

**CompleteFTP**

**User's Guide**



# Table of Contents

## Overview

- [Product Overview](#)
- [Requirements](#)
- [How to select the edition](#)
- [CompleteFTP Server Software License Agreement](#)
- [CompleteFTP revision history](#)

## Installation

- [Installation](#)
- [Uninstalling](#)
- [How to run the server](#)
- [How to use the manager](#)
- [Migrating to another machine](#)
- [How to automate installation](#)
- [How to activate an installation of CompleteFTP](#)
- [Installing CompleteFTP in the Cloud](#)
- [How to upgrade](#)

## Tutorials

- [Getting started with CompleteFTP](#)
- [Getting started with Amazon EC2](#)
- [CompleteFTP: Connecting to CompleteFTP with Filezilla](#)
- [Connecting to CompleteFTP with WinSCP](#)

## Step-By-Step guides

- [Step-by-step guides](#)
- [Add a non-Windows user](#)
- [Add a Windows user \(Standard edition only\)](#)
- [Add a Windows user \(Professional edition and greater\)](#)
- [Set up public-key authentication for a user](#)
- [Allow all users from a Windows group to log in](#)
- [Authenticate users from a database](#)
- [Change the home folder of a user](#)
- [Add a network folder](#)
- [Allow users to share a folder \(Method 1\)](#)
- [Allow users to share a folder \(Method 2\)](#)
- [Allow users to share a folder \(Method 3\)](#)
- [Block connections from specific IP addresses \(blacklisting\)](#)
- [Accept connections only from specific IP addresses \(whitelisting\)](#)
- [Accept connections only from specific users at specific IP addresses](#)
- [Restrict some users to connect only from specific IP addresses](#)
- [E-mail notification when a file is uploaded](#)
- [Move a file to another directory after it's been uploaded](#)
- [Add a scheduled task](#)
- [Accepting connections from the Internet](#)
- [Share a file using CompleteBox](#)
- [How to customize the welcome page](#)
- [Host a web-app in CompleteFTP](#)
- [Add my own authentication method](#)
- [Configure SAML Single Sign On with OneLogin](#)
- [Request a CA certificate](#)
- [Install a CA certificate](#)
- [Activate CompleteFTP free](#)
- [Activate the paid editions](#)
- [Set up a cluster](#)

## Overview panel

- [Overview](#)

## Site configuration

- [Site/Sites](#)
- [FTP/FTPS](#)
- [SFTP/SCP/SSH](#)
- [HTTP/HTTPS settings](#)
- [Two Factor Authentication](#)
- [Account Management](#)
- [File Sharing Settings](#)
- [Filesystem settings](#)
- [General user settings](#)
- [IP Filtering and Auto-Banning](#)
- [Email Settings](#)
- [Limits and Timeouts](#)
- [Messages](#)
- [How to secure your server](#)
- [How to set a default Windows domain](#)
- [How to use HTTP/HTTPS](#)
- [How to use the File Manager](#)
- [How to format HTML listings](#)
- [How to customize the welcome page](#)
- [How to use CompleteFTP with firewalls](#)
- [How to set an external IP address](#)
- [How to set a PASV port range](#)
- [How to set up the server certificate](#)

[How to set up a CA SSL certificate](#)  
[How to generate a CSR](#)  
[How to purchase a CA SSL certificate](#)  
[How to install a CA SSL certificate](#)  
[How to import SSL certificates in other formats](#)  
[How to set up the server keys](#)  
[How to use SSH terminal access](#)  
[How to use SSH forwarding](#)  
[How to configure anonymous logins](#)  
[How to use multiple sites](#)  
[How to view the Admin site](#)  
[How to set up a secure server](#)  
[How to use HTTP](#)  
[How to use HTTPS](#)  
[How to perform HTTP uploads](#)

## **File system**

[File System](#)  
[How to create a virtual folder](#)  
[How to add a Windows folder](#)  
[How to add a Windows special folder](#)  
[How to view system folders](#)  
[How to create a network/macro folder](#)  
[Gateway folders](#)  
[Azure Share Folders](#)  
[Amazon S3 Folders](#)  
[How to insert macros into folder-names](#)  
[How to give users shared access to a folder](#)  
[How to set a folder's permissions](#)  
[How to enable encryption at rest](#)  
[How to list a folder-tree](#)  
[How to extend the filesystem](#)  
[How to stop executables being uploaded](#)  
[How to mount ZIP files as directories](#)  
[How to download a folder as a ZIP file](#)  
[How to enable backslash separators](#)  
[How to customize the welcome page](#)

## **File manager**

[File Manager](#)  
[Uploading Files](#)  
[Downloading Files](#)  
[Viewing Files](#)  
[Deleting Files](#)  
[Renaming File](#)  
[Adding Folders](#)  
[Deleting Folders](#)  
[Renaming Folders](#)  
[Share Files Privately](#)  
[Sharing Files Publicly](#)  
[Changing Languages](#)  
[Changing List View Type](#)

## **Users**

[Users](#)  
[General user settings](#)  
[How to add a user](#)  
[How to add a Windows user](#)  
[How to add a non-Windows user](#)  
[How to add multiple users](#)  
[How to change user properties](#)  
[How to retrieve/reset admin password](#)  
[How to view system users](#)  
[How to enable encryption at rest](#)  
[How to configure external database users](#)  
[How to configure Windows/AD group users](#)  
[How to set a default Windows domain](#)  
[How to configure gateway users](#)  
[How to configure Single Sign-ON \(SAML\) users](#)  
[How to configure LDAP users](#)  
[Two Factor Authentication](#)  
[How to set user quotas](#)  
[CompleteFTP: How to set up public key authentication](#)  
[How to control user access by IP address](#)  
[How to use SSH terminal access](#)  
[How to use SSH forwarding](#)  
[How to configure anonymous logins](#)  
[How to set up client certificates \(FTPS\)](#)  
[How to make a user's home dir root](#)  
[How to give users shared access to a folder](#)  
[How to set password policies](#)  
[How to make user accounts expire](#)  
[How to manage groups](#)  
[How to Export Users](#)

## **File sharing**

### [File sharing](#)

#### [How to enable file sharing](#)

#### [How to install CompleteBox](#)

#### [File sharing vs file storing](#)

#### [How to share files with CompleteBox](#)

#### [How to store files with CompleteBox](#)

#### [File synchronization](#)

#### [How to configure CompleteBox](#)

#### [CompleteBox Software License Agreement](#)

## **Clustering**

### [1. Introduction to CompleteFTP Clustering](#)

### [7. Security Considerations](#)

#### [Clustering: Management](#)

#### [Upgrading a Cluster](#)

#### [CompleteFTP Clustering - Troubleshooting](#)

## **Multiprotocol gateway**

### [Gateway](#)

#### [Gateway configuration](#)

#### [Gateway folders](#)

#### [How to configure gateway users](#)

#### [How to configure a gateway authenticator](#)

## **CompleteAgent**

### [How to use CompleteAgent](#)

## **Events**

### [Events](#)

#### [How to set up email notifications](#)

#### [How to set up process execution triggers](#)

#### [Scheduled Events](#)

#### [How to write FTP scripts](#)

#### [Script command reference](#)

#### [Notification macros](#)

#### [How to write your own events](#)

#### [How to write Batch Scripts](#)

#### [How to write PowerShell scripts](#)

## **Custom extensions**

### [Custom Extensions](#)

#### [Authenticator Class Reference](#)

#### [JSS-to-.NET Bridge Extensions](#)

#### [.NET Extensions](#)

##### [How to write a .NET extension](#)

##### [.NET Authentication Extensions](#)

##### [Simple .NET Authentication Extensions](#)

##### [Advanced .NET Authentication Extensions](#)

##### [.NET file-system extensions](#)

##### [.NET custom command extensions](#)

##### [.NET Event Extensions](#)

##### [.NET IP Filter Extensions](#)

##### [How to add logging to a .Net extension](#)

## **Scripting Guide**

### [Scripting Overview](#)

#### [JSS \(Javascript Server-Side\)](#)

##### [JSS API Reference](#)

##### [JSS Remote Procedure Calls \(RPC\)](#)

#### [Web-Apps](#)

##### [JSS Web-Apps](#)

##### [JSS Web-App Basics](#)

##### [Using templates](#)

##### [Web-App Examples](#)

#### [How to write JSS process triggers](#)

##### [How to write a JSS event](#)

##### [JSS event tutorial 1 - Upload file to server](#)

##### [JSS event tutorial 2 - Error handling](#)

##### [JSS event tutorial 3 - Multiple Uploading](#)

##### [JSS event tutorial 4 - Upload event-handler](#)

##### [JSS event tutorial 5 - Record events in a file](#)

##### [JSS event tutorial 6 - Move a file from this Path to another Path](#)

#### [JSS Extensions](#)

##### [How to write a JSS extension](#)

##### [JSS Authentication Extensions](#)

##### [Simple JSS Authentication Extension](#)

##### [Advanced JSS Authentication Extension](#)

##### [JSS File-System Extensions](#)

##### [Example: DB-based file-system extension](#)

##### [JSS Custom Command Extensions](#)

##### [JSS IP Filter Extensions](#)

#### [Sample scripts](#)

##### [Overview of Sample Scripts](#)

##### [Purge old files](#)

##### [Low drive-space email](#)

##### [Maintain user stats](#)



[Trigger mobile notifications from CompleteFTP](#)  
[Add user to group from SSH terminal](#)

## Monitoring

[Monitoring](#)  
[Real-time Logging](#)  
[How to view the server log files](#)  
[How to view the manager log files](#)  
[How to change the log file location](#)  
[How to enable auditing](#)  
[How to log to syslog](#)  
[How to produce clean log files](#)  
[How to get diagnostics for support](#)  
[Getting server diagnostics](#)

## Reporting

[Reporting](#)  
[Designing Custom Reports](#)

## Performance

[Performance](#)

## Administrator access

[Administrator Access](#)  
[How to change the administrator site settings](#)  
[How to manage the server remotely](#)  
[How to add extra administrator users](#)  
[How to do command-line administration](#)  
[Administration with UNIX-like scripts](#)  
[Administration with REST API - AdminAPI](#)  
[Administration with JSS scripts](#)  
[Administrator command-line](#)  
[How to secure your server](#)  
[Administration Log Feature](#)

## WebAdmin

[WebAdmin Overview](#)

### Users Panel

[Adding Users](#)  
[Editing Users](#)  
[Removing Users](#)

### Folders Panel

[Adding folders](#)  
[Editing Folders](#)  
[Removing Folders](#)

### Monitoring Panel

[Monitoring overview](#)  
[Killing connections](#)

## Security

[Security overview](#)  
[How to secure your server](#)  
[How to set password policies](#)  
[How to disable SSL 3.0, TLS 1.0, and TLS 1.1](#)

## Licensing

[Licensing overview](#)  
[How to activate an installation of CompleteFTP](#)  
[Migrating to another machine](#)  
[How to resolve restricted mode](#)  
[How to automatically import a license](#)  
[How to set up a disaster recovery \(DR\) server](#)  
[How to Activate Free Edition](#)  
[How to Activate Standard, Professional or Enterprise MFT](#)  
[Universal activation keys](#)

## Support

[Support](#)  
[How to get diagnostics for support](#)  
[How to contact support](#)  
[Frequently Asked Questions](#)  
[CompleteFTP revision history](#)  
[CompleteBox revision history](#)

## Reference

### Revision histories

[CompleteFTP revision history](#)  
[CompleteBox revision history](#)

### File transfer guide

#### FTP

[FTP Protocol Overview](#)  
[Active and Passive Modes](#)  
[FTP Commands](#)  
[Sample Scenarios](#)  
[Data Types](#)  
[Session Commands](#)  
[File Commands](#)  
[Directory Commands](#)

#### FTPS

[FTPS - Securing FTP with TLS](#)

[Implicit FTPS and Explicit FTPS](#)  
[Securing Control and Data Channels](#)  
[FTPS Commands](#)  
[FTPS Usage](#)

## **Security**

[The Essentials of FTP Security](#)  
[Public Key Cryptography](#)  
[Certificates and Certificate Authorities \(CAs\)](#)  
[Obtaining Keys and Certificates](#)  
[Server and Client Validation](#)  
[Hostname Checking](#)  
[Selecting Ciphers](#)

## **SFTP and SCP**

[SFTP and SCP](#)  
[SSH - Secure Shell](#)  
[SCP - Secure Copy](#)  
[SFTP - SSH File Transfer Protocol](#)  
[Comparison of FTPS and SFTP](#)

## **HTTP and HTTPS**

[HTTP Protocol Overview](#)  
[HTTP methods](#)  
[HTTP sessions](#)  
[HTTPS](#)

## **IETF Standards**

[About This Manual](#)  
[CompleteFTP Custom Configurer: cftpconfig](#)  
[CompleteFTP third-party copyright notices](#)  
[mustache\(5\) – Logic-less templates.](#)  
[How to activate an installation of CompleteFTP](#)

# Product Overview

**CompleteFTP** is a high performance Windows file server supporting secure file transfer via FTPS(FTP over SSL), SFTP (FTP over SSH), HTTP, HTTPS, SCP and S. CompleteFTP also supports web applications written in Javascript, and collaborative file sharing.

## Editions

**CompleteFTP** comes in four different editions:

Edition	Features
Free	Supports FTP and FTPS
Standard	Free features + SFTP, Windows users
Professional	Standard features + SSH, SCP, HTTPS, server-side Javascript, triggers, file sharing, and more
Enterprise MFT	Professional features + Encryption at rest, Azure and Amazon folders, automation, customization, clustering, and more. Our Managed File Transfer edition.

## Key Features

Easy to install, fast, and easy to configure, CompleteFTP is the ideal server solution for securing file transfers on Windows, either on your premises or in the cloud. Features include:

- Deploy on cloud platforms such as Windows Azure and Amazon Elastic Compute Cloud (EC2) as well as on your premises.
- Full support for **FTP and FTPS**(FTP over SSL) [All editions]. Includes implicit and explicit FTPS, and TLS 1.2. Now includes **Elliptic Curve Cryptography (ECC) Cipher Suites** (from 11.0.6) and **AES Galois Counter Mode (GCM) Cipher Suites** (from 12.1.2).
- Full support for **SFTP** (FTP over SSH) [Standard Edition or better]. Includes support for **ECDSA** key exchange algorithms and keys, and **aes128-gcm@openssh.com** and **aes256-gcm@openssh.com** ciphers.
- Support for **HTTP and HTTPS**[Professional Edition or better], including support for **JavaScript FileManager**, directory listings and HTTP uploads (unlimited in size!).Custom MIME types are supported.
- Support for robust, automated uploads via **CompleteAgent client** [Enterprise MFT required].
- Full support for **SCP** [Professional Edition or better].
- Support for **Two factor authentication** [Enterprise MFT].
- Multiple administration options:
  - **Modern Windows application** for full control of server configuration [All editions].
  - **Mobile-friendly web-app** for user management and monitoring [Professional Edition or better].
  - **Command-Line Interface (CLI)** [Enterprise MFT only].
  - **Configuration Object Model** accessible through **JavaScript (JSS)** [Enterprise MFT only].
- Support for **encryption at rest** [Enterprise MFT only].
- Support for **SSH tunneling/SSH local port forwarding** [Professional Edition or better].
- Support for **SSH terminals** [Professional Edition or better].
- Support for **collaborative file sharing** [Professional Edition or better].
- Support for **CompleteFTP as a SAML Service Provider** [Enterprise MFT only].
- Support for web applications using **server side Javascript (JSS)** [Professional Edition or better].
- Support for **customized and automated installation** [Enterprise MFT only].
- Support for **custom Javascript or .NET extensions** for authentication, custom file-systems, events, custom commands and IP filters [Enterprise MFT only].
- Support for **generation of Certificate Signing Requests (CSRs)** for submission to a Certificate Authority (CA) to obtain a CA certificate [All editions].
- SSL self-signed **certificate generator** [All editions].
- Ability to act as a **multiprotocol gateway** for other servers. Use CompleteFTP to provide secure access via FTPS or SFTP to an existing FTP server. Add FTP or FTF support to an existing SFTP server [Enterprise MFT only].
- Support for file sizes greater than 4 GB [All editions].
- Unlimited number of users [All editions].
- User **disk and bandwidth quotas** [Professional Edition or better].
- Extensive real-time charts for performance **monitoring** [Professional Edition or better].
- **Robust Reporting Feature** [Professional and Enterprise MFT Editions]: Gain insights into server events with structured, customizable reports. Monitor server activity, track use behavior, and optimize performance with data presented in a user-friendly format. The reporting feature includes options to view, filter, and analyze event information within a specified date range, with added flexibility in the Enterprise MFT Edition to create custom reports tailored to specific needs.
- **Email notification** of events such as upload, download, delete [Professional Edition or better].
- **Process Triggers** for launching server processes, Powershell scripts, batch scripts or FTP scripts when events such as uploading or downloading occur [Professional Edition or better].
- **Auto-banning** [Available in all editions - customizable in Professional Edition or better].
- **IP filtering** [Professional Edition or better].
- **Remote administration** [All editions].
- Full support for IPv6 (from 13.1.0).
- **Virtual File System** [All editions].
- **ZIP-file navigation** [Professional Edition or better]. Users can change into folders within ZIP files and download specific files without downloading the entire ZIP file.
- **CompleteFTP (non- Windows) users** [All editions].
- **Windows users** [Standard Edition or better].
- **Automatic Windows / AD (Active Directory) users** [Professional Edition or better]. Support for Windows domain users and can set a default domain.
- Support for **external users** (e.g. stored in an external database) [Professional Edition or better].
- **Step-by-step guides** built into management interface.



# Requirements

**CompleteFTP** runs on 64 bit versions of Windows, including Windows Server 2022, 2019, 2016, 2012, 2008 and 2003, as well as Windows 11, 10 and 8. It also requires .NET Framework 4.7.2 or later.

**Legacy environments:** Versions prior to 22.0 will run on 32 bit and 64 bit versions of Windows. Versions up to 22.1.3 require only .NET Framework 4.

**Cloud:** CompleteFTP also runs on cloud platforms such as Windows Azure and Amazon's [Elastic Compute Cloud \(EC2\)](#).

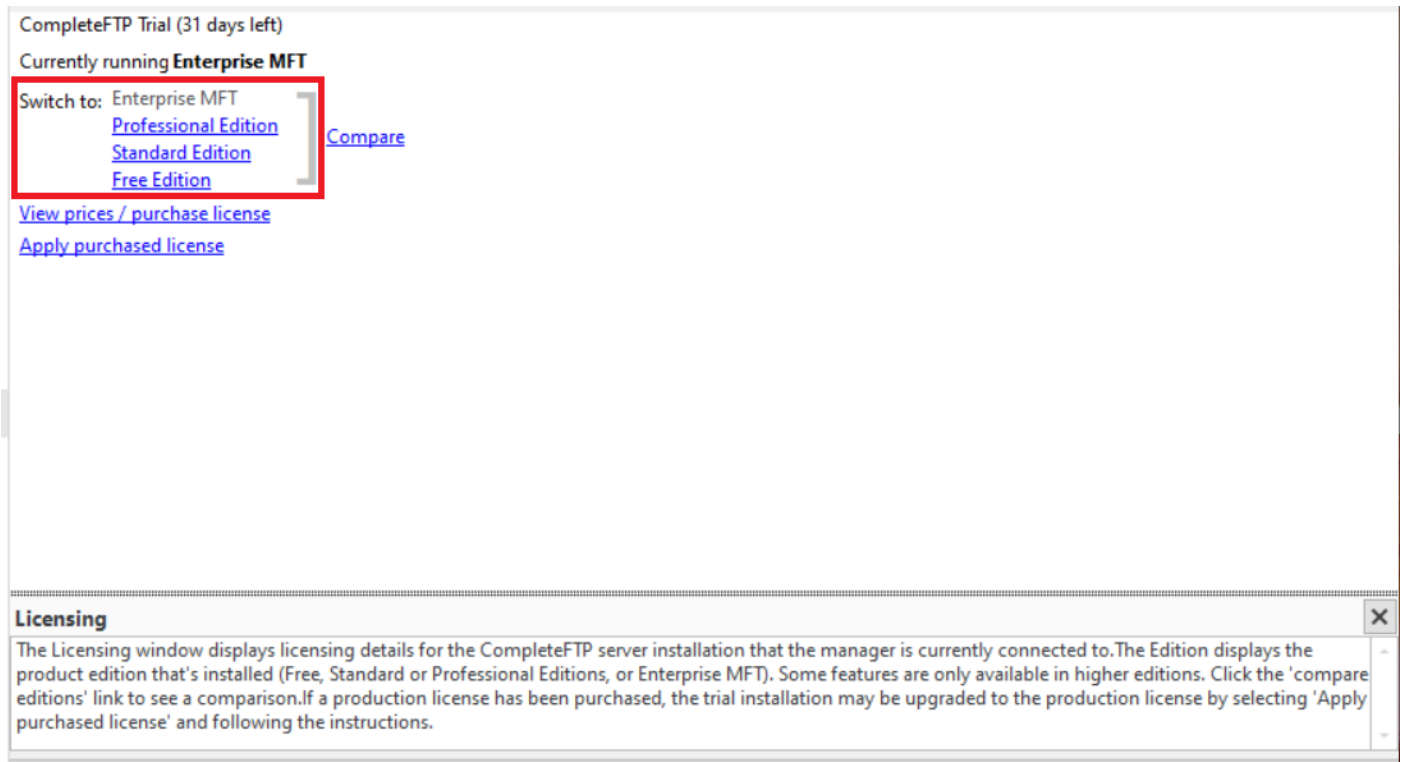
**CPU:** In terms of processing power, we recommend at least a dual core CPU, as encrypted transfers are CPU intensive. If the network connection is fast, CPU will be the bottleneck. If CompleteFTP is being run in a VM, we recommend allocating at least 4 vCPUs to the VM.

