the user name
EnableAccount	int(11)		enable account?, 1=yes,0=no
EnablePassword	int(11)		enable password?, 1=yes,0=no
Password	varchar (64)	NOT NULL	the user password
ProtocolType	int(11)		enabled protocols mask, FTP=1,FTP(TLS)=2,FTP(SSL)=4,HTTP=8,HTTP(SSL)=16,SSH=32,if only allow FTP and HTTP, the mask number is 1+8 = 9
EnableExpire	int(11)		enable account expire?, 1=yes,0=no
ExpireTime	varchar (32)		the account expire time,such as "2009-09-09 00:00:00"
MaxDownloadSpeed	int(11)		max session download speed
MaxUploadSpeed	int(11)		max session upload speed
MaxConnection	int(11)		max number of connections
SessionNoCommandTimeOut	int(11)		ftp connection timeout value
SessionNoTransferTimeOut	int(11)		ftp idle timeout value
ConnectionPerIp	int(11)		max number of connections per ip
PasswordLength	int(11)		max password length
ShowHiddenFile	int(11)		show the hidden file?, 1=show,0=hide
CanChangePassword	int(11)		user can change password?, 1=allow,0=deny
CanSendMessageToServer	int(11)		send chat message?, 1=allow,0=deny
CurrentCredit	bigint(20)		current ratio credit
RatioDownload	int(11)		ratio download
RatioUpload	int(11)		ratio upload
RatioCountMethod	int(11)		ratio count method,0=count for files,1=count for bytes
EnableRatio	int(11)		enable ratio?, 1=yes,0=no
MaxQuota	bigint(20)		max quota size



Field Name	Type	NULL	Description
CurrentQuota	bigint(20)		current quota size
EnableQuota	int(11)		enable quota?, 1=yes,0=no
NotesName	varchar (64)		note name text
NotesAddress	varchar (255)		note address text
NotesZipCode	varchar (64)		note zipcode text
NotesPhone	varchar (64)		note phone number text
NotesFax	varchar (64)		note fax number text
NotesEmail	varchar (128)		note email text
NotesMemo	varchar (255)		note memo text
TotalReceivedBytes	bigint(20)		total received bytes
TotalSentBytes	bigint(20)		total sent bytes
LoginCount	int(11)		login count
FileDownload	int(11)		total downloaded files
FileUpload	int(11)		total uploaded files
FailedDownload	int(11)		total download failed files
FailedUpload	int(11)		total upload failed files
LastLoginIp	varchar (64)		the last login IP
LastLoginTime	varchar (64)		the last login time
EnableGroup	int(11)		enable group?, 1=yes,0=no
EnableSchedule	int(11)		enable access schedule?, 1=yes,0=no
LimitResetTime	int(11)		transfer limit reset time
LimitResetType	int(11)		transfer limit reset type, 0=reset never,1=reset hourly,2=reset daily,3=reset weekly,4=reset monthly
EnableUploadLimit	int(11)		enable transfer limit upload?, 1=yes,0=no
CurLimitUploadSize	int(11)		current upload size
MaxLimitUploadSize	int(11)		max upload size
EnableDownloadLimit	int(11)		enable transfer limit download?, 1=yes,0=no
CurLimitDownLoadSize	int(11)		current download size
MaxLimitDownloadSize	int(11)		max download size
SSHPublicKeyPath	varchar (255)		SSH public key path (optional)
EnableSSHPubKeyAuth	int(11)		enable ssh public key authentication, 1=yes,0=no
SSHAuthMethod	int(11)		ssh authentication method, 1=use both public key and password authentication,0=only use public key authentication
EnableWeblink	int(11)		enable web download link, 1=yes,0=no
EnableUplink	int(11)		enable web request (upload) link, 1=yes,0=no
EnableTwoFactor	int(11)		enable two-factor authentication (TOTP) for Web Client, 1=yes,0=no
TwoFactorCode	varchar (32)		TOTP secret code

Field Name	Type	NULL	Description
ExtralInfo	varchar (255)		extra information for user account