**Memory:** CompleteFTP's memory usage is strongly correlated with the number of concurrent users and transfers. We recommend allocating 16GB.

# How to select the edition

**CompleteFTP** comes in four editions: *Free Edition*, *Standard Edition*, *Professional Edition* and *Enterprise MFT*.

A comparison of the four editions can be found [here](#). The trial installer will start you off using Enterprise MFT, but you can easily downgrade and upgrade between editions in the *Licensing* panel during the trial period.



The differences between the editions are shown in a table [here](#).

Note that CompleteFTP is licensed on a per machine basis, i.e. one license is required for each machine it is installed on, unless a *Universal Activation Key (UAK)* is purchased. This permits installation on any number of machines in the organisation, and online activation is not required. It is available for certain Volume Licenses, on request.

## Upgrading

If you have purchased the Standard or Professional Edition but subsequently decide that you require the features of a higher edition, you can upgrade for the cost of the price difference plus a USD 30 administration fee - please [contact us](#) to arrange this (and include your purchase reference in your email).

# CompleteFTP Server Software License Agreement

We **Enterprise Distributed Technologies Ltd** grant you license to use this Software on the following terms:

## 1. License and support

1. This Software is licensed and not sold to you. Your use of this Software is governed solely by the terms of this License. This License is personal to you.
2. This Software is owned by us, save for certain elements of it which are owned by The KPD-Team; The Legion of the Bouncy Castle, Inc; Chew Keong Tan; Routrek Networks, Inc; ComponentAce; Developer Express Inc; The Mentalis.org Team; Novell, Inc; Motus Technologies Inc; Studio 42; Evgen Noskov; The Outercurve Foundation; Digitaliseringsstyrelsen; The Legion of the Bouncy Castle Inc., and akveo.com. This Software is protected by copyright law, and the owners reserve ownership of all Intellectual Property Rights in it, and all rights other than those expressly granted by this Agreement. Copyright notices for other parties can be found [here](#).
3. You must take reasonable steps to protect the Software from unauthorised copying, publication, disclosure or distribution.
4. Technical support is initially available via email for 12 months after the date of purchase of a Full License (and can be renewed annually). We will endeavour to respond to queries as soon as possible, but cannot guarantee a response within any particular time frame. Note that if excessive use is being made of technical support, an additional payment not exceeding the payment required for 12 months' technical support may be required. Excessive use is defined on a calendar year basis as exceeding five separate issues per server license. All technical support queries must be submitted via a designated support person.
5. Support agreements must be renewed within 30 days of expiry, otherwise the higher expired support renewal price will apply.
6. Where use is permitted by this License, such use includes
  1. the making of no more than a reasonable number of backup copies of the Software; and
  2. the right to utilise and make prints for your own personal use (but not otherwise copy) any instructional and/or operational manuals relating to the Software.

## 2. Free License

1. The Free License is applicable to the Free Edition of this Software. It does not apply to the Standard Edition, the Professional Edition, or Enterprise MFT.
2. You are entitled to install the Free Edition of the Server Application on any number of machines within your organization.
3. Administrator Application. We grant to you the right to install and use the administrator application on an unlimited number of machines.
4. Technical support is not included with the Free Edition, but may be purchased separately. If support has not been purchased, queries can be posted on the Enterprise Distributed Technologies Ltd public forums, but responses and response times are not guaranteed.
5. Free Licenses can be converted to Full Licenses by payment of the applicable License fee.
6. You may not use the *edftp.exe* program independently of CompleteFTP. It cannot be invoked or executed by any other program or from the command-line.
7. You may not modify the Software. You may not disassemble, decompile, or reverse engineer the Software, or otherwise attempt to discover the source code of the Software.

## 3. Trial License

Until and unless you buy a Full License or convert to a Free License for the use of the Software:

1. Your rights are limited to
  1. use for evaluation purposes only, and
  2. use for 30 days following first installation only; thereafter, the Trial License converts to a Developer License (the 30 day Trial License period may be extended on request).
2. You may not use the *edftp.exe* program independently of CompleteFTP. It cannot be invoked or executed by any other program or from the command-line.
3. You may not modify the Software. You may not disassemble, decompile, or reverse engineer the Software, or otherwise attempt to discover the source code of the Software.

## 4. Developer License

Until and unless you buy a Full License or convert to a Free License for the use of the Software:

1. Your rights are limited to use for evaluation and code-development purposes only, indefinitely.
2. You may not access the Software in any way from a machine other than the machine on which the Software is installed.
3. You may not use the *edftp.exe* program independently of CompleteFTP. It cannot be invoked or executed by any other program or from the command-line.
4. You may not modify the Software. You may not disassemble, decompile, or reverse engineer the Software, or otherwise attempt to discover the source code of the Software.

## 5. Full License

Subject to payment of the full License fee:

1. Server Application. We grant to you a non-transferable and non-exclusive license to install the Server Application for use on the number of server machines for which you have purchased separate licenses.
2. In the case of an unlimited Volume License, you are entitled to install the Server Application on as many machines as you wish within the business unit which purchased the license.
3. Licenses are only applicable only within the country of purchase.
4. Administrator Application. We grant to you the right to install and use the administrator application on an unlimited number of machines provided that you are licensed to use the Server Application.
5. The *edftp.exe* program is not designed to be run independently of CompleteFTP, and is not supported if it is used independently of CompleteFTP.
6. You may not modify the Software. You may not disassemble, decompile, or reverse engineer the Software, or otherwise attempt to discover the source code of the Software.

## 6. CompleteBox User Licenses

Subject to payment of the full license fee:

1. We grant to you fifty (50) CompleteBox User Licenses in the case of the Professional Edition, and 1000 CompleteBox User Licenses in the case of Enterprise MFT. Each CompleteBox User License permits a single CompleteBox user to connect to the Server Application for the purposes of file sharing. A CompleteBox user is a human being, not a user account shared between people. Each person using CompleteBox requires a CompleteBox User License. They may use their license from any client machine, i.e. a

person with a CompleteBox User License can connect from multiple machines. File sharing is not supported in the Standard Edition, and no CompleteBox User Licenses are granted.

2. All CompleteBox technical support queries must be submitted via the designated support person for the Server Application.

## **7. Warranties, and Limitation of Liability:**

1. You acknowledge that
  1. you have had the opportunity to evaluate the Software without charge, and that you have satisfied yourself that it meets your requirements in all material respects
  2. it is not technically practicable to guarantee software to be error-free, and you agree that if any such errors are found to exist they shall not constitute a breach of this Agreement
  3. the Software is provided for use by you as an expert, and that where you decide to incorporate the Software in an application created by you, it is solely your responsibility to carry out all appropriate testing and to determine its suitability for your intended purpose. You will therefore indemnify us against any and all claims that may be brought by a third party in relation to any applications distributed by you which incorporate the Software, whether or not the terms of the License you have bought permit such distribution.
2. THE SOFTWARE IS NOT WARRANTED TO BE FAULT-TOLERANT, AND IS NOT INTENDED FOR THE DESIGN, CONSTRUCTION, MAINTENANCE, OPERATION, CONTROL, OR ANY OTHER USE IN CONNECTION WITH HIGH RISK SYSTEMS. WE SPECIFICALLY DISCLAIM ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR PURPOSE IN CONNECTION WITH HIGH RISK SYSTEMS. 'High Risk Systems' means systems in environments requiring fail-safe performance (such as nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems), in which the failure of the software could lead directly to death, personal injury, or severe physical or environmental damage. You agree that you will not use the software for any purpose in connection with High Risk Systems.
3. THIS SOFTWARE IS PROVIDED ``AS IS'', AND TO THE EXTENT PERMITTED BY THE APPLICABLE LAW, WE DISCLAIM ALL WARRANTIES WITH RESPECT TO THE SOFTWARE, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.
4. ALTHOUGH WE DO NOT WARRANT THAT THE SOFTWARE SUPPLIED HEREUNDER SHALL BE FREE FROM ALL KNOWN VIRUSES WE HAVE USED COMMERCIALY REASONABLE EFFORTS TO CHECK FOR THE MOST COMMONLY KNOWN VIRUSES. YOU ARE NEVERTHELESS SOLELY RESPONSIBLE FOR VIRUS SCANNING THE SOFTWARE.
5. WE SHALL NOT BE LIABLE TO YOU OR TO ANYONE ELSE FOR ANY LOSS OR DAMAGE WHATSOEVER OR HOWSOEVER CAUSED ARISING DIRECTLY OR INDIRECTLY IN CONNECTION WITH THIS LICENSE, THE SOFTWARE, ITS USE OR OTHERWISE, EXCEPT TO THE EXTENT THAT SUCH LIABILITY MAY NOT BE LAWFULLY EXCLUDED UNDER THE APPLICABLE LAW.
6. NOTWITHSTANDING THE GENERALITY OF THE ABOVE, IN NO EVENT WILL WE BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL LOSS OR DAMAGE WHICH MAY ARISE IN RESPECT OF THE SOFTWARE, ITS USE, OR IN RESPECT OF OTHER EQUIPMENT OR PROPERTY, OR FOR LOSS OF PROFIT, BUSINESS, REVENUE, GOODWILL OR ANTICIPATED SAVINGS, EVEN IF WE HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSS.
7. IF ANY EXCLUSION OR LIMITATION CONTAINED IN THIS LICENSE SHALL BE HELD TO BE INVALID FOR ANY REASON AND WE BECOME LIABLE FOR LOSS OR DAMAGE THAT MAY LAWFULLY BE LIMITED, SUCH LIABILITY SHALL BE LIMITED TO THE LICENSE FEE PAID BY YOU FOR THE SOFTWARE.
8. We do not exclude liability for death or personal injury to the extent only that it arises as a result of our negligence or that of our employees, agents or authorized representatives.
9. So far as any parts of this Software which were not created by us are concerned, THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ANY PERSON WHO HAS CONTRIBUTED TO OR IS THE OWNER OF ANY PART OF THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. This provision shall operate for the benefit of any and all such persons.

## **8. Miscellaneous**

1. You acknowledge that these terms supersede all prior agreements, and are complete and exclusive. No oral or written information given by us or on our behalf shall create a warranty or collateral contract, or in any way increase the scope of this warranty in any way, and you may not rely on any such advice.
2. If any provision in this Agreement shall be determined to be invalid, such provision shall be deemed omitted; the remainder of this Agreement shall stand.
3. This License shall be governed by the laws of Australia.



# CompleteFTP revision history

Version	Changes
Version 25.0.4 (4 June, 2025)	<ul style="list-style-type: none"><li>• Add support for aes128-gcm@openssh.com and aes256-gcm@openssh.com ciphers to SSH/SFTP.</li><li>• Remove MITM and SIZE caching variables in FTP client. This was causing weird permission problems in the gateway.</li></ul>
Version 25.0.3 (15 May, 2025)	<ul style="list-style-type: none"><li>• Revert SAML and Jint DLLs - since 25.0.1, the wrong builds were being used.</li></ul>
Version 25.0.2 (9 May, 2025)	<ul style="list-style-type: none"><li>• Fix TimeoutLogin issue with some SSH clients.</li></ul>
Version 25.0.1 (31 March, 2025)	<ul style="list-style-type: none"><li>• When generating certificate requests, save the private key with a PEM extension.</li><li>• Fix host header injection vulnerability.</li></ul>
Version 25.0.0 (12 March, 2025)	<ul style="list-style-type: none"><li>• Added FileManager3 (see <a href="#">here</a>).</li><li>• Fixed issue re importing TLS certificates.</li><li>• Fixed issue importing large audit files for reporting.</li><li>• SFTP performance optimizations.</li><li>• Fixed bug in regenerating Ed25519 server key - modified key was not saved.</li></ul>
Version 24.1.0 (30 December, 2024)	<ul style="list-style-type: none"><li>• Add support for encrypt-then-MAC HMAC algorithms in SFTP.</li><li>• Resolved a vulnerability where sensitive file system information could be exposed to unauthorized users.</li><li>• Fixed an issue allowing entire webapp directory structures to be downloaded in a ZIP archive.</li><li>• Patched a security flaw preventing malicious HTML files from being rendered within the user's browser.</li><li>• Optimized SFTP subsystem by reducing unnecessary thread creation, improving resource usage.</li><li>• Broadened TLS handshake support to handle larger handshake messages.</li><li>• Ensured that open file handles are correctly closed if a passive FTP connection times out.</li><li>• Improved authentication by directly passing User Principal Names (UPNs) to the LogonUser function.</li><li>• Introduced a new hook allowing custom validation logic for share passwords in FileManager2.</li><li>• Corrected a filtering logic error that affected IP-based access control.</li><li>• Fixed a bug in the Manager interface that prevented certain auditing configurations from being applied correctly.</li><li>• Resolved a conflict preventing remote host changes when FTPS was in use (Gateway).</li><li>• FileManager fixes re shared file expiry date.</li></ul>
Version 24.0.0 (24 July, 2024)	<ul style="list-style-type: none"><li>• Add support for LDAP authentication.</li><li>• Improved error handling and CPU utilization of the gateway. Both when downloading any file, and when the home folder is a gateway folder.</li><li>• Fix directory traversal vulnerability in the File Manager.</li><li>• Fix HTTP header vulnerability.</li><li>• In the CompleteFTP manager app, go to top of users and folder lists when they are displayed.</li><li>• Fix issue with installer failure if email notifications are saved in an old format. Happens if email notifications saved up until 21.1.0.</li><li>• Fix random password generation in manager.</li><li>• Fix IP filter bug re Allow All. If Allow All is enabled, and there's an allow always rule for the same IP address, the allow always can get overwritten in the IPFilter code.</li><li>• Fix issue where can't login with Domain user when setting the Default windows domain as 'mydomain' only.</li><li>• Fix issue where AD user's group-specific log-in-as user's access control property is ignored (each authenticator has a log-in-as user, but the AD authenticator allows this to be overridden by another log-in-as user that can be associated with an AD group).</li><li>• Fix issue where promises in auto-generated proxies for JSS RPC aren't compatible with async/await.</li></ul>
Version 23.1.2 (15 May, 2024)	<ul style="list-style-type: none"><li>• Fix issue with top level Amazon S3 Folder Accurate folder timestamps.</li><li>• Add ability to create new sites without automatically adding users.</li><li>• Support downloading of zip files bigger than 4 GB in the file manager folder download (support ZIP 64).</li><li>• Make randomly generated passwords the same length as the minimum password length defined in the password policy.</li><li>• Fix uneditable fields such as ClientPrivateKeyBytes in Gateway Connections.</li><li>• Update Azure assemblies and related assemblies.</li><li>• Update jQuery in file manager 2.</li><li>• Fix issue with manager password being supplied on upgrade even if it was not set to be saved.</li><li>• Fix an issue with SFTP request ids, which were overflowing with one particular client because they were ints instead of uints.</li><li>• Update French translations in the file manager.</li><li>• Disable SSL 3.0, which is redundant and insecure.</li><li>• Fix JSON.parse failure when reading JSON files containing top-level arrays.</li><li>• Fix JSS examples which weren't working properly.</li><li>• CompleteFTP now requires .NET framework 4.7.2 or higher.</li></ul>

Version 23.1.1 (7 February, 2024)	<ul style="list-style-type: none"> <li>• Optional second layer of encryption of credentials during HTTPS logins has been added. Enable with the <code>credentialEncryptionEnabled</code> variable in <code>customizations.jss</code>.</li> <li>• FileManager language settings now correctly respect customizations.</li> <li>• Don't set Windows home directory permissions when saving new domain users.</li> <li>• User guide enhanced with images on reporting pages for better clarity and user experience.</li> <li>• Fixed a bug where the FileManager's search would include VFS subfolders regardless of search match.</li> <li>• Corrected the "No usage recorder found" error after restarting the activated Standard Edition.</li> <li>• New customization option added to hide username and password fields on the login form via the <code>hideUserNamePassword</code> variable in <code>customizations.jss</code>.</li> <li>• Removed nonsensical 'Grand Total' values for 'Most recent' field.</li> </ul>
Version 23.1.0 (29 December, 2023)	<ul style="list-style-type: none"> <li>• Usage data storage and reporting (major new feature!).</li> <li>• Add support for ed25519 key types in SSH/SFTP.</li> <li>• Add connection info event property for use in triggers, that provides details of the protocol, ciphers etc.</li> <li>• Update Javascript editors to support new syntax.</li> <li>• Allow user to add code to validate changing folder in the FileManager.</li> <li>• Optionally allow showing context menu in FileManager by double-clicking.</li> <li>• Improve Portuguese translation in FileManager.</li> <li>• JSS API: make it easier to make configuration changes by adding <code>system.changeConfig(callback)</code>.</li> <li>• Fix bug reading OpenSSH-formatted private key when loading a TLS certificate.</li> <li>• Fix bug re accurate timestamps flag for S3 subfolders.</li> <li>• Fix process trigger issue where a session becomes invalid if the user logs out while trigger is running.</li> </ul>
Version 23.0.4 (13 September, 2023)	<ul style="list-style-type: none"> <li>• Add support for diffie-hellman-group-15-sha512 to diffie-hellman-group-18-sha512 in SSH key exchange.</li> <li>• Fix bug in Admin REST API: add group member fails.</li> <li>• Fix bug in IP filter code: if a Javascript filter throws an exception, the filter is disabled until the server is restarted!</li> <li>• Fix bug in process triggers that handle the LogIn on error event. If a client uses an invalid username, an exception is recorded in the log.</li> <li>• Fix User IP filtering issue with multiple sites in cases where there's a superfluous site left over from an Enterprise downgrade.</li> <li>• Fix TLS session memory leak.</li> </ul>
Version 23.0.3 (9 August, 2023)	<ul style="list-style-type: none"> <li>• Improved performance, reliability and resource usage of JSS IP filters, and fixed bugs (1) filters stopped working after modification, (2) returning filter results as Javascript objects resulted in errors.</li> <li>• FileManager: add Download button in toolbar.</li> <li>• FileManager: make days before expiry configurable.</li> <li>• FileManager: Show message box if user doesn't have permission when opening or downloading a file or directory.</li> <li>• Fix bug with credentialed network connections.</li> <li>• Fix bug with phantom connections showing as &lt;not logged in&gt; persisting in CompleteFTP manager connection list.</li> </ul>
Version 23.0.2 (30 June, 2023)	<ul style="list-style-type: none"> <li>• Update AWS assemblies to support IMDSv2.</li> <li>• Fix bug preventing all manual activation.</li> <li>• Fix bug in reading the new OpenSSH format for private keys that are not encrypted (i.e. no passphrase).</li> </ul>
Version 23.0.1 (12 June, 2023)	<ul style="list-style-type: none"> <li>• Fix for Microsoft error "The requested security protocol is not supported", related to TLS 1.3 not being supported prior to .NET 4.8. The solution is to upgrade to .NET 4.8, or apply this version. NOTE: the minimum .NET framework required is still .NET 4.6.2 - it's just that 4.8 fixes this error. 23.0.2 only requires .NET 4.6.2.</li> <li>• In SFTP, if ssh-rsa is disabled, do not allow clients to connect with public key authentication using ssh-rsa. Previously, clients could ignore the server-sig-algs sent from the server to indicate what was available.</li> </ul>
Version 23.0.0 (8 June, 2023)	<ul style="list-style-type: none"> <li>• Add support for TLS 1.3 for FTPS.</li> <li>• Updated Javascript interpreter to support most language features up to and including ECMAScript 2023.</li> <li>• Add user web-apps. Previously, users have not been able to add their own web-apps.</li> <li>• Add 'auto-create' checkbox to Active Domain authenticator dialog.</li> <li>• Improve activation process.</li> <li>• Allow a single email address to be used for multiple accounts.</li> <li>• Add support for UTF-8 paths and underscores for the REST API.</li> <li>• Fix bug in Download as ZIP - a customer had reported not all files were downloaded.</li> <li>• Installer now allows import of sdf files (the old config format) and db files (the new config format).</li> <li>• Requires .NET 4.6.2 or higher.</li> </ul>
Version 22.2.3 (4 April, 2023)	<ul style="list-style-type: none"> <li>• Tighten security for built-in users.</li> <li>• Fixed bug in recently added switch-user functionality in HTTP/HTTPS.</li> </ul>
Version 22.2.2 (21 February, 2023)	<ul style="list-style-type: none"> <li>• Fix SFTP issue where Windows paths were exposed in error messages returned to the client.</li> <li>• Fix bug where sites filter didn't work properly.</li> <li>• Fix SAML problem where importing metadata containing type attributes failed.</li> <li>• "Download as ZIP" in file manager now uses UTF-8.</li> </ul>
Version 22.2.1 (12 January, 2023)	<ul style="list-style-type: none"> <li>• Enable JSS scripts to switch from one authenticated account to another.</li> <li>• Fix bug where FTPS data channels did not copy allowed algorithms from control channel.</li> <li>• Fix bug where Audit configuration form could not be opened.</li> <li>• Fix bug where installer would report an error while copying email notifications and process triggers during installation.</li> </ul>