3) table for user's directory([wftp\\_mysqltable\\_dir](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
User_Name	varchar (64)	NOT NULL	the user name
DirPath	varchar (255)	NOT NULL	the directory real path
DirAlias	varchar (128)		the directory virtual path
Home_Dir	int(11)		is home directory?, 1=yes,0=no
File_Read	int(11)		enable file read?, 1=yes,0=no
File_Write	int(11)		enable file write?, 1=yes,0=no
File_Append	int(11)		enable file append?, 1=yes,0=no
File_Delete	int(11)		enable file delete?, 1=yes,0=no
Directory_List	int(11)		enable directory list?, 1=yes,0=no
Directory_Rename	int(11)		enable directory/file rename?, 1=yes,0=no
Directory_Make	int(11)		enable directory create?, 1=yes,0=no
Directory_Delete	int(11)		enable directory delete?, 1=yes,0=no
File_Rename	int(11)		enable file rename?, 1=yes,0=no
Zip_File	int(11)		enable zip files?, 1=yes,0=no
Unzip_File	int(11)		enable unzip files?, 1=yes,0=no

4) table for user's filemask([wftp\\_mysqltable\\_filemask](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
User_Name	varchar (64)	NOT NULL	the user name
Filename	varchar (128)	NOT NULL	the file mask
Refuse	int(11)		refuse access?, 1=yes,0=no

5) table for user's ipmask([wftp\\_mysqltable\\_ipmask](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
User_Name	varchar (64)	NOT NULL	the user name
Ip	varchar (128)	NOT NULL	the ip mask
Refuse	int(11)		refuse access?, 1=yes,0=no

6) table for user's access schedule([wftp\\_mysqltable\\_schedule](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
User_Name	varchar (64)	NOT NULL	the user name
Weekday	int(11)		the day for access schedule. 0=Sunday, 1=Monday, 2=Tuesday, 3=Wednesday, 4=Thursday, 5=Friday, 6=Saturday
TimeFrom	varchar (32)		start time for access
TimeTo	varchar (32)		end time for access

7) table for group information([wftp\\_mysqltable\\_group](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
Group_Name	varchar (64)	NOT NULL	the group name
MaxDownloadSpeed	int(11)		max session download speed
MaxUploadSpeed	int(11)		max session upload speed
MaxConnection	int(11)		max number of connections
SessionNoCommandTimeOut	int(11)		ftp connection timeout value
SessionNoTransferTimeOut	int(11)		ftp idle timeout value
ConnectionPerIp	int(11)		max number of connections per Ip
ShowHiddenFile	int(11)		show the hidden file?, 1=show,0=hide
CanSendMessageToServer	int(11)		send chat message?, 1=allow,0=deny
TotalReceivedBytes	bigint(20)		total received bytes

Field Name	Type	NULL	Description
TotalSentBytes	bigint(20)		total sent bytes
LoginCount	int(11)		login count
FileDownload	int(11)		total downloaded files
FileUpload	int(11)		total uploaded files
FailedDownload	int(11)		total download failed files
FailedUpload	int(11)		total upload failed files

8) table for group's directory([wftp\\_mysqltable\\_dir2](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
Group_Name	varchar (64)	NOT NULL	the group name
DirPath	varchar (255)	NOT NULL	the directory real path
DirAlias	varchar (128)		the directory virtual path
Home_Dir	int(11)		is home directory?, 1=yes,0=no
File_Read	int(11)		enable file read?, 1=yes,0=no
File_Write	int(11)		enable file write?, 1=yes,0=no
File_Append	int(11)		enable file append?, 1=yes,0=no
File_Delete	int(11)		enable file delete?, 1=yes,0=no
Directory_List	int(11)		enable directory list?, 1=yes,0=no
Directory_Rename	int(11)		enable directory/file rename?, 1=yes,0=no
Directory_Make	int(11)		enable directory create?, 1=yes,0=no
Directory_Delete	int(11)		enable directory delete?, 1=yes,0=no
File_Rename	int(11)		enable file rename?, 1=yes,0=no
Zip_File	int(11)		enable zip files?, 1=yes,0=no
Unzip_File	int(11)		enable unzip files?, 1=yes,0=no

9) table for group's filemask([wftp\\_mysqltable\\_filemask2](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
Group_Name	varchar (64)	NOT NULL	the group name

Field Name	Type	NULL	Description
Filename	varchar (128)	NOT NULL	the file mask
Refuse	int(11)		refuse access?, 1=yes,0=no

10) table for group's ipmask([wftp\\_mysqltable\\_ipmask2](#))

Field Name	Type	NULL	Description
Domain	varchar (64)	NOT NULL	the domain name
Group_Name	varchar (64)	NOT NULL	the group name
Ip	varchar (128)	NOT NULL	the ip mask
Refuse	int(11)		refuse access?, 1=yes,0=no