Version 22.2.0 (25 November, 2022)	<ul style="list-style-type: none"> <li>• Support for password expiration and password reset for non-Windows users. JSS authenticators can now define a setPassword function that will be called when the user requests a password change.</li> <li>• Strengthen default password policy for setting of new passwords, and ensure generated passwords comply.</li> <li>• Document use of two-factor authentication (2FA) for Microsoft Authenticator and others.</li> <li>• Add support for AWS instance name based Activation Keys.</li> <li>• Add the ability to get accurate timestamps from Amazon S3 folders. This must be enabled, as it can cause performance issues as timestamps must be fetched individually.</li> <li>• Add option to not update permissions in the Windows AD authenticator. Normally, CompleteFTP adds a permission giving the logged-in user full control over their home directory. This options prevents the permission being added.</li> <li>• Add extra features to the manager user list in the Users panel, including extra columns, filtering, search, grouping.</li> <li>• SMTP configuration settings moved to a more central location in Site/Settings of the manager.</li> <li>• Enable entry of language translations in the File Manager web application, so they can be sent to EnterpriseDT and included in the distribution.</li> <li>• Ensure SHA256 is used in FTPS peer signing if supported by the client.</li> <li>• Fix bug where transfers to/from network folders with custom credentials can be cut short.</li> <li>• Fix autoban duration to be always in seconds.</li> <li>• Fix license import bug in command-line license tool.</li> <li>• Fix bug in Windows AD group sorting and removal (after sorting, removal removed a non-selected group).</li> <li>• Fix bug in sorting user data on the CompleteFTP manager's user panel.</li> <li>• Fix bug in real-time logging (mangled messages).</li> <li>• Fix bug where file sharing fails when home dir is set to root for a user.</li> </ul>
Version 22.1.3 (15 August, 2022)	<ul style="list-style-type: none"> <li>• Fixed bug causing HTTP login page to fail to render (only when public HTTP access was disabled).</li> </ul>
Version 22.1.2 (5 August, 2022)	<ul style="list-style-type: none"> <li>• Make all pages in Login, Logout and Account localizable. Japanese translation of these pages.</li> <li>• Allow configuring of server-sig-algs string via SSH/SFTP server host algorithms. This allows the server to send a SSH_MSG_EXT_INFO message to the client to indicate what client public key algorithms will be accepted.</li> <li>• Upgrade-proof user customization of HTML pages.</li> <li>• Fix key re-exchange bug introduced in previous version. If a client supports server-sig-algs the server sends a SSH_MSG_EXT_INFO message on connection. In the previous version, it also sent SSH_MSG_EXT_INFO during key re-exchange, which is incorrect.</li> </ul>
Version 22.1.1 (5 July, 2022)	<ul style="list-style-type: none"> <li>• Recursive search added to web-based File Manager.</li> <li>• Fixed infinite loop in gateway that occurs if the remote file is truncated during transfer.</li> <li>• Fixed security vulnerability in HTTPS.</li> <li>• Fix bug in rsa-sha2-256 and rsa-sha2-512 host key algorithms. Implement rsa-sha2-256 and rsa-sha2-512 for user authentication.</li> </ul>
Version 22.1.0 (11 May, 2022)	<ul style="list-style-type: none"> <li>• Add support for ssh-256-rsa and ssh-512-rsa in SFTP.</li> <li>• Support new PuTTY private key format (v3) for SSH private keys.</li> <li>• Allow S3 folders to use role based security instead of secret keys.</li> <li>• Add AdminAPI: REST API for CompleteFTP administration.</li> <li>• Can now specify key size for certificate signing requests (CSRs).</li> <li>• The CompleteFTP Manager now (mostly) supports other languages (Japanese to begin with).</li> <li>• Fix HTTP PUT failure when home-root is enabled.</li> <li>• Fix Amazon S3 folder issue, where only one S3 bucket could be accessed by a user in one session.</li> <li>• Fix timeout issue in HTTP uploads.</li> <li>• File-manager: Change appearance of breadcrumb to a simple slash-separated path.</li> <li>• File-manager: added refresh button to update listing.</li> <li>• User Guide improvements and updates.</li> </ul>
Version 22.0.2 (22 February, 2022)	<ul style="list-style-type: none"> <li>• Fix certificate loading issue for private keys in PEM and PVK formats.</li> <li>• Unknown FTP commands ignored after logging in.</li> </ul>
Version 22.0.1 (10 February, 2022)	<ul style="list-style-type: none"> <li>• Add random read access to AzureFileAdapter.</li> <li>• Rename Enterprise Edition to Enterprise MFT (Managed File Transfer Edition).</li> <li>• Fix SAML bug where HTML appended to SAML metadata.</li> <li>• Fix "Save Diagnostics" in the manager (broken in 22.0).</li> <li>• Fix license utility (broken in 22.0).</li> <li>• Allow CLNT and CSID without being logged in.</li> <li>• Check for 64 bit Windows in installer.</li> </ul>

Version 22.0.0 (24 January, 2022)	<ul style="list-style-type: none"> <li>• Now a 64 bit Windows application.</li> <li>• <b>IMPORTANT!</b> Firewall rules will need to be updated as CompleteFTP executables are now under C:\Program Files rather than C:\Program Files (x86).</li> <li>• Replace SQL Server Compact with SQLite for configuration database.</li> <li>• Improve installer for upgrades (no longer forces uninstall of previous version).</li> <li>• Improve iOS support in the file-manager (use double-press instead of long-press).</li> <li>• More meaningful error message when sharing from Azure and Amazon folders.</li> <li>• Passive FTP: accept data connections from white-listed IP addresses. Allows CompleteFTP to operate behind active/active firewalls requiring use of SNAT, including the Azure firewall.</li> <li>• Remove simple permission model - all editions now use the advanced permission model.</li> <li>• Improve handling of unknown commands in FTP.</li> <li>• Disabled elliptical curves in TLS that are non-compliant with PCI DSS requirements.</li> <li>• Add uname to SSH commands.</li> <li>• Support regular expressions in filename filters (via a 'regex:' prefix).</li> <li>• Support for new OpenSSH private key format.</li> <li>• Add support for reading ECDSA private keys in PuTTY format.</li> <li>• Restore original default autoban setting of 60 seconds for sample period.</li> <li>• Fix issue opening private key files (PVK) on certain machines (usually as part of loading a CA certificate).</li> <li>• Support for opening mp4 files from file-manager.</li> </ul>
Version 21.1.0 (6 September, 2021)	<ul style="list-style-type: none"> <li>• Improve autobanning to prevent repeated failed connections. Increase default autoban counting period to 300 seconds for new installations.</li> <li>• Add copy menu item to user list.</li> <li>• Lots of minor improvements to the CompleteFTP manager GUI.</li> <li>• Fix issue of configuration database bloating if IP address of server is regularly changing.</li> <li>• Add support for importing ECDSA keypairs in new OpenSSH format.</li> <li>• Add support for user@domain syntax for Windows users when logging in.</li> <li>• Fix append for Azure files.</li> <li>• Add test Unix command to the command-line shell.</li> <li>• Enable Manager to be localisable by users. Localise side-menu to Japanese. Add Japanese translation for the web file manager.</li> <li>• Add support for long path names. Requires minimum Windows 10 version 1607, or Windows Server 2016, and LongPathsEnabled to be set in the registry. This doesn't allow file-names &gt; 255 chars as Windows doesn't support this - it is only paths.</li> <li>• Hide keys in Azure and S3 configurations.</li> <li>• Add SELECT support to cftpconfig.</li> <li>• Log memory consumption on user login/logout.</li> <li>• Add support for groups and public keys in web admin.</li> </ul>
Version 21.0.1 (9 June, 2021)	<ul style="list-style-type: none"> <li>• Fix decryption issue for Encryption at Rest.</li> <li>• Fix permission error with JSS authenticators that access the virtual file-system.</li> <li>• Fix User Guide search.</li> </ul>
Version 21.0.0 (19 May, 2021)	<ul style="list-style-type: none"> <li>• Add Two Factor Authentication (2FA) to HTTPS (Enterprise MFT).</li> <li>• Add support for CompleteAgent automatic downloads (Enterprise MFT).</li> <li>• Strengthen security in JSS scripting (used in web file-manager).</li> <li>• Add %protocol% macro for email notifications.</li> <li>• Script chaining in JSS web apps. Combined HTML/JSS files (.htmljss).</li> <li>• Implement SITE HELP command.</li> <li>• Use SHA256 to sign certificate requests.</li> <li>• Allow SIZE to be sent during FTP transfers.</li> <li>• Implement exporting of ECDSA public host keys.</li> <li>• Web-based account management added.</li> <li>• Fix issue where the PASV IP address could not be entered.</li> <li>• Strengthen Encryption At Rest.</li> <li>• Fix bug in event scheduler where monthly events get stuck in a loop.</li> <li>• Fix race condition when creating temporary scriptN.bat files for process triggers.</li> <li>• Fix bug with ReportDecryptedSize - it was causing HTTP downloads to fail because the wrong size was being reported and used for content length.</li> </ul>
Version 13.1.3 (1 February, 2021)	<ul style="list-style-type: none"> <li>• Fix issue of manager hanging on initial connection (introduced in 13.1.2).</li> </ul>
Version 13.1.2 (27 January, 2021)	<ul style="list-style-type: none"> <li>• Add 'user matching' option for the SAML authenticator, which causes CompleteFTP to use the settings of the CompleteFTP with the same username as the user logging in via SAML.</li> <li>• Fix ConsumeWindowSpace issue with SunOS sftp client.</li> <li>• Don't return an error in SFTP when a client attempts to cd to the parent of root (/).</li> <li>• Fix FTPS bug - VMware backup was failing with a decode error.</li> </ul>
Version 13.1.1 (14 December, 2020)	<ul style="list-style-type: none"> <li>• Upgrade SAML library to help resolve some customer issues. Improve signature verification.</li> <li>• Add "Report Decrypted Size" as a user setting and remove the site setting.</li> <li>• Improve Active Directory authenticator by better checking that a user is in certain AD groups (now takes domain name into account).</li> <li>• Render HTML files opened from the file-manager as plain text (to prevent XSS attacks).</li> <li>• Add Swedish to web-based file-manager.</li> <li>• Close a vulnerability to oversize messages prior to authentication.</li> <li>• Produce an informative error message when incoming passive FTP connections fail.</li> </ul>

Version 13.1.0 (18 November, 2020)	<ul style="list-style-type: none"> <li>• Add IPv6 support for all protocols. For FTP/FTPS, this includes EPRT/EPSV support.</li> <li>• Add ECDSA support. This includes ECDSA key exchange algorithms, ECDSA host keys and ECDSA user keys.</li> <li>• If encryption at rest is enabled, the size of encrypted files is reported incorrectly as it will include the file header containing encryption data. The "Listings show decrypted size"™ setting in the Encryption At Rest section of File System Settings can be checked to obtain the true file sizes. Note that this has a performance cost for directories with many files, and so it is not enabled by default.</li> <li>• Simplified Windows/AD group configuration so that if the default Windows user is used as the template user, it will work with normal paths set for the home directory (previously it did not, and only supported Windows special folders).</li> <li>• Fix HTML error template Javascript injection attack vulnerability.</li> <li>• Add support for STRU F in FTP/FTPS (as Azure seems to require it).</li> <li>• Provide a way for migrated free edition installations to be reactivated.</li> </ul>
Version 13.0.1 (7 September, 2020)	<ul style="list-style-type: none"> <li>• Improve memory management in SFTP.</li> <li>• Fix issue in SFTP where occasionally a file would be locked.</li> </ul>
Version 13.0.0 (24 August, 2020)	<ul style="list-style-type: none"> <li>• Announcing the first release of CompleteAgent - a client-side Windows service and command-line tool that is designed for performing robust, automatic uploads to CompleteFTP. It requires CompleteFTP Enterprise Edition.</li> <li>• Added setting that allows overwriting of files on rename.</li> <li>• Added advanced Active Directory group integration with CompleteFTP groups.</li> <li>• Improve authentication speed for Automatic Windows Users.</li> <li>• Changing Windows permissions of existing directories can be very slow (and sometimes times out), so now only newly created Windows directories will have Windows permissions set.</li> <li>• The new file-manager now allows a single folder to be downloaded as a ZIP file.</li> <li>• Added support for extended master secret in TLS 1.2.</li> <li>• In SFTP, queue FXP_REMOVE and FXP_RENAME requests to prevent delete and rename failures after upload (because the server is still processing write requests).</li> <li>• Remove need to re-enter password when upgrading or reinstalling from versions after 13.0.</li> <li>• Major revision of user guide. Now more modern and mobile-friendly. Numerous fixes.</li> <li>• Enable changing passwords of Windows users (local and AD) via HTTP.</li> <li>• Bug fixes in HTTP PUT. The the HTTP reply (200) was sent back to the client too early, and in some cases the client was immediately closing the connection rather than completing the PUT.</li> <li>• Ensure Powershell scripts work with Powershell 5.x (so if Powershell 2 is disabled, scripts still run).</li> </ul>
Version 12.1.5 (12 February, 2020)	<ul style="list-style-type: none"> <li>• Add support public key authentication in the gateway authenticator.</li> <li>• Improve the internal web pages.</li> <li>• Fix bug maximizing the manager on a secondary monitor.</li> <li>• Fix bug when Automatic Windows Users log in via public key authentication - password authentication was not used and so even though the login seemed to succeed, operations would later fail.</li> <li>• Auditing now reports transfer errors - previously a partial transfer was not recorded as a failure.</li> <li>• Fix issue re certain external RSA keys failing to be loaded.</li> <li>• Fix file sharing for active directory users.</li> <li>• Fix memory leak when custom command called from process trigger.</li> <li>• Allow adding cluster nodes with same SID.</li> </ul>
Version 12.1.4 (7 October, 2019)	<ul style="list-style-type: none"> <li>• Fix append issue in FTP for certain VM configurations.</li> <li>• Security fix for manager password saving.</li> <li>• <b>Security fix for SSH exec</b> - now restricted to members of 'admins' group.</li> <li>• Security fix for new filemanager node permissions.</li> <li>• JQuery update.</li> <li>• Optimise when user-profile is loaded.</li> <li>• Fix installer issue with non-canonical access control lists.</li> <li>• Fix for importing latest Filezilla server config.</li> </ul>
Version 12.1.3 (9 September, 2019)	<ul style="list-style-type: none"> <li>• Enable rename command to be called after PASV in FTP.</li> <li>• Update script engine.</li> <li>• Ensure manager supports TLS 1.2 for HTTP requests (such as checking latest version).</li> <li>• Fix manager bug with expand arrow for Quotas &amp; Limits category hidden in some skins.</li> <li>• <b>Installer security fix.</b></li> <li>• Remove direct dependence on CryptoAPI for TLS 1.0 and TLS 1.1.</li> <li>• Use ephemeral RSA keys to prevent buildup of files in MachineKeys folder.</li> </ul>
Version 12.1.2 (16 August, 2019)	<ul style="list-style-type: none"> <li>• Add AES Galois Counter Mode (GCM) cipher suites to TLS.</li> <li>• Change email notifications to use TLS 1.2 if available.</li> <li>• Fix ConsumeWindowSpace issue.</li> <li>• Change the URL the manager contacts to find the latest version to HTTPS.</li> <li>• Improve Automatic Windows Users for authenticating groups in other domains.</li> <li>• Fix Gateway issue: changing permissions causes an error for FTP/FTPS.</li> <li>• Fix signature issue verifying SAML responses.</li> </ul>

Version 12.1.1 (17 June, 2019)	<ul style="list-style-type: none"> <li>• Add diffie-hellman-group14-sha256 to SSH key exchange.</li> <li>• Audit and diagnostic log files now support non-ASCII characters.</li> <li>• Support modifying permissions from SFTP client via gateway folder to remote SFTP host.</li> <li>• Fixed a bug in SFTP directory listings (permissions and owner were sometimes incorrect).</li> <li>• Installer now prompts to remove previous installation to prevent issues with non-standard install locations.</li> <li>• Fix poor handling of permission denied error in new file-manager.</li> <li>• Web file-manager can now download multiple files.</li> <li>• Fixed SCP bug where could not upload a file when the filename contains Unicode characters.</li> <li>• Fixed SCP bug where could not upload a file after failing to upload a file blocked by a filename filter.</li> <li>• Fixed bug where generating RSA keys of keylengths not divisible by 8 caused an error. Now keylengths must be a multiple of 8.</li> <li>• Fixed annoying bug in manager GUI where changes to SFTP advanced properties did not trigger the enabling of the Apply Changes button.</li> <li>• Fixed error that occurs when clicking the CompleteBox icon in the system tray.</li> <li>• Minor documentation fixes.</li> </ul>
Version 12.1.0 (31 March, 2019)	<ul style="list-style-type: none"> <li>• Password-protected sharing in CompleteBox.</li> <li>• Import PEM format certificate/private keys for server certificate.</li> <li>• Support for downloading a single dynamically created file that contains a recursive directory listing.</li> <li>• Add CSID and CLNT FTP commands.</li> <li>• Fix bug where RSA keys were generated with 2,047 bits instead of 2,048 bits.</li> <li>• Enable PASV and STOR/RETR FTP commands to be separated by other commands to support clients that do this.</li> <li>• Support SAML group claim and a custom attribute called 'group' to allow CompleteFTP group membership to be specified via IDP.</li> <li>• Numerous bug fixes in CompleteBox.</li> <li>• Improved memory usage when downloading large files via SFTP.</li> <li>• Fix for ncftpput clients hanging.</li> <li>• Fix for bug introduced in 12.0 where authentication would fail if the Domain Users group was specified as allowed group for automatic Windows users (AWU).</li> <li>• Support for better integration with <a href="#">SyncbackPro</a>.</li> </ul>
Version 12.0 (5 December, 2018)	<ul style="list-style-type: none"> <li>• New Web file-manager which includes file-sharing.</li> <li>• New WebAdmin application for basic user administration.</li> <li>• File system adapters for Amazon S3 and Azure</li> <li>• TLS/SSL certificate expiring event.</li> <li>• Many enhancements to the JSS (Javascript Serverside) Config API.</li> <li>• Add authenticator panel to users panel.</li> <li>• Ability to set a per-user home-root.</li> <li>• Enter path of Windows folders via text-editor.</li> <li>• Allow user to return a log-in-as user-name from their DB authenticator query.</li> <li>• Automatic Windows/AD authentication: Allow log-in-as user to be specified per AD group.</li> <li>• Expose client certificate to custom .NET authentication extensions.</li> <li>• Custom authenticators: make connection information accessible to authenticators.</li> <li>• Allow administrators to change folder type in the manager.</li> <li>• Allow administrators to add a date column in the real-time log viewer. .</li> <li>• Fix bug where HomeDirectory was ignored in advanced custom authentication extensions that use PK</li> <li>• Fix bug where 'Ignore filters' permission did not work properly for uploading of files via SFTP.</li> <li>• Fix CompleteBox errors after sleep/resume.</li> </ul>
Version 11.0.6 (19 September, 2018)	<ul style="list-style-type: none"> <li>• Added Elliptic Curve key exchange algorithms for TLS 1.2.</li> </ul>
Version 11.0.5 (25 July, 2018)	<ul style="list-style-type: none"> <li>• Fixed memory leak introduced in 11.0.3.</li> </ul>
Version 11.0.4 (11 July, 2018)	<ul style="list-style-type: none"> <li>• Fixed bug where JSS mail attachments file streams were not closed, meaning the files were locked after sending.</li> <li>• Fixed ConsumeWindowSpace error that occurred with an SFTP client reading a file.</li> </ul>
Version 11.0.3 (5 June, 2018)	<ul style="list-style-type: none"> <li>• Significant memory optimisation in certain situations for SFTP.</li> </ul>
Version 11.0.2 (4 May, 2018)	<ul style="list-style-type: none"> <li>• Allow substitution of environment variables, such as %LOCALAPPDATA%, in network/macro folders.</li> <li>• Fix CompleteFTP manager bug where the error "Configuration changes have been made by a third party" prevents connecting</li> <li>• Fix rare certificate issue where a microsoft\crypto\RSA entry is created by what ends up as an unresolved SID.</li> <li>• Fix logging bug where some log tags are blank, causing issues in real-time logging.</li> </ul>
Version 11.0.1 (22 March, 2018)	<ul style="list-style-type: none"> <li>• Fix install failure on non-English installs (IdentityNotMappedException).</li> </ul>

Version 11.0.0 (9 March, 2018)	<ul style="list-style-type: none"> <li>• Major new feature. Encryption at rest now supported.</li> <li>• Major new feature. CompleteFTP is now a SAML Service Provider.</li> <li>• Logout page now customizable.</li> <li>• Login page now responsive.</li> <li>• Various JavaScript Server-Side (JSS) enhancements, including the JSS Config API (in beta) that allows the user to make CompleteFTP configuration changes via JSS.</li> <li>• Enhancements to usermod command (Added enable flag and full-name argument).</li> <li>• Allow text selection in real-time logging view.</li> <li>• Fix issue where for certain users on Windows Server 2016 certificate generation was failing during installation with "Can't acquire cryptographic context" error message.</li> <li>• Simplify installer.</li> <li>• Minor CompleteBox fixes.</li> <li>• Fix ClientIP macro.</li> <li>• Fix bug where the manager was crashing when attempting to delete the server certificate of a non-Default site.</li> <li>• Fix bug where an error message was shown when trying to delete a secondary server from a cluster.</li> <li>• Failed logins now shown in audit log.</li> <li>• <u>Minimum of .NET 4 required</u> (earlier versions ran on .NET 2.0). Later versions such as 4.6 or 4.7 are also fine. Enterprise .NET extension DLLs must be recompiled with the .NET 4 framework. It can be downloaded from Microsoft <a href="#">here</a>.</li> <li>• Tightened security of configuration data. <b>WARNING!</b> - users who have changed the 'log on' user of the CompleteFTP service from the default (i.e. local SYSTEM user) may need to manually configure Windows file permissions after installation (see <a href="#">here</a>).</li> </ul>
Version 10.2.1 (20 October, 2017)	<ul style="list-style-type: none"> <li>• Add support for HTTP 0.9 (used by some monitoring clients).</li> <li>• Added ability to enable and disable users in administrator scripting.</li> <li>• Fixed ClientIP macro in e-mail notification and process triggers</li> <li>• Fix permissions issue with multipart HTTP requests.</li> <li>• Configure scripting engine to work with .NET 2.0 and .NET 4.x.</li> <li>• Fixed JSS API documentation errors.</li> </ul>
Version 10.2.0 (21 August, 2017)	<ul style="list-style-type: none"> <li>• Add server initiated key re-exchange for SSH.</li> <li>• Allow PASV to immediately follow PORT and vice versa for FTP/FTPS (first command is ignored).</li> <li>• Add CSRF token to web file manager to prevent CSRF attacks.</li> <li>• Added support for admin site authenticators.</li> <li>• Fix SSH bug when reading client version string.</li> <li>• Upgrade the binary that runs process trigger scripts (edtfp.exe).</li> <li>• Upgrade CompleteBox.</li> </ul>
Version 10.1.1 (12 May, 2017)	<ul style="list-style-type: none"> <li>• Increase timeout when adding new secondary server to cluster, and make call asynchronous. Prevents timeouts for servers that take a minute or two to restart.</li> <li>• Fix bug where login failures caused by unrecognised user-names would only cause process triggers to be called if an alternative authentication method (such as AD authentication) was enabled.</li> <li>• Fix bug limiting the number of gateways that could be configured to three.</li> <li>• Fix gateway timestamp bug - gateway folders now allow timestamps of files to be set.</li> </ul>
Version 10.1.0 (24 April, 2017)	<ul style="list-style-type: none"> <li>• New feature - added ability to add custom IP filter extensions (.NET or Javascript).</li> <li>• External database users can now use RSA &amp; DSA public keys for authentication.</li> <li>• Added extensive step-by-step user tutorials.</li> <li>• Content-Type was missing in HTTP responses, which was a problem if X-Content-Type-Options "nosniff" is set, as the browser can no longer detect the content type.</li> <li>• Prevent the first "none" SSH auth method often sent by clients from triggering a failed login event.</li> <li>• Fix HTTP/HTTPS upload bug where multipart uploads ended up with an extra char.</li> <li>• Fixed bug where 'create a home folder automatically' when adding a new windows user can't be unenabled.</li> <li>• Fixed permissions bug in SCP transfers - impersonation wasn't working correctly initially..</li> <li>• Fixed issue with confusing log file entries.</li> <li>• Fixed HTTP login error that occurred when 'Use browser-based logins' and 'User's home folder appears as root' are both enabled.</li> <li>• Fixed numerous minor manager GUI issues.</li> </ul>
Version 10.0 (21 December, 2016)	<ul style="list-style-type: none"> <li>• Major new feature - add ability to write process triggers in Javascript (JSS).</li> <li>• Modernize manager GUI with side menu.</li> <li>• User and site filtering in events (email and process triggers).</li> <li>• Add new system events (banned or autobanned IP address, server startup, clustering failure, certificate expiry).</li> <li>• Added TCP port usage view.</li> <li>• Numerous manager GUI fixes and minor enhancements.</li> <li>• Add ephemeral Diffie-Hellman (DHE) ciphers to FTPS, so perfect forward secrecy now supported.</li> <li>• Add ability to set additional HTTP headers so various security headers can be used.</li> <li>• Memory optimizations.</li> <li>• Disable SSH EXEC command in Standard Edition (and only enable in higher editions if SSH terminals are enabled).</li> <li>• Fix starting CFPT service from the manager (asks for elevated permissions).</li> <li>• Fix client certificate support in TLS 1.2 (i.e. FTPS).</li> <li>• Fix bug in FTP listing of absolute paths with a wildcard.</li> <li>• Fix bug where couldn't set the external passive IP if there was a 0 in the IP address.</li> <li>• Fixed bug where the manager gets into a loop if the real-time log download continually fails.</li> </ul>



Version 9.1.3 (10 October, 2016)	<ul style="list-style-type: none"> <li>Added ability to import Filezilla server users, folders and groups.</li> <li>Fixed bug in gateway preventing multiple downloads and fixed issue with a SFTP &lt;-&gt; FTP failure.</li> <li>Fixed bug in FTPS experienced when thousands of small files are transferred.</li> <li>Fixed "invalid message number" issue experienced by WinSCP when transferring via SFTP.</li> </ul>
Version 9.1.2 (12 September 2016)	<ul style="list-style-type: none"> <li>Fixed SFTP bug that can cause server cpu to spike when certain clients such as perl sftp are used.</li> <li>Fixed real-time logging bug that causes a loop in the manager when FTP is used as the connection protocol and real-time logging is enabled. Workaround is to use the default SFTP protocol to connect manager to the server (on port 14983).</li> </ul>
Version 9.1.1 (30 August 2016)	<ul style="list-style-type: none"> <li>Fixed cluster bug stopping servers from being added and also preventing synchronization (introduced with automatic deactivation in 9.1.0).</li> <li>Fix for "Unable to load DLL 'dwmapi.dll'" error in CompleteFTP manager on Windows 2003 Server.</li> <li>Fixed bug where min/max/close wouldn't be visible in Windows 2008.</li> <li>Fixed bug preventing macros from working when used in names of virtual file-system folders.</li> <li>Fixed bugs in new certificate chain importation feature - some certificates failed to load.</li> <li>Fixed bug on machines with desktop logins disabled that allowed users to access folder that they did not have Windows permissions to access.</li> <li>Prevented deactivation of servers if current activation key is invalid (esp if config moved to new machine).</li> <li>Added error-message if user tries to deactivate on machine with invalid activation key.</li> </ul>
Version 9.1.0 (5 August 2016, see also 9.0.1 changes)	<ul style="list-style-type: none"> <li>Release of Free Edition supporting FTP and FTPS.</li> <li>Support for automatic activation and deactivation added.</li> <li>Added support for Tags in authentication extensions.</li> <li>Various Javascript Server-Side (JSS) fixes and improvements: <ul style="list-style-type: none"> <li>Fixed bug preventing multiple JSON DBs</li> <li>Added tags property to system.user.</li> <li>Added adapterAbsolutePath to File objects.</li> <li>Prevent circular includes.</li> <li>Added support for x-www-form-urlencoded (i.e. standard form) encoding in HTTP POSTS.</li> </ul> </li> </ul>
Version 9.0.1 (19 July 2016, not made generally available)	<ul style="list-style-type: none"> <li>Support for loading of intermediate SSL certificates, and sending the entire certificate chain to clients.</li> <li>TLS 1.2 fix where signalling ciphers were being selected as genuine ciphers. Error in debug log was "Offset and length were out of bounds for the array".</li> <li>Ensure FTP welcome message and SSH user auth message is converted to ISO-8859-1 (as some Unix clients can't cope with UTF).</li> <li>Improve efficiency of SFTP threading when many small files are transferred.</li> <li>Fix FTP directory listing so wildcards are handled.</li> <li>Improve memory utilisation with the multiprotocol gateway.</li> <li>Ensure clients don't complete a transfer until the secondary transfer is complete in the multiprotocol gateway.</li> </ul>
Version 9.0.0 (23 March 2016)	<ul style="list-style-type: none"> <li>Real-time log viewing in CompleteFTP Manager.</li> <li>Script-based extensions. Authentication, file-system and custom command extensions can now be written in Javascript (JSS) and entered directly into CompleteFTP Manager. Enterprise MFT only.</li> <li>Read-only sites. Sites can be switched to read-only using a single checkbox.</li> <li>Added FTP STAT command.</li> <li>Added passwd command to SSH shell so internal users can change their password.</li> <li>Groups can now be set from authentication extensions.</li> <li>New flat look-and-feel for CompleteFTP Manager.</li> <li>Many improvements to JSS resulting from the development of our customer portal, <a href="#">editConnect</a>, which is hosted on CompleteFTP. Most notable is the introduction of JSON-DB - an indexed database where objects are stored in individual JSON files.</li> <li>Changed gateway so that FTPS can specify max &amp; min SSL connections.</li> <li>Upgraded log4net DLL to 1.2.15. Resolves syslog bug in earlier log4net DLL.</li> <li>Fixed SCP bug where filenames containing spaces could not be downloaded.</li> <li>Fixed WinSCP issue where a directory's name was appearing in its content listing.</li> <li>Fixed anonymous HTTP bug.</li> <li>Fixed HTTP session time-out bug. Session time-outs were measure from time of login rather than time of last access.</li> <li>Over 60 minor bug fixes and enhancements in CompleteFTP Manager resulting from a comprehensive usability review.</li> <li>Made a start at improving CompleteFTP Manager's accessibility by making more user-interface elements compatible with Windows Narrator. It's still early days, so we're looking for feedback on this.</li> </ul>
Version 8.5.1 (15 January 2016)	<ul style="list-style-type: none"> <li>Fix bug affecting database authentication logins (login failure).</li> </ul>
Version 8.5.0 (30 November 2015)	<ul style="list-style-type: none"> <li>Password hashing now uses PBKDF2-SHA256.</li> <li>New command-line commands for administering permissions (chmod, chown).</li> <li>Support for TLS_FALLBACK_SCSV in TLS.</li> <li>Fix bug in SSL3 renegotiation.</li> <li>Change anonymous FTP behaviour so that if disabled there is still a password prompt.</li> <li>Fix issue where login failures did not trigger the login event when set.</li> <li>Remove RC4 support from default ciphers (as no longer regarded as secure).</li> <li>Fix issue where downloading files with special chars in their name from HTTP works on Firefox but not with Internet Explorer.</li> <li>Fix bug where all connections in the manager application are shown as from one user.</li> </ul>
Version 8.4.0 (9 October 2015)	<ul style="list-style-type: none"> <li>Now supports <a href="#">TLS 1.2</a>.</li> <li>Now supports <a href="#">TLS secure renegotiation</a>.</li> <li>Fix bug where downloading in SFTP via Gateway failed.</li> </ul>



Version 8.3.3 (7 September 2015)	<ul style="list-style-type: none"> <li>• More efficient memory utilization, and now uses .NET 4.x framework if it is installed.</li> <li>• Bugfix for scheduled events, where some events were not firing due to a race condition.</li> <li>• Bugfix for downloading from a zip folder.</li> <li>• Fixed OpenSSH 6.7 (and higher) issue using HMAC SHA2 with SFTP.</li> <li>• Fixed incorrect IP address in logging.</li> </ul>
Version 8.3.2 (22 July 2015)	<ul style="list-style-type: none"> <li>• Now reuses PASV port for the same client, meaning PASV port range is used more efficiently.</li> <li>• Fix to FTPS implicit mode.</li> <li>• Better throttling of server messages being sent to clients, reducing memory consumption.</li> <li>• Fix to SSH key re-exchange when initiated by clients.</li> <li>• New version of CompleteBox (1.1.1) included.</li> </ul>
Version 8.3.1 (17 June 2015)	<ul style="list-style-type: none"> <li>• Replace Microsoft HMAC in SSH/SFTP as it was causing problems for a user.</li> <li>• Fix a directory traversal security vulnerability.</li> <li>• Minor fixes to logging.</li> </ul>
Version 8.3.0 (8 May 2015)	<ul style="list-style-type: none"> <li>• Added scheduled events. This means events can be set up to fire at a predefined schedule, such as once per hour or at midnight every weekday.</li> <li>• Selectable ADO.NET providers for database authenticator.</li> <li>• Add SHA1 to database authentication option, so now MD5 and SHA1 are supported.</li> <li>• A user's Active Directory home directory can now be set as a macro variable.</li> <li>• Set service recovery options to automatically restart for new installations.</li> <li>• If the anonymous user is disabled in FTP, give an error rather than prompting for password.</li> <li>• Fix issue with certificate signing requests that meant Verisign/Symantec would not accept them.</li> <li>• Make IPAddresses configuration field larger so more network interfaces can be supported.</li> <li>• Gateway authenticator fix - for multiple authenticators, the connection details of the first remote server were being re-used.</li> </ul>
Version 8.2.0 (18 March 2015)	<ul style="list-style-type: none"> <li>• Per-user IP filtering.</li> <li>• Add SHA-1 password hashing to Authenticator. Now extensions can use SHA-1 hashed passwords.</li> <li>• Revert to using SHA-1 in certificate signing requests (CSR) because Verisign doesn't accept SHA-256 requests!</li> <li>• In CompleteFTP manager, the FTP/FTPS Server Certificate property is now also available under Advanced HTTP/HTTPS Settings.</li> <li>• There is now an Extensions directory for loading extension assemblies from.</li> <li>• Allow filenames which have '.' as part of the name(three dots).</li> <li>• Many new Javascript Server-Side (JSS) features.</li> <li>• New maximum connections per user setting.</li> <li>• Fixed issue starting CompleteFTP manager on .NET 4.x-only installations on machines with non-English Windows. Manager would terminate with a FileNotFoundException referencing CompleteFTPManager.resources.</li> <li>• Fixed issues with initial blank page appearing after logging in via HTTP/S.</li> <li>• Fixed "Couldn't create unsigned certificate" exception that occurred on install with Windows Server 2003.</li> </ul>
Version 8.1.6 (11 February 2015)	<ul style="list-style-type: none"> <li>• Use SHA-256 for certificate requests (CSRs) and for generating self-signed certificates.</li> <li>• Can now downgrade the edition automatically when applying purchased activation key.</li> <li>• Added Irename and Idelete to FTP scripting engine, and direct debug output to server log.</li> <li>• Add ability to export IP filter data from manager.</li> <li>• Modify installer so that CompleteFTP service is not removed when installer is run (if it exists).</li> <li>• Tidy overview panel.</li> </ul>
Version 8.1.5 (16 January 2015)	<ul style="list-style-type: none"> <li>• Permit USER/PASS commands to be sent after logging in as per RFC 959.</li> <li>• Drag and drop for IE in the web file manager.</li> <li>• Fixed short timeout for starting downloads in web file manager.</li> <li>• Fix Gateway problem of closing the FTP connection when accessing a gatewayed FTP server via HTTP.</li> <li>• Fixed bug with default domain being cached when authenticating via automatic Windows users.</li> <li>• Improve memory management for SCP transfers of large files.</li> <li>• Fix OnChannelClosed() exceptions in log in SFTP.</li> <li>• Fix unusual stability issue reported by a user.</li> <li>• Fixed some URLs in the manager application.</li> </ul>
Version 8.1.4 (5 December 2014)	<ul style="list-style-type: none"> <li>• Improve performance of SFTP under heavy load, and generally reduce memory consumption.</li> <li>• Include client's IP address in logging.</li> <li>• Improve TLS session resumption.</li> <li>• Fix disabling of SSL renegotiation.</li> <li>• Store logging config in separate file so that it doesn't get overwritten on reinstall.</li> <li>• Resizable panels in the group view, process trigger and email notification panels.</li> <li>• Undeprecated IUserInfo.HomeDirectory.</li> <li>• Fixed pscp bug where "Lost connection" was seen.</li> <li>• Fix HTTP/S intermittent download failures.</li> <li>• Fix gateway OutOfMemory error for large transfers.</li> </ul>
Version 8.1.3 (22 October 2014)	<ul style="list-style-type: none"> <li>• SSL 3.0 is now disabled by default for FTPS and HTTPS. This was motivated by the <a href="#">POODLE vulnerability</a> in SSL 3.0.</li> <li>• Added hmac-sha2-256 and hmac-sha2-512 support to SSH/SFTP.</li> <li>• Optimize use of SMTP client for the server so that email triggers scale better.</li> <li>• Fix minor installer issues - some files weren't being deleted on uninstall, and the manager is now detected if running when the installer is started.</li> <li>• Ensure that the Admin site is always listening on localhost no matter what configuration changes are made.</li> <li>• Fixed an SFTP bug that caused problems for file transfers in the multi-protocol gateway.</li> </ul>

Version 8.1.2 (26 September 2014)	<ul style="list-style-type: none"> <li>• Add the ability to prevent files with name matching set filemasks from being uploaded or renamed to.</li> <li>• No longer a restriction on number of gateways that can be set up.</li> <li>• Improved error-reporting in web file-manager.</li> <li>• Updated the edtfp.exe scripting binary - adds some new features.</li> <li>• Fixed problems with multi-file upload in web file-manager ('invalid backend response' error).</li> <li>• Fix hanging session problem for users with disk quota.</li> <li>• Fixed "The path is not of a legal form" error in process trigger.</li> <li>• Fixed bug in adding multiple non-Windows users, which added a generated username if the user was already present.</li> </ul>
Version 8.1.1 (29 July 2014)	<ul style="list-style-type: none"> <li>• Change from superceded OID for sha1withRSA to PKCS#1 OID in certificate request generator.</li> <li>• Fix bug where HTTP SSL ciphers couldn't be disabled.</li> <li>• Fix problem of Filezilla 3.6.x and later versions getting a parse error on newly generated self-signed certificates in FTPS.</li> <li>• Now rejects client-initiated SSL renegotiation.</li> <li>• Fixed cross-site scripting security hole.</li> <li>• Fix quota bug for random write access.</li> <li>• Fix problem where clients that stop transmitting during transfers but keep connection open do not time out.</li> <li>• Fix so that Chrome FTP works correctly (sends two PASV commands).</li> <li>• Rationalise macros across email &amp; process triggers.</li> <li>• Fixed search in CompleteFTP manager.</li> <li>• Fixed bug where HTTP failed on a site if anonymous user was disabled for that specific site.</li> <li>• Fix Elfinder download bug which occurs when home is set to root.</li> </ul>
Version 8.1.0 (24 February 2014)	<ul style="list-style-type: none"> <li>• Numerous improvements to the <a href="#">CompleteBox</a> file-sharing client, including file synchronization.</li> <li>• Multi-protocol wrapper now supports publickey authentication for SFTP.</li> <li>• SSH user authentication methods can now be specified on a per user basis in the Enterprise Edition.</li> <li>• Discard changes button in the manager client.</li> <li>• Search added to User Guide, and PDF version included.</li> </ul>
Version 8.0.1 (12 December 2013)	<ul style="list-style-type: none"> <li>• Add HTTP PUT.</li> <li>• Fix bug where some users received an "Index does not exist" error message in the CompleteFTP manager when deleting a folder.</li> <li>• Fix CompleteBox right-button click from Windows Explorer stopped working bug.</li> </ul>
Version 8.0.0 (2 December 2013)	<ul style="list-style-type: none"> <li>• Added support for file-sharing, via a file-sharing client, <a href="#">CompleteBox</a>.</li> <li>• Added the ability to write web-apps in Javascript, making CompleteFTP a powerful platform for developing file-oriented web applications.</li> <li>• Improved ability to download diagnostic information for support.</li> <li>• Significantly improved HTTP support.</li> <li>• Many minor enhancements and bug fixes.</li> </ul>
Version 7.4.0 (4 October 2013)	<ul style="list-style-type: none"> <li>• SSH local port forwarding added, so that SSH tunnels can be set up.</li> <li>• Add new SSH admin shell commands: folderadd, foldermodify, folderdel.</li> <li>• Added timestamp preservation to scp.</li> <li>• Added SSL session resumption to FTPS.</li> <li>• Implemented rename in the Gateway.</li> <li>• Fixed a connection leak in the Gateway.</li> <li>• Fixed incorrect key fingerprint displayed for user keys.</li> <li>• Implemented FXP_FSETSTAT in SFTP.</li> <li>• Fixed scp file overwrite bug.</li> <li>• Fixed bug where RSA host key length was only 1024 in new sites - now 2048 bits.</li> <li>• Manager does not need to be running as administrator to import a new certificate and PVK file.</li> <li>• Fix disk quota bug.</li> <li>• Fix cyclic error message when using cp and mv commands in the SSH shell to copy a file into a subdirectory.</li> <li>• Ensure allowed protocol list is viewable in Standard Edition.</li> </ul>
Version 7.3.0 (3 June 2013)	<ul style="list-style-type: none"> <li>• Addition of an "Allow always" category in IP filtering. These IPs can't get autobanned.</li> <li>• Allow numeric ranges in IP filters.</li> <li>• In SFTP, if file exists don't send an error in response to FXP_STAT if size or last modified time can't be obtained. This helps some Unix SFTP clients to work correctly.</li> <li>• Improved connection establishment performance by using cached local IPs.</li> <li>• Fixed bug where HTTP session timeout of 0 was not infinite.</li> <li>• Fix problem in email triggers where duplicate emails were sent under certain circumstances.</li> <li>• Move jquery code to local disk to prevent security error.</li> <li>• Call Dispose() on extensions.</li> </ul>
Version 7.2.1 (23 April 2013)	<ul style="list-style-type: none"> <li>• Now <a href="#">Windows Server 2012 certified</a>.</li> <li>• Improved installer to better detect .NET versions.</li> <li>• Fixed bug in SSH authentication when not all auth methods enabled.</li> </ul>

Version 7.2.0 (21 March 2013)	<ul style="list-style-type: none"> <li>Added HTTP file management - a Javascript file manager supporting upload, download, file viewing and more. Professional and Enterprise Editions only.</li> <li>Added support for Windows RODCs when listing users (previously needed to be writable).</li> <li>Enabled public key authentication to be used with Automatic Windows Users (the password must still be supplied).</li> <li>Database authentication now supports use of password salts.</li> <li>Allow admin user-name change in Standard and Professional Editions.</li> <li>Make sure # can be used in file-names in HTTP.</li> <li>Minor display bug fixes in the CompleteFTP manager.</li> <li>Documented how to achieve syslog integration.</li> </ul>
Version 7.1.2 (5 February 2013)	<ul style="list-style-type: none"> <li>Unlimited-size, unbuffered HTTP uploads. Previously uploads were capped at 20MB because they were buffered in memory before being written to disk.</li> <li>Added support for customized and automated installation.</li> <li>Corporate Editions now come with an optional universal activation key (so that online activation is not required).</li> </ul>
Version 7.1.1 (18 December 2012)	<ul style="list-style-type: none"> <li>Performance enhancements.</li> <li>Fixed SFTP bug where if first_kex_packet_follows=true is set, the guess is handled incorrectly.</li> <li>Fixed "Server offline" bug, where if the server is taken offline in the manager, it can't easily be put online again.</li> </ul>
Version 7.1.0 (26 November 2012)	<ul style="list-style-type: none"> <li>Added support for customizable HTTP listings.</li> <li>Added support for remote ZIP-file navigation. ZIP-files on the server can now be mounted as a directory, changed into, and individual files retrieved (write is not supported as this stage).</li> <li>Added support for SSH keypair generation for users.</li> <li>Make restarts of file downloads possible for &gt; 2GB restarts (in FTP and FTPS).</li> <li>Login/logout events in HTTP.</li> <li>Add Groups to ISession for use in extensions.</li> <li>Disallow slashes in virtual folder-names and prevent duplicate folder names. Add hidden folder flag for virtual folders.</li> <li>Allow CoreFTP to use MDTM in a non-standard way for compatability.</li> <li>Add option to exclude admin logging in log-view</li> <li>Make virtual directories non case-sensitive.</li> <li>Fix broken audit log, which wasn't logging anything.</li> </ul>
Version 7.0.1 (5 October 2012)	<ul style="list-style-type: none"> <li>Make CompleteFTPServer.dll strongly named.</li> <li>Add the FilenameOnly listing format for FTP.</li> <li>Enable macros to be placed in the names of virtual folders.</li> <li>Fixed trial installation problem where the error "Key not valid for use in specified state" was seen.</li> <li>Fixed bug applying Windows permissions of auto-created folders.</li> <li>Fixed folder re-ordering bug that occurred when renaming folders.</li> </ul>
Version 7.0.0 (20 September 2012)	<ul style="list-style-type: none"> <li>Major new feature - cluster support in Enterprise Edition!! This means configuration changes on the primary server in the cluster are automatically propagated to the secondary servers. Very useful for failover and load balancing.</li> <li>Added ability to add users via a script.</li> <li>Fixed listings so if a single file was specified, its details are shown.</li> <li>Added IsValidRSAKey and IsValidDSAKey methods to Authenticator so that users can validate public keys themselves as part of implementing an authentication extension.</li> </ul>
Version 6.4.2 (23 July 2012)	<ul style="list-style-type: none"> <li>Groups can now be defined for database users by providing an additional field in the query that supplies a comma-separated list of group-names.</li> <li>Ensure any temporary script files are created by the process user.</li> <li>Make standard RSA keylengths 2048 bits (increased from 1024 bits).</li> <li>Fix bug when trying to log in as a disabled user (IsAnonymous NRE).</li> <li>Fix security vulnerability in directory navigation.</li> <li>Fix potential security vulnerability in SSH DSA signing. Thanks to Nadia Heninger for pointing this out.</li> <li>Fixed bug that made using multiple email addresses as targets for email triggers difficult to use.</li> </ul>
Version 6.4.1 (28 May 2012)	<ul style="list-style-type: none"> <li>Modify response when anonymous user is disabled so that Internet Explorer pops up its login prompt.</li> <li>Fix bug where PVK files generated during CSR creation failed to import when a new certificate was being imported.</li> <li>Fixed bug in when adding multiple Windows users. If UseHomeForAllSites was not selected then would get "Please make sure every enabled site has a home-folder defined."</li> </ul>
Version 6.4.0 (7 May 2012)	<ul style="list-style-type: none"> <li>Added support editing of properties of multiple users in a single operation. Users can be selected by group membership and wildcard filter. Users may be added to groups when they're being added using the 'add multiple non-Windows users' form.</li> <li>Group membership is now shown in the user-list in the Users panel.</li> <li>Users can now be set an expiry date beyond which they cannot log on.</li> <li>Added support for the generation of Certificate Signing Requests (CSRs) via the CompleteFTP manager.</li> <li>Made it easier to implement custom authenticators, and added support for doing public key authentication.</li> <li>Added support so that SSH/SFTP users can change their passwords if this is permitted.</li> <li>Added support in custom authenticators for forcing password changes for SSH/SFTP users.</li> <li>When using FTP scripts as process triggers, sometimes they fail with the error "The directory name is invalid". Now fixed. Also upgraded script engine.</li> <li>Fixed bug where automatic Windows users failed to authenticate in HTTP/HTTPS.</li> <li>Permit paths to contain '.</li> <li>Fix bug where the manager overview panel fails to restore after minimizing.</li> </ul>

Version 6.3.0 (27 February 2012)	<ul style="list-style-type: none"> <li>• Add HTTP to the multiprotocol gateway (Enterprise MFT only).</li> <li>• Add new macros, TIMESTAMP:format and TransferStatus for use in events.</li> <li>• Create new email notifications and process triggers by copying existing ones.</li> <li>• Redirect to user's home directory when logging in via /login page in HTTP.</li> <li>• System folders can now be viewed in folder browser in folder tab of manager.</li> <li>• Minor GUI improvements in the manager.</li> <li>• Fix bug where encrypted passwords were enabled and changing the admin password meant they could not be decrypted (and resulted in an error being displayed).</li> <li>• Fix trigger macro bug where WindowsPath and WindowsFolder macros were returning incorrect values when the home-appears-as-root feature was enabled.</li> <li>• Fix bug in email triggers where "Notify on Error" could not be saved.</li> <li>• Fix problem with license.exe - error message was "Could not load file or assembly 'System.Data.SqlServerCe, Version=3.5.1.0'".</li> <li>• Fixed file-system permission problem, which applied incorrect folder permissions for sub-folders if "User's home appears as root" was enabled. Previously the permissions of the user's home folder would be applied to all subfolders, overriding any that may have been defined for these.</li> </ul>
Version 6.2.0 (28 December 2011)	<ul style="list-style-type: none"> <li>• Major Enterprise Edition feature added - the Gateway. This allows CompleteFTP to act as a gateway to other FTP servers - translating protocols in the process.</li> <li>• Added ability to define custom MIME types.</li> <li>• Permit non-Windows users to use the SSH terminal.</li> <li>• Minor bug fixes, and tweaks to manager GUI.</li> <li>• Fix bandwidth limiting bug in SFTP.</li> <li>• Fixed bug where invalid event data could hang subsequent events.</li> </ul>
Version 6.1.0 (20 November 2011)	<ul style="list-style-type: none"> <li>• Introduced overview panel for Complete FTP Manager.</li> <li>• Added support for HTTP FORM uploads.</li> <li>• Make SFTP the default protocol for the Complete FTP Manager for new installations.</li> <li>• Add flag to enable support of backslashes in paths.</li> <li>• Fix HTTP session bug that can result in multiple authentication prompts.</li> <li>• Simplify installer options and operation.</li> <li>• Fix bug re client certificate authentication. Enforce admin permissions for managing the trusted certificate store.</li> </ul>
Version 6.0.0 (23 August 2011)	<ul style="list-style-type: none"> <li>• Major Enterprise Edition feature: added support for multiple sites.</li> <li>• Added HTTP directory listings (Professional and Enterprise Editions).</li> <li>• Fixed bug preventing change of directory with a relative path into a virtual dir.</li> <li>• Fixed bug giving inconsistent timestamps for virtual dirs.</li> <li>• Fix bug where administrator password could not be changed if SFTP protocol was chosen.</li> <li>• Added support for multiple deletion of folders and users.</li> <li>• Improved memory management significantly.</li> <li>• Allow indefinite auto-bans (i.e. until next server restart).</li> <li>• Allow additional administrative users.</li> <li>• OPTS UTF ON/OFF added for FTP.</li> <li>• Make contextual help hidable.</li> </ul>
Version 5.2.1 (18 May 2011)	<ul style="list-style-type: none"> <li>• Introduction of Enterprise Edition with support for custom extensions for authentication, events, site commands and the file system.</li> <li>• If AUTH argument not understood the client can now respond with USER if FTPS is not mandated.</li> <li>• Now examines the first character of the TYPE argument and compares with I or A rather than comparing the entire argument string.</li> </ul>
Version 5.2.0 (27 April 2011)	<ul style="list-style-type: none"> <li>• When adding multiple Windows users a single home directory can be now be specified (usually a macro folder).</li> <li>• Added read-only flag for users. If a user is read-only then uploading (and other write operations) is not permitted for that user regardless of what file permissions are set.</li> <li>• Make FEAT more RFC compliant (remove '211-' in front of each feature in the list).</li> <li>• More firewall friendly for FTP active mode transfers (server binds to FTP data port 20).</li> <li>• FTP commands are now uppercased to cater for clients that send 'pasv' rather than 'PASV'.</li> <li>• Improved installer. Status updates are only displayed for certificate and key generation if they are actually being generated.</li> <li>• Fixed bug caused by clashes between naming of virtual file-system folders and Windows file-system folders. Windows folders now take precedence.</li> <li>• Prevent a file being closed prior to last FXP_WRITE by queuing the message (also for FXP_SETSTAT).</li> <li>• Fix bug limiting the number of email notifications (again!), and a bug where changed notifications were saved to the config database but not immediately applied.</li> <li>• Fix bug preventing notifications for HTTP.</li> <li>• Fix SFTP authentication bug re enabled flag.</li> <li>• Fix Windows user directory traversal bug.</li> </ul>
Version 5.1.1 (15 March 2011)	<ul style="list-style-type: none"> <li>• Fixed bug where FTP connection would be broken by the passive wait timeout if there was a error while getting a listing or transferring in passive mode.</li> <li>• Fix bug in "Admin" configuration re error message "The value for column "ExtUserID" in table "Site" is DBNull".</li> <li>• Fix bug in email and process trigger filename filters, where wildcards were being treated as regex's.</li> </ul>

Version 5.1.0 (21 February 2011)	<ul style="list-style-type: none"> <li>• Permit export of server's private key to file (only if manager is running locally).</li> <li>• Fix for Fetch (Mac OSX), which can't cope with FXP_READ replies that are out of order.</li> <li>• Fix for scp when "User's home folder appears as root" is set.</li> <li>• Fix in FTP so that 550 is returned when attempt made to delete a non-empty directory.</li> <li>• Support for aes128-ctr, aes192-ctr and aes256-ctr in SFTP added.</li> <li>• Support for diffie-hellman-group-exchange-sha1 and diffie-hellman-group-exchange-sha256 in SFTP added.</li> <li>• Fixed bug re anonymous user being permitted when the anonymous user was disabled (but anonymous enabled in Settings).</li> <li>• Fixed bug with multi-line welcome text in FTP when product details are hidden (Windows 7 ftp.exe hung).</li> <li>• Automatic backup of configuration file on a daily basis (if altered).</li> <li>• Fix for infrequent occurrence of server stopping listening.</li> </ul>
Version 5.0.0 (20 December 2010)	<ul style="list-style-type: none"> <li>• Now supports HTTP and HTTPS.</li> <li>• Added support for external users via an OLEDB connection.</li> <li>• Added support for bulk user imports.</li> <li>• Process triggers now support FTP scripts.</li> <li>• Now supports TLS 1.1 for FTPS.</li> <li>• Handles anonymous user logins in a more standard manner for FTP.</li> <li>• Improved scalability re simultaneous connections.</li> <li>• Add login and logout event auditing.</li> <li>• SITE commands now case insensitive.</li> <li>• Listeners are started/restarted appropriately when "Apply Changes" is selected in the manager and protocols have been enabled or ports changed.</li> <li>• Can now hide product and version number in welcome messages/banners, for PCI compliance.</li> <li>• Fixed error with FTPS implicit where AUTH is (unnecessarily) required to be sent by the client.</li> </ul>
Version 4.1.0 (10 September 2010)	<ul style="list-style-type: none"> <li>• Pro version now supports multiple SSH keys per user.</li> <li>• Fix bug where timestamp of newly uploaded file can be incorrectly set.</li> <li>• Implicit FTPS now supported.</li> <li>• SSL email servers now supported for notifications.</li> <li>• SSH exec now supported (for the same commands available in the SSH shell).</li> <li>• Fix bug that allowed guest users access if guest is enabled on the server machine.</li> <li>• Fix bug limiting the number of email notifications.</li> <li>• Improve server memory management.</li> </ul>
Version 4.0.3 (1 June 2010)	<ul style="list-style-type: none"> <li>• Security fix to disallow paths containing '...' and '!. This prevents unauthorised directory listings and transfers from outside a user's home directory.</li> <li>• Fix so that Powershell scripts are piped directly into Powershell rather than via a temporary file.</li> <li>• Minor GUI fixes to process trigger setup. Also path filter was not being used in process trigger - now fixed.</li> </ul>
Version 4.0.2 (30 April 2010)	<ul style="list-style-type: none"> <li>• Domain name filters added.</li> </ul>
Version 4.0.0 (16 April 2010)	<ul style="list-style-type: none"> <li>• Extensive real-time charts for performance monitoring. (Professional Edition only).</li> <li>• Added Process Triggers - the ability to launch processes on the server when events such as uploading or downloading occur (Professional Edition only).</li> <li>• SSH terminal access added for Windows users (Professional Edition only). Includes ability to execute processes on the server.</li> <li>• Disk quotas introduced (Professional Edition only).</li> <li>• Bandwidth quotas introduced (Professional Edition only).</li> <li>• Can now create a new folder from within the add-user forms.</li> <li>• Fix bug where creating a directory failed in WinSCP (when using SCP only).</li> <li>• Bug fix where it was reported that there was an unhandled exception causing the service to terminate.</li> <li>• Fixed bug where changing directory using forward slashes permitted navigation out of user's directory.</li> <li>• Fixed problem with determining AD group membership.</li> <li>• By default a maximum of 10 anonymous connections are now permitted (prevents someone denying service by connecting as anonymous many times).</li> </ul>
Version 3.3.0 (22 February 2010)	<ul style="list-style-type: none"> <li>• Add support for autobanning of IP addresses (configurable in Professional Edition). If a client IP has too many authentication failures, it is automatically banned for a period.</li> <li>• Add support for WinSCP with the SCP protocol (already works with WinSCP/SFTP).</li> <li>• Some minor SCP fixes.</li> <li>• Fix manager logging problem - now logs correctly to ApplicationData (for the user).</li> <li>• Fix manager problem where new filters could not be added for the administration server.</li> </ul>
Version 3.2.2 (2 February 2010)	<ul style="list-style-type: none"> <li>• Fix bug in handling of invalid SSH_FXP_EXTENDED messages.</li> <li>• Add filtering for reducing the number of users returned from large active directory domains.</li> <li>• Can now install the manager only for remote administration from another machine (production install only).</li> <li>• Fixed upload bug in SCP resulting from an infrequent race condition.</li> <li>• Increased minimum SSH window size to 64K to fix problem uploading with cURL.</li> <li>• CompleteFTP manager now logs to the same directory as the server.</li> <li>• Fixed installer bug whereby it always asks if the existing configuration should be kept even when it doesn't exist.</li> </ul>
Version 3.2.1 (18 December 2009)	<ul style="list-style-type: none"> <li>• Fix "User's home folder appears as root" bug which prevented users from logging in when this flag is set (SFTP only).</li> <li>• Code sign the installation executable.</li> </ul>

Version 3.2.0 (8 December 2009)	<ul style="list-style-type: none"> <li>• By default permit all Windows users to logon without explicitly adding them to the user database (see Settings). This can be restricted to users in active directory and/or local groups</li> <li>• Add option for home directory to appear as root ("/").</li> <li>• Add new permissions to Pro version: list folder, delete, create folder permissions. Make Windows and non-Windows permissions distinct.</li> <li>• Added license import/export command-line tool.</li> <li>• Permit non-Windows users to change their passwords (disabled by default).</li> <li>• Fix to prevent error when Cyberduck uploads files.</li> <li>• Fixed key re-exchange bug when kex initiated by clients.</li> <li>• Fixed Blowfish bug in SFTP.</li> <li>• Can now optionally delete the configuration on reinstalling.</li> <li>• When making a license request, allow copy to clipboard for separate emailing.</li> <li>• Allow admin to highlight and kill multiple connections. Auto-refresh of connection view.</li> </ul>
Version 3.1.2 (26 October 2009)	<ul style="list-style-type: none"> <li>• Added new folder-type called 'Network Folder', which supports UNC paths.</li> <li>• Fixed erroneous license warning from Manager when used remotely.</li> <li>• Fix idle timeout problem causing SFTP sessions to never time out if disconnected abruptly.</li> <li>• Fix 'All users (read)' folder permission in Standard edition.</li> <li>• Allow path to config.sdf to be specified on CompleteFTPConsole.exe command-line.</li> <li>• Better error-handling in Windows file-listings.</li> <li>• Fixed error in new-user form causing half-configured user to be added when the user-root directory does not exist.</li> <li>• Improved login error messages.</li> </ul>
Version 3.1.1 (16 October 2009)	<ul style="list-style-type: none"> <li>• Fixed problem where not all active directory users were listed if greater than 1000 users.</li> <li>• Fixed problem where failed to login domain users with non-alphanumeric chars in username.</li> <li>• Fixed problem where failed to log in local Windows users that were also on a domain.</li> <li>• Fixed problem where certificate parsing failed if there was a comma or semicolon in the subject.</li> <li>• Fixed problem where multiple Windows users could not simultaneously share MyDocuments.</li> <li>• Stop CFTP Manager minimizing to tray.</li> <li>• Stop Windows users from accessing other home directories.</li> </ul>
Version 3.1.0 (4 September 2009)	<ul style="list-style-type: none"> <li>• User-specific protocol settings.</li> <li>• Reduce memory usage during log file viewing.</li> <li>• Stop scp client from hanging if SCP is disabled in the server and a command is attempted.</li> <li>• Fixed error where target filename could not be specified in scp.</li> <li>• Improved passive mode port range handling.</li> <li>• Logging improvements.</li> <li>• Fixed bug preventing deletion of e-mail notifications.</li> <li>• A variety of minor GUI fixes.</li> </ul>
Version 3.0.1 (7 August 2009)	<ul style="list-style-type: none"> <li>• Fix logging bug where can't change log level to Debug.</li> <li>• Fix notification bug where can't delete last notification.</li> <li>• Fix incorrect log file location (ProgramFiles -&gt; ProgramData).</li> <li>• Fix SCP bug where copy failed if target was a filename.</li> <li>• Minor performance tuning changes.</li> </ul>
Version 3.0.0 (30 July 2009)	<ul style="list-style-type: none"> <li>• Added support for SCP (Professional Edition).</li> <li>• Permission editing added (more sophisticated version in Professional Edition).</li> <li>• Email notification for a variety of events such as file upload, download, delete (Professional Edition).</li> <li>• Auditing (i.e. tracking all changes to files - Professional Edition).</li> <li>• FTPS client certificate verification (Professional Edition).</li> <li>• IP filtering added - accept/reject connections based on IP address (Professional Edition).</li> <li>• Can now specify the network interfaces to listen on.</li> <li>• No longer disconnects channel when client sends SSH_CHANNEL_EOF.</li> <li>• Added support for MODE S (which is the default).</li> <li>• Fix re file overwrite problem.</li> </ul>
Version 2.2.1 (24 June 2009)	<ul style="list-style-type: none"> <li>• Now closes console gracefully.</li> <li>• Fixed bug whereby a disabled user could still log in via SFTP.</li> <li>• Fixed bug where newly generated SSL certificate failed to save to the config.</li> <li>• Fixed auto-add user checkbox in 'Add Windows user form'.</li> <li>• Fixed bug where when a new activation key is entered, it is saved but the manager appears to show that it has not.</li> <li>• Manager now uses compression to speed up log file transfers.</li> <li>• Manager connection editor now allows clearing of properties.</li> <li>• Added extra SFTP and FTPS properties to Manager connection settings.</li> <li>• Improved handling of socket accept errors.</li> </ul>

Version 2.2.0 (2 June 2009)	<ul style="list-style-type: none"> <li>• Introduction of professional version supporting Windows domain users.</li> <li>• Now works on x64 machines.</li> <li>• MODE Z compression added for FTP and FTPS (also OPTS MODE Z).</li> <li>• Can now directly download log files to local disk.</li> <li>• Fixed SFTP authentication bug, where if the username was supplied in the incorrect case, public key authentication failed.</li> <li>• Users can no longer list /home, i.e. they can't see what the other user directories are.</li> <li>• Administrator can retrieve user list (Professional version only).</li> <li>• Manager now displays the version number.</li> <li>• Fixed FEAT bug (indenting too many spaces for each feature listed).</li> <li>• Fixed timeout that occurs in passive mode with large transfers if the passive timeout is set.</li> <li>• Fixed bug whereby SFTP admin could only log in once.</li> <li>• Manager now displays any critical server errors when connecting.</li> </ul>
Version 2.1.2 (7 May 2009)	<ul style="list-style-type: none"> <li>• Can now call TYPE command in between PORT/PASV and RETR/STOR (which some clients do).</li> <li>• Administration functionality can now be performed via SFTP.</li> <li>• Invalid user names aren't immediately rejected, forcing hackers to try authenticating.</li> <li>• Fix for WinSCP problem with files very occasionally being truncated.</li> <li>• Provide alternative for obtaining machine id hash if email client is not installed (for product activation).</li> <li>• Fix problem with using a non-default home directory when creating a Windows user.</li> <li>• Fix resource cleanup bug when client abruptly disappears.</li> </ul>
Version 2.1.1 (23 March 2009)	<ul style="list-style-type: none"> <li>• Added support for the XCRC FTP command to check file integrity.</li> <li>• Better feedback while installer is running.</li> <li>• Added diffie-hellman-group14-sha1 key exchange method.</li> <li>• Should now work on .NET 2.0 SP1 (previously required a minimum of 2.0 SP2).</li> <li>• Fix problem with generated FTPS certificate being initially invalid.</li> </ul>
Version 2.0.0 (11 February 2009)	<ul style="list-style-type: none"> <li>• Now offers full support for SFTP (FTP over SSH). Support for password and publickey authentication, RSA and DSA keys etc.</li> <li>• Can now set logging level in the manager.</li> <li>• User's Guide expanded.</li> <li>• Generation of server certificate takes place on the server.</li> <li>• Numerous enhancements to the Manager.</li> </ul>
Version 1.1.0 (12 November 2008)	<ul style="list-style-type: none"> <li>• Fixed license reading bug - sometimes the license was not read correctly, resulting in the server running in restricted mode.</li> <li>• Fixed user impersonation bug that meant special folders (e.g. My Documents) could not be navigated to.</li> <li>• Database and log files now stored in C:\ProgramData (or equiv) rather than in Program Files.</li> <li>• Default encoding is now UTF-8. Encoding can now be changed.</li> <li>• Added User's Guide.</li> <li>• Improved responsiveness of Windows folder browser.</li> <li>• Set default name of Windows virtual folder to name of actual folder.</li> <li>• Fixed FTPSEnabled property.</li> <li>• Added ability to add multiple windows users in one operation. Fixed NRE when NodeForm was cancelled.</li> <li>• More descriptive message when max number of connections exceeded.</li> <li>• Returns meaningful message when AUTH is used and FTPSEnabled is off.</li> <li>• Rearranged controls and changed text to make it clearer that the settings are the settings of the admin connection and not of the server itself.</li> </ul>
Version 1.0.0 (10 October 2008)	<ul style="list-style-type: none"> <li>• First commercial version! After many months of further development and testing.</li> <li>• New manager, better FTPS support, remote administration, too many other new features to explain here.</li> </ul>
Version 0.5.1 (5 December 2007)	<ul style="list-style-type: none"> <li>• Fix permitting LIST to accept '-' arguments.</li> <li>• Disable idle time forced logouts by default.</li> </ul>
Version 0.5.0 (13 November 2007)	<ul style="list-style-type: none"> <li>• First release, supporting FTP and FTPS.</li> </ul>

# Installation

**CompleteFTP** is supplied as an executable Windows installer. It is also possible to [customize and automate installation](#).

Prior to installing, check that the software and hardware [requirements](#) are satisfied.

To install, double-click on the installer executable, and you will be taken through the steps required to set up CompleteFTP. You may require Administrator permissions to run the installer.

The trial installer installs both the server components and the Complete FTP Manager. The production installer by default installs both, but optionally permits just the Complete FTP Manager to be installed if required, so that it can be used for remote administration of a server installed on another machine.

Please note that CompleteFTP installs as a [Windows Service](#), running as the local SYSTEM user. It will be started automatically when your computer starts. Once installed, run the Complete FTP Manager to begin [adding users](#).

The trial edition expires after 31 days. Once the trial period is over, the server converts to Developer Mode, and will only accept connections from *localhost* (but will otherwise remain fully functional). CompleteFTP can be used in Developer Mode indefinitely. It may be converted to production mode by [purchasing a license](#).

Once a license has been purchased, a trial or developer mode installation (or a production installation) can be **activated** as described [here](#).

There are detailed [getting started tutorials](#) available that explain how to connect with Filezilla and WinSCP.



# Uninstalling

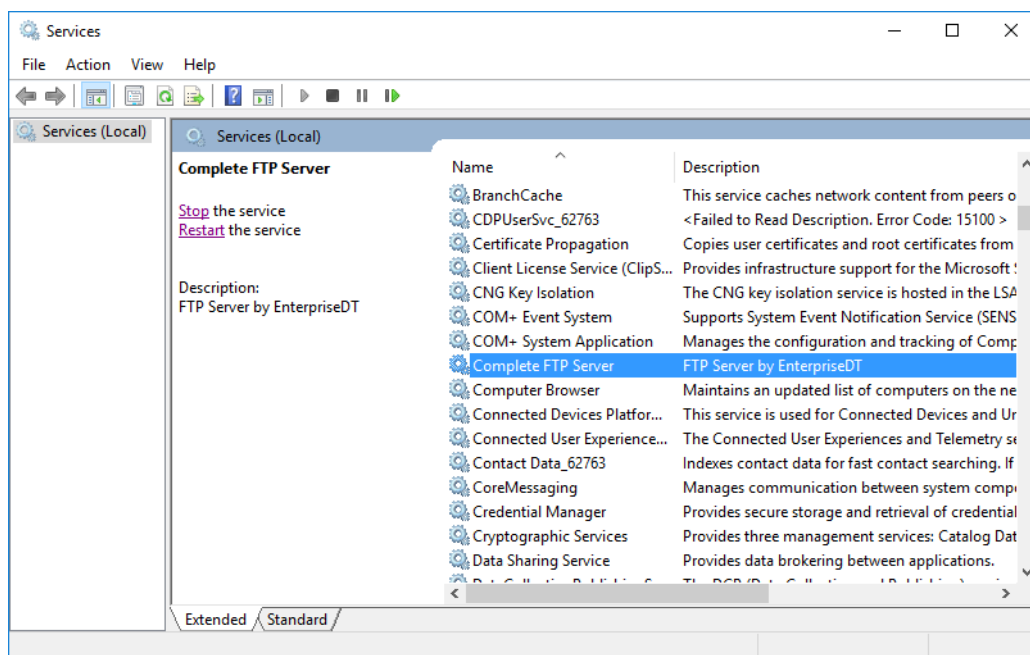
To uninstall CompleteFTP:

1. Press Windows+X
2. Select Apps and Features to show the following:  

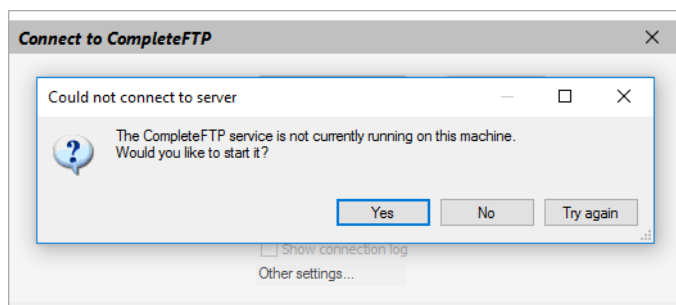
□
3. Click the vertical dots next to CompleteFTP
4. Select Uninstall
5. Follow the prompts

# How to run the server

*CompleteFTP* is installed as a Windows service, running as the local SYSTEM user, and by default is configured to run automatically on machine startup. The service can be stopped and started via Control Panel -> Administrative Tools -> Services. Select the CompleteFTP Service. Under more recent versions of Windows you will need to have administrator privileges to do this.



The service can also be started from the CompleteFTP Manager - but only if 1) it has been started on the server machine and 2) it was launched with administrator privileges.



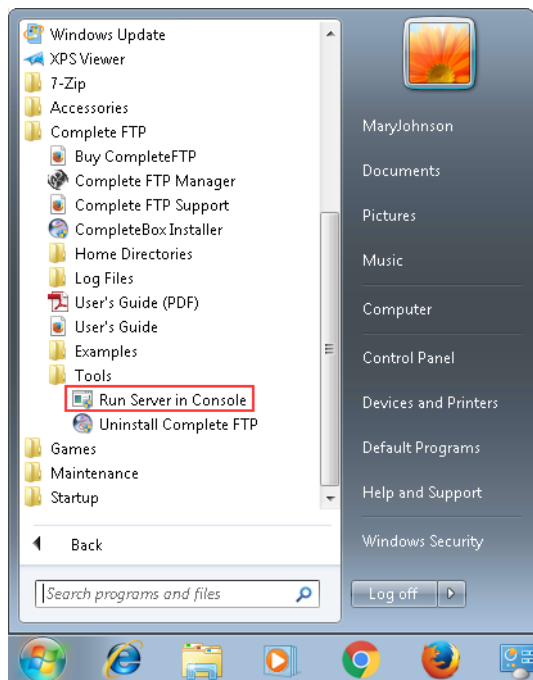
If the CompleteFTP service fails to start, or starts with errors, the most likely problem (other than the lack of administrator privileges) is:

- the standard FTP port, 21, is already in use on the machine
- the standard SSH port, 22, is already in use on the machine
- the standard HTTP (80) or HTTPS (443) ports are already in use on the machine

Using 'netstat -a' in a DOS prompt will reveal what ports are in use. IIS is usually the most likely application already using ports.

If IIS will continue to be used for HTTP and HTTPS, then HTTP and HTTPS will need to be disabled in CompleteFTP - see [here](#).

CompleteFTP can also be run as a console application (rather than as a service). This can often be useful in debugging problems. Select *Run server in console* from the CompleteFTP program menu (with the service stopped) to do so. It is usually necessary to "Run as administrator".



# How to use the manager

The *CompleteFTP manager* controls all settings for the CompleteFTP server (which typically runs as a Windows service).

The manager is started by selecting the CompleteFTP Manager icon from the CompleteFTP menuNormally, the service must already be running for the manager to connect to it. If the server is not running, the manager will prompt to ask if the CompleteFTP Service should be started.If the manager is running with sufficient permissions, the service will be started. If not, the Windows Control Panel -> Administrative Tools -> Services must be used to start the service.

The connect dialog box is brought up when the manager is selected. This allows details of the host that the CompleteFTP service is running on, and the administrator credentials to be entered.

Connect to CompleteFTP

Server name or IP address:localhost

CONNECT

Server admin port:14983

Enable auto-connect

Administrator username:admin

Administrator password:\*\*\*\*\*

☒ Save password

☐ Show connection log

Other settings...

The defaults should generally be used (they are set in installation), and are shown below:

Setting name	Default value
Server name/IP	localhost
Server admin port	14983
Administrator username	admin
Administrator password	[whatever was entered on install]

Note that more detailed settings for the connection can be accessed via the *Other settings...* link. Normally these should not be modified. The "Save password" checkbox can be checked to save the administrator password so it is automatically entered, and "Show connection log" can be checked if there are problems connecting to help resolve the issue.

Once the CompleteFTP manager is running, all settings can be accessed via the list of panels on the lefthand side. These include settings for the [Site\(s\)](#), [File System](#), [Users](#), [Events](#) (Professional Edition), [Extensions](#) (Enterprise MFT), [Monitoring](#) and [Licensing](#).

# Migrating to another machine

At times it may be necessary to migrate a CompleteFTP installation to another machine. We recommend you have a current support agreement prior to migrating your installation. This gives you access to assistance with the migration and the latest version of CompleteFTP.

If you do not have a current support agreement, you can still freely transfer your license if you are on version 9.1 or higher (via the release license option). If you're using versions prior to 9.1 or if you don't have access to the old server then you will need technical assistance, which requires a current support agreement.

## Migration steps

To migrate CompleteFTP server from one machine (source) to another machine (target) while preserving settings such as user logins, passwords and home folders, follow the steps below:

1. a) If you are on version 9.0 or below, [contact support](#) to request an additional activation key (unnecessary for licenses that included unlimited keys). If you do not have a current support agreement you will need to renew support to move your license.  
OR  
b) If you are on version 9.1 or above, you can use the release license option in the Licensing panel of the CompleteFTP manager. This puts the server into migration mode, and it is fully operational for a further 30 days. We advise doing this only when the new machine is ready to activate (Step 7). Note that you can upgrade to the latest version on the original machine so this option can be used.
2. Download the latest version of CompleteFTP that you are entitled to from our [customer site](#). You'll need to know your registered email address for your account if you've lost your account details. If you can't log in, please [contact support](#). You'll need a current support agreement to get the latest version - the trial won't work.
3. Install CompleteFTP on the target machine. This creates all the default directories and a blank configuration file.
4. Locate the configuration file (config.sdf for versions before 22.0.0, config.db for 22.0 and later) on the target machine and delete it (see "Locating the data directory" below).
5. Locate the configuration file (config.sdf for versions before 22.0.0, config.db for 22.0 and later) on the source machine (see "Locating the data directory" below).
6. Copy the configuration file from the source machine to the same location on the target machine (see "Locating the data directory" below).
7. Copy all data directories from the source to the target. By default, these are subdirectories of the directory where the configuration file is stored (see "Locating the data directory" below), but directories in other locations may be referenced as well.
8. Install CompleteFTP (again) on the target machine, choose the option to keep the configuration file, and confirm everything works locally (it will have the wrong activation key so you won't be able to connect from external machines). Note: if you have specific IP addresses configured for certain protocols, you will need to alter these if your new machine has a different IP address.
9. [Reactivate](#) CompleteFTP on the target machine. If you are running 9.1 or above, this is the time to release the license from the original machine (Step 1b)). If you have trouble activating your new installation then please [contact support](#).

## Locating the data directory

On almost all Windows machines, the data directory is:

```
C:\ProgramData\Enterprise Distributed Technologies\Complete FTP
```

On earlier machines running Windows XP and Server 2003 the data directory is:

```
C:\Documents and Settings\All Users\Application Data\Enterprise Distributed Technologies\Complete FTP
```

# How to automate installation

The CompleteFTP installer is a standard **NSIS-based** Windows installer, which uses a GUI to prompt for various options during installation.

Some organizations prefer to automate installations, and use command-line scripts to roll out software. A GUI-based installer is usually not suitable for this - instead a command-line installer that requires no user input is required. This article describes how to create and use command-line installers for CompleteFTP.

There's a distinction between a first-time installer and an update installer. A first-time installer is used only when a completely new installation is required, or when an existing installation is to be completely overwritten. An update installer should be used when an existing installation is to be updated to a more recent version.

The general pattern for creating and using automated installers is to:

1. Install CompleteFTP using the standard GUI installer on one machine (the 'development' machine)
2. Create archives of the required files
3. Un-archive these on another machine (the 'production' machine)
4. Execute various commands to make the software operable on that machine.
5. If necessary, activate the installation with an activation key (AK).

The sections below don't contain the full scripts that must be executed, but rather describe the actions that must be performed.

## First-Time Installation

A first-time installation is an installation onto a machine that either has no current CompleteFTP installation, or has an existing CompleteFTP installation that is to be entirely overwritten.

An activation key (AK) will be required to activate the installation. Certain Volume Licenses provide access to a **universal activation key** which is very suitable for automated installation as it can be done as part of the install. Without using a universal activation key, each installation will later need to be activated manually.

### Development machine: Creating archives

1. Install CompleteFTP using the standard GUI installer on the development machine
2. Configure CompleteFTP as desired using CompleteFTP Manager
3. Activate the installation using the universal activation key
4. Create a ZIP file called `binaries.zip` that contains the directory `C:\Program Files\CompleteFTP` (including the directory itself)
5. Remove `install.log` and `server\bostrapper.log` from the ZIP-file
6. Create a ZIP file called `data.zip` that contains `C:\ProgramData\Enterprise Distributed Technologies` (including the directory itself)
7. Remove the contents of the `Temp`, `Logs` and `Backups` directories from the ZIP-file
8. Remove any unwanted directories and files from the `Users` directory

### Production machine: Installation

1. Unzip `binaries.zip` in `C:\Program Files` (unzip app needs admin privileges)
2. Unzip `data.zip` in `C:\ProgramData` (unzip app needs admin privileges)
3. Locate the `InstallUtil.exe` program - usually in `C:\Windows\Microsoft.NET\Framework64\v4.0.30319` or later version
4. Install the CompleteFTP service with the command:

```
InstallUtil.exe "C:\Program Files\Complete FTP\Server\CompleteFTPService.exe"
```

5. Change the startup type of the service to Automatic with the command:

```
reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\CompleteFTP /v Start /t REG_DWORD /d 2 /f
```

6. Stop parent directory of user accounts from inheriting permissions from its parent:

```
icacls "C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Users" /inheritance:d
```

7. Remove read permissions for the Users group for the same folder (so that Windows users can't get a list of other users):

```
icacls "C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Users" /remove:g Users
```

8. Each Windows user-account which is registered in CompleteFTP must have permission to their own home-directory, so if there are any such users then use `icacls` to grant them
9. Start service using the command

```
net start CompleteFTP
```

### Updating an Existing Installation

When a production installation is being updated, the new binaries may be copied over the old ones, but the configuration data on that machine must not be overwritten. The main difference between the first-time installation and the update is therefore related to the treatment of the data.

### Development machine: Creating archives

1. Update existing CompleteFTP using the standard GUI installer on the development machine
2. Create a ZIP file called `binaries.zip` that contains the directory `C:\Program Files\CompleteFTP`

3. Remove `install.log` and `server\bostrapper.log` from the ZIP-file
4. Create a ZIP file called `webcontent.zip` that contains the following subdirectories of `C:\ProgramData\Enterprise Distributed Technologies\Complete FTP`:
  - Account
  - FileManager
  - Login
  - Logout
  - Public
  - Share\App
  - Share\Client
  - Share\Help
  - WebAdmin

## Production machine: Installation

1. Stop CompleteFTP Service using the command

```
net stop CompleteFTP
```

2. Delete the directory `C:\Program Files\CompleteFTP` and all its subdirectories
3. Unzip `binaries.zip` in `C:\Program Files` (unzip app needs admin privileges)
4. Unzip `webcontent.zip` in `C:\ProgramData\Enterprise Distributed Technologies\Complete FTP` (unzip app needs admin privileges)
5. Update the configuration database to latest version using the command

```
C:\Program Files\CompleteFTP\cftpconfig
```

6. Perform any configuration modifications (see [cftpconfig](#))
7. Start service using the command

```
net start CompleteFTP
```

## Special notes on upgrading from pre-22.0.0 to 22.0.0 and higher

Versions prior to 22.0.0 were 32-bit applications, so all binaries were under the directory, `C:\Program Files (x86)\Complete FTP`. From version 22.0.0, CompleteFTP is a 64-bit application, so all binaries are under `C:\Program Files\Complete FTP`.

Obviously, any references to CompleteFTP files under `C:\Program Files (x86)\Complete FTP` will need to be modified to point to their new locations. One important change is to the CompleteFTP service, which will need to be changed to point to the new location of the service binary. The easiest way to do this is to uninstall and reinstall the service using the Windows utility *InstallUtil.exe*. For example:

```
InstallUtil.exe /u "C:\Program Files (x86)\Complete FTP\Server\CompleteFTPService.exe"  
InstallUtil.exe "C:\Program Files\Complete FTP\Server\CompleteFTPService.exe"
```

Version 22.0.0 also brought the shift from SQL Server Compact to SQLite, and the name of the configuration file has changed from `config.sdf` to `config.db`.

# How to activate an installation of CompleteFTP

Most CompleteFTP installations start off in trial mode. If an installation is not activated before the trial ends then CompleteFTP will automatically switch to developer mode. In developer mode incoming connections are accepted only if they're from the local machine.

The method of activation of CompleteFTP differs between the [Free Edition](#) and the three [paid editions](#), Standard, Professional and Enterprise.

If you are trying to activate a very old version of CompleteFTP (pre 9.1.0) that does not support automatic activation, then see [here](#).



# Installing CompleteFTP in the Cloud

CompleteFTP can be installed on cloud servers hosted by services such as [Microsoft Azure](#) and [Amazon Web Services](#). Once a virtual machine has been created and Remote Desktop access to it has been obtained, installing CompleteFTP on it is no different to installing it on a local PC - you simply download the installer and run it. CompleteFTP will run on the lowest specced Windows virtual machines available (e.g. 1 vCPU with 2GB RAM), but higher loads will require more vCPUs and RAM.

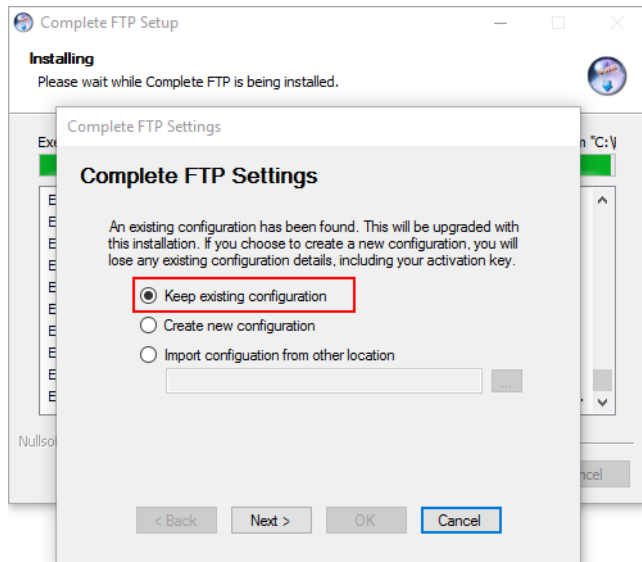
# How to upgrade

To upgrade an existing installation, the new production installer is required. This can be downloaded from the EnterpriseDT website **using the link and login details you received when you purchased your license**. You **cannot** use the trial installer that is available on the website to upgrade a production installation. Please **contact support** if you cannot locate your login details.

Prior to upgrading, we strongly advise doing an upgrade test on a different machine to ensure the upgrade will go smoothly on your production machine, particularly if you are skipping numerous versions. **This FAQ** explains how to test your upgrade.

If you have upgraded and it is not working as you expect, it is straightforward to roll back to your previous version as described in **this FAQ**.

To perform the upgrade, exit the CompleteFTP manager application if it is running, and **install** the new version. The default option is to upgrade your existing configuration to the new version (preserving your settings). You do not need to uninstall prior to running the new installer.

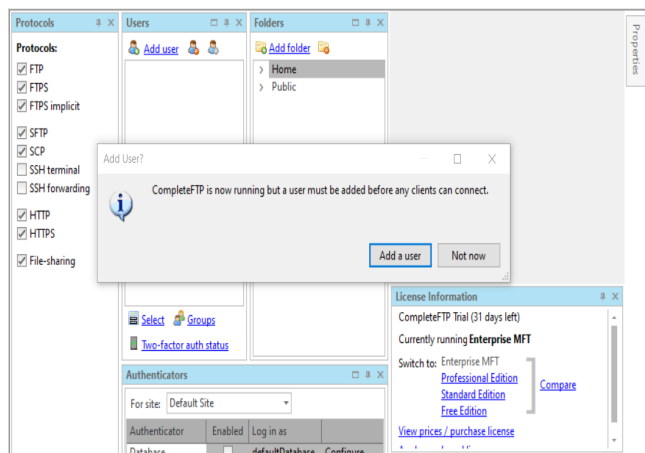


## Getting started with CompleteFTP

This tutorial explains how to get started with **CompleteFTP**, our Windows secure file server supporting a variety of secure file transfer protocols, including FTPS, SFTP, SCP and HTTPS.

## Steps

1. Installation. The latest Windows trial installer can be downloaded from [here](#). If you have already purchased CompleteFTP, the production installer can be downloaded from our [customer portal](#). To install, double-click on the installer executable. More details can be found [here](#).
2. After the installation is complete, [run the Complete FTP Manager](#) to begin adding users.



Add User

Add a non-Windows user

Step 1: Set Username

Username

HenryC

Full name (optional)

Henry Cole

E-mail (recommended - used when resetting password)

test@test.com

Description (optional)

test

Previous

Next

Cancel

3. A password is required for any user account you create. It is advisable for this to be as secure as possible. Alternatively, use the 'Random' button to quickly generate a secure password.

**Add a non-Windows user****Step 2: Set Authentication**

Set a password and/or add one or more public keys. Passwords may be used for all protocols. Public keys are used for SFTP, SCP and SSH. At least one authentication method must be enabled.

**Password**☒ Enabled☐ Show charactersPassword [Random](#)

\*\*\*\*\*

Password again

\*\*\*\*\*

**Public keys (SFTP/SCP/SSH)**☐ EnabledRSA keys: DSA keys: Ed25519 keys: ECDSA keys: 

4. Select the home directory where the data for this user will be stored. The default is usually the best option,

**Add a non-Windows user****Step 3: Select home folder**

Home folder:

☒ Create default home folder☐ Select non-default home folder

- > Home
- ▼ Public
  - > Documentation
  - Examples**

[Add root folder](#) [Add sub-folder](#)

5. Verify the information entered previously and select *Apply*:

Add User

### Add a non-Windows user

#### Step 4: Verify Details

Summary:


Username:  
HenryC

Full name:  
Henry Cole

Description:  
test

E-mail:  
test@test.com

Home folder:  
/Home/HenryC (C:\ProgramData\Enterprise Distributed Technologies\Complete

<  >

Previous

Apply

Cancel

Server Ready

Please click 'APPLY CHANGES' to send your changes to the server.

After that you may connect using any client application that supports any of the following protocols: **FTP, FTPS, SFTP, SCP, HTTP, HTTPS**.

You may like to view our guides describing how to connect using the free clients, [FileZilla](#) or [WinSCP](#).

CompleteAgent is a client application that is designed for performing robust, automatic uploads to CompleteFTP Enterprise

Ok

6. The server is now available for connection using an FTP, FTPS or SFTP client.

## How to connect

For detailed instruction on how to allow users to connect to this server over the internet, please refer to Step-by-step guide: [Accepting connections from the internet](#).

For detailed information on how to connect to this server using popular clients, see our tutorials:

- [Connecting with WinSCP](#)
- [Connecting With FileZilla](#)

# Getting started with Amazon EC2

This tutorial explains how to set up CompleteFTP on Amazon's Elastic Compute Cloud (Amazon EC2).

## Steps

1. Set up your EC2 instance (see [here](#)). You must create a security group (basically a firewall configuration) as per Amazon's instructions. At a minimum, you must allow connecting to your instance from your IP address using RDP (so you can connect remotely to your new instance).
2. Install CompleteFTP as usual- see [Getting Started](#). Your RDP client should allow you to copy the installer to your instance. Afterwards, you should be able to run the manager and connect to the server.
3. Configure your security group for CompleteFTP. You need to add rules that permit users of your server to access it. This means opening up ports for FTP/FTPS, SSH, HTTP and HTTPS (if you are using all these protocols) -see below.

Type <small>i</small>	Protocol <small>i</small>	Port Range <small>i</small>	Source <small>i</small>
Custom TCP Rule	TCP	21	0.0.0.0/0
SSH	TCP	22	0.0.0.0/0
HTTP	TCP	80	0.0.0.0/0
HTTPS	TCP	443	0.0.0.0/0
Custom TCP Rule	TCP	990	0.0.0.0/0

If you are using FTPS (explicit or implicit) then you also need open up a range of port for the data transfer connection, e.g.

Custom TCP Rule	TCP	10000 - 11000	0.0.0.0/0
-----------------	-----	---------------	-----------

In this case ports 10,000 to 11,000 are open, so you'll need to tell CompleteFTP that it should use these ports too, by setting the port range:

FTP / FTPS

FTP enabled

FTPS explicit enabled (standard

FTPS implicit enabled (legacy FT

Anonymous Logins

Advanced FTP/FTPS Settings

Port for FTP / FTPS explicit

21

Port to use for FTPS implicit

990

Listening IP addresses

All available

Folder listing format

Unix

Character set

Default - Unicode (UTF-8)

Passive (PASV) Transfer Settings

External IP address

Use external IP address fo

FTP/FTPS

FTPS only

Port Range

Minimum port number

10000

Maximum port number

11000

If you are not using FTPS then a port range need not be set. FTP (i.e. unencrypted) should work without setting the portrange since the firewall is able to work out which ports to open up by inspecting the (unencrypted) commands being sent on port 21.

# Connecting to CompleteFTP with Filezilla

This tutorial demonstrates how to connect to your **CompleteFTP** server using the popular FTP/SFTP client, **FileZilla**, which can be downloaded [here](#). **FileZilla** supports FTP, FTPS and SFTP.

There are two authentication methods: [password authentication](#) and [public key authentication](#).

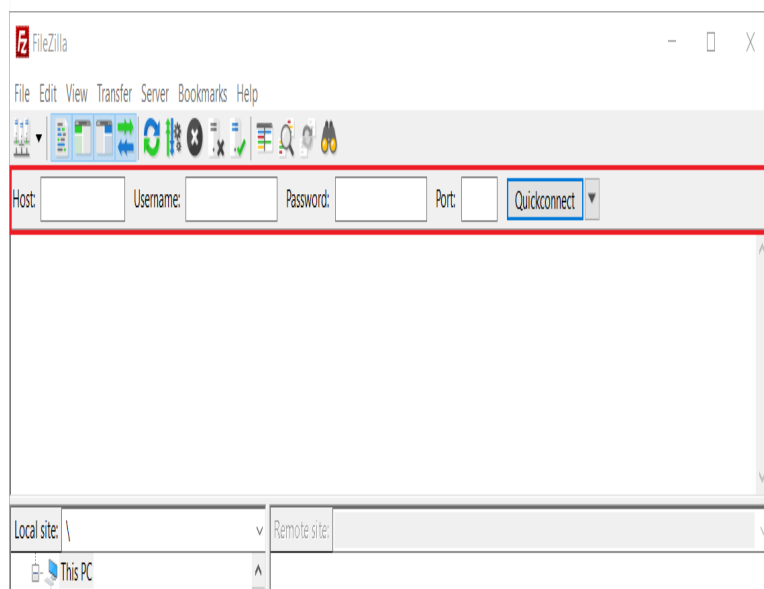
Before looking at this tutorial, make sure you have CompleteFTP installed, and a user created (see [here](#)).

## I. Connecting to CompleteFTP with Filezilla via password authentication

### Steps:

1. After installing FileZilla, open up the program.

Near the top of the FileZilla window, there is a *QuickConnect* Bar:



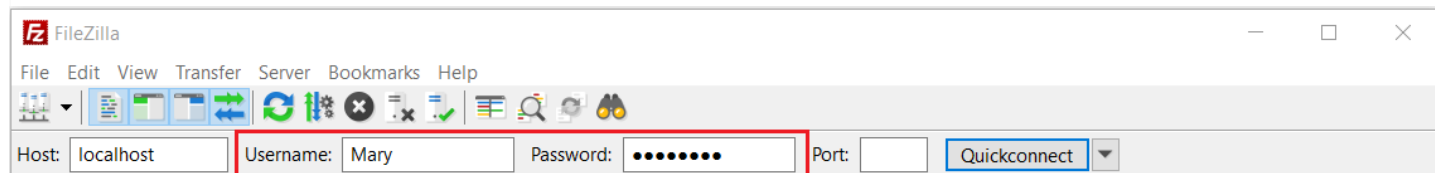
2. Enter your server's address into the *Host* Section (use `127.0.0.1`, or `localhost` if you are connecting from the same computer you are hosting the server on).



If you want to connect using FTPS (FTP over SSL), enter the address as 'ftps://server.org'.

If you want to connect using SFTP (FTP over SSH) enter the address as 'sftp://server.org'.

3. Enter your users's details into the *Username* and *Password* sections. These should be the same details you entered for this user when they were added to CompleteFTP using the **CompleteFTP** Manager (see [Getting Started With CompleteFTP](#))

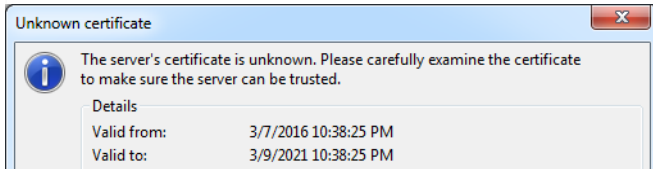
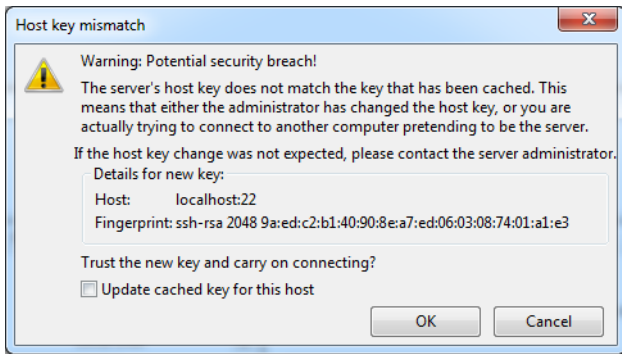


(you can leave the Port section blank, unless you require a specific port number)

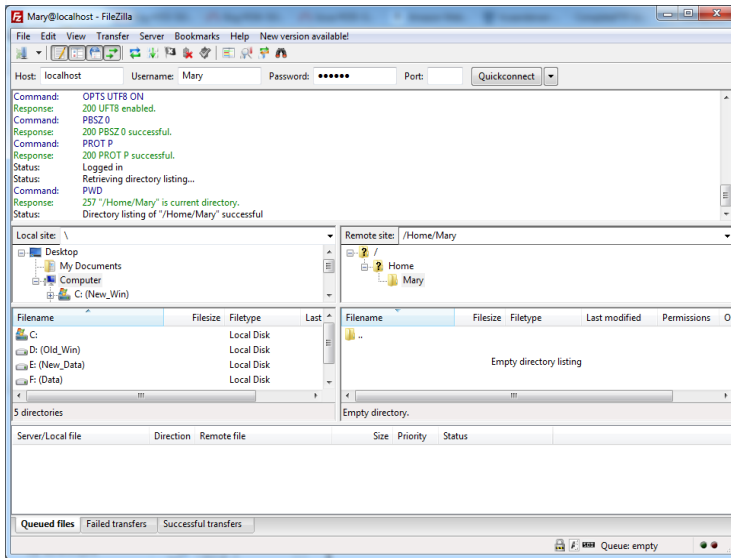
5. Press QuickConnect to connect to the server.

6. If you have chosen SFTP or FTPS, a warning about the host will appear. This warning is because you have never connected to this server before, and FileZilla does not have its host key (SFTP/SCP) or its certificate (encrypted FTP)- which identifies the server to clients - installed.

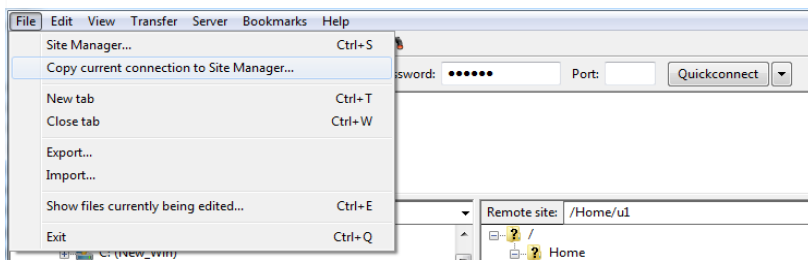
Click Yes to the warning to add the certificate or host key to your cache. Both warnings are displayed below - you will receive only one of them, depending on the protocol used.



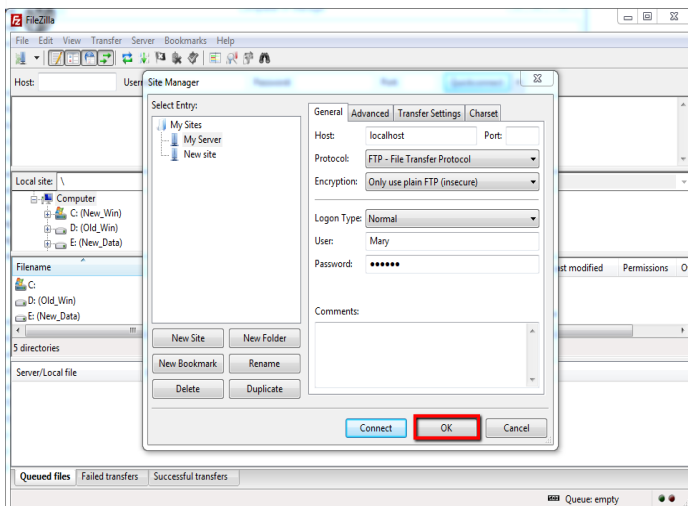
You should now be connected to CompleteFTP through FileZilla, using whatever protocol specified in your URL.



To add the server information to the site manager for later reference, select *Copy current connection to Site Manager* in the File menu;



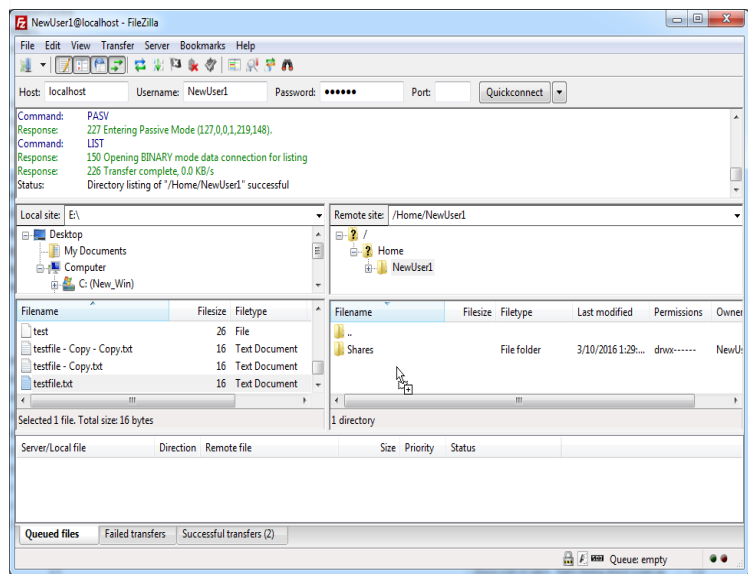
You will find all the important details automatically filled out, leaving you to press OK;



Instead of the Quick Connect, you can also use the site manager to select a site and then connect.



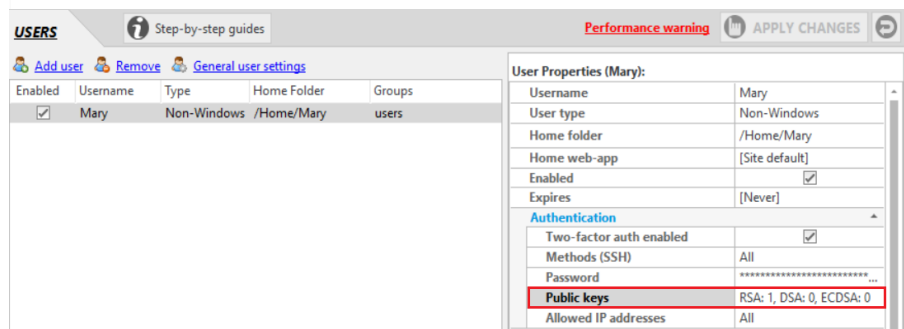
To add files to the server, drag and drop from the bottom left panel (your computer's directory) to the bottom right panel (the server's directory). Alternatively, drag and drop directly from your Computer to the right panel;



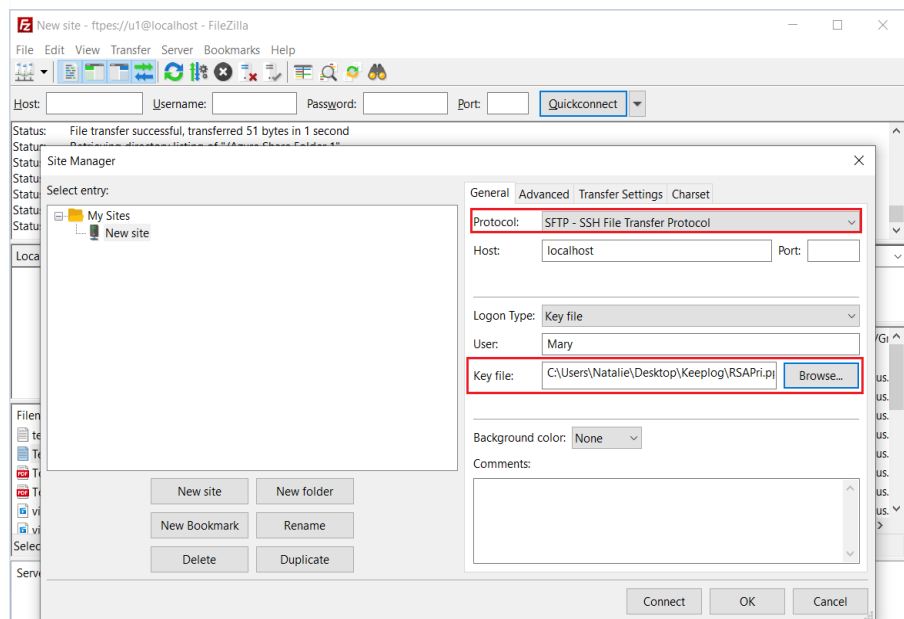
## II. Connecting to CompleteFTP with Filezilla via public key authentication

### Steps:

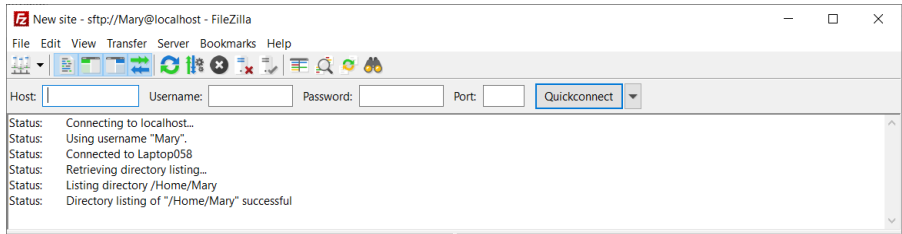
1. To connect with public key authentication requires a user to have a *keypair*, which consists of a private key and a public key. The private key is kept secret (usually with a password), while the public key can be given to anyone. For public key authentication with Filezilla, the private key is supplied to Filezilla while the public key is loaded into CompleteFTP to verify the user when they log in. The keypair can be generated in CompleteFTP as explained in [Setting up the user's keys](#). The public key is automatically loaded into CompleteFTP, while the private key is saved in a text file.



2. The private key must be loaded into Filezilla. Go to FileZilla/File/Site Manager and create a new site. Set the protocol to *SFTP - SSH File Transfer Protocol*, and the Logon Type to *Key file*. Load the private key that was just generated and saved using the Browse button. You will need to select "All files" in the file type dropdown, which is set to PPK by default (Filezilla's own private key format). Filezilla will ask to save the private key in PPK format.



Once the private key is loaded as a PPK file, you should be able to connect to CompleteFTP.



# Connecting to CompleteFTP with WinSCP

This tutorial demonstrates how to connect to your **CompleteFTP** server using the popular FTP/SFTP client, **WinSCP**, which can be downloaded [here](#). **WinSCP** supports FTP, FTPS, SFTP and SCP.

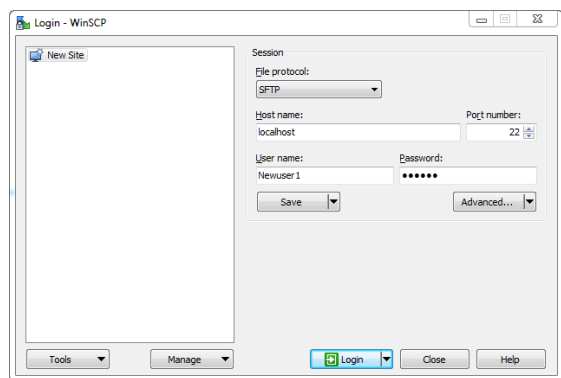
Before looking at this tutorial, make sure you have CompleteFTP installed, and a user created (*see [here](#)*).

## Steps

1. After installing and launching WinSCP, the Login window is shown, below.

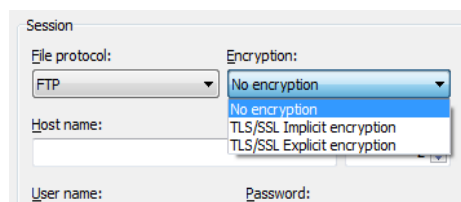
Enter the user name and password of a user that has been created in CompleteFTP previously.

If you are connecting from the same machine that CompleteFTP was installed on, you can use *127.0.0.1*, or *localhost* for the host name. Otherwise use the hostname of the server. If you are unsure what that is, ask your network administrator.



Now select the protocol to connect with - FTP, SCP or SFTP.

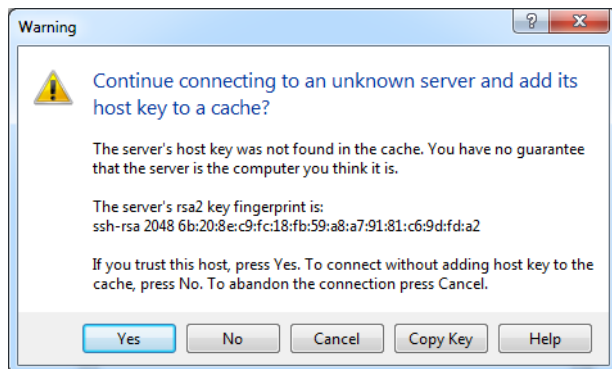
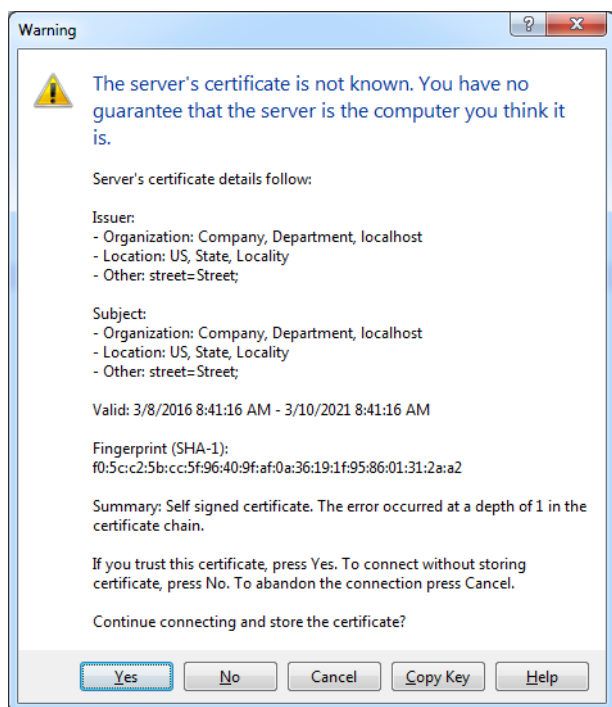
If FTP is chosen, use the Encryption drop-down menu to select what type of FTP you want. If you want plain, unencrypted FTP, select "No encryption". If you want FTPS, which is FTP over a secure connection, select "TLS Explicit encryption".



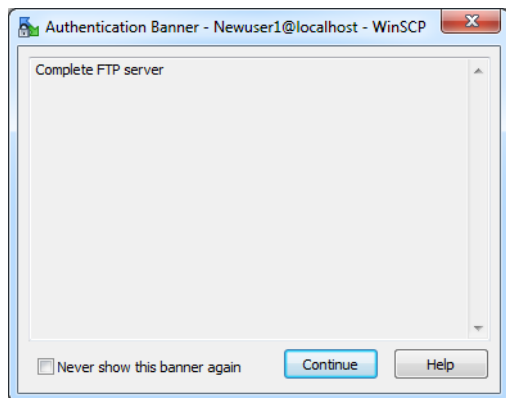
Then select "Login" to log into the server.

2. If you have chosen SFTP, SCP or an encrypted FTP, a warning about the host will appear. This warning is because you have never connected to this server before, and WinSCP does not have its host key (SFTP/SCP) or its certificate (encrypted FTP)- which identifies the server to clients - installed.

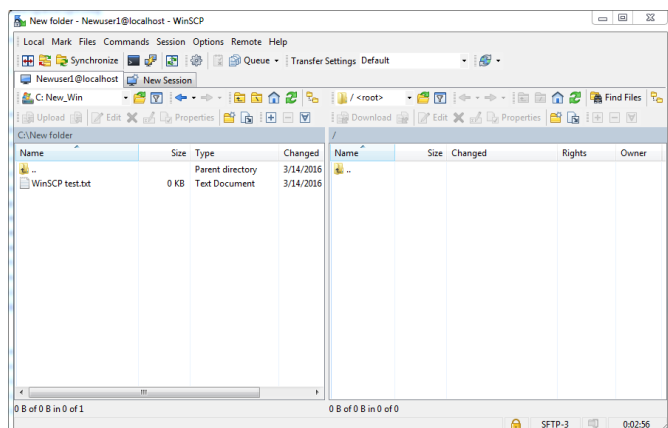
Click Yes to the warning to add the certificate or host key to your cache. Both warnings are displayed below - you will receive only one of them, depending on the protocol used.



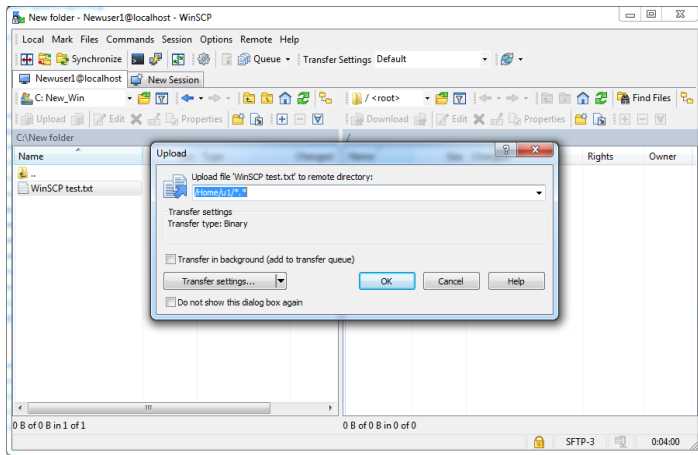
3. Next, the server's banner message will be displayed, as below. Select "Continue" to finish connecting. Check "Never show this banner again" to prevent it being displayed again.



4. You are now connected through WinSCP to the CompleteFTP server, and can start transferring.



To transfer files to the server, simply drag and drop from the left panel to the right. The left panel is your home computer and the right panel is your CompleteFTP server.



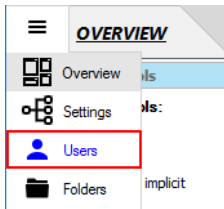
# Step-by-step guides

Step-by-step guides present instructions for common configuration tasks using lots of screenshots and only a little text. They're accessible directly from the CompleteFTP Manager allowing users to easily work through the instructions. Since some features are not available in lower editions, the following list indicates which edition each guide applies to.

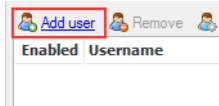
- [Add a non-Windows user](#) (All editions)
- [Add a Windows user](#) (Standard edition only)
- [Add a Windows user](#) (Professional edition and greater)
- [Set up public-key authentication for a user](#) (Standard edition and greater)
- [Allow all users from a Windows group to log in](#) (Professional edition and greater)
- [Authenticate users from a database](#) (Professional edition and greater)
- [Change the home folder of a user](#) (All editions)
- [Add a network folder](#) (Standard edition and greater)
- [Allow users to share a folder \(Method 1\)](#) (All editions)
- [Allow users to share a folder \(Method 2\)](#) (Standard edition and greater)
- [Allow users to share a folder \(Method 3\)](#) (Professional edition and greater)
- [Block connections from specific IP addresses \(blacklisting\)](#) (Enterprise MFT)
- [Accept connections only from specific IP addresses \(whitelisting\)](#) (Professional edition and greater)
- [Accept connections only from specific users at specific IP addresses](#) (Professional edition and greater)
- [Restrict some users to connect only from specific IP addresses](#) (Professional edition and greater)
- [E-mail notification when a file is uploaded](#) (Professional edition and greater)
- [Move a file to another directory after it's been uploaded](#) (Professional edition and greater)
- [Add a scheduled task](#) (Professional edition and greater)
- [Accepting connections from the internet](#) (All editions)
- [Share a file using CompleteBox](#) (Professional edition and greater)
- [Customize the web home page](#) (Professional edition and greater)
- [Host a web-app in CompleteFTP](#) (Professional edition and greater)
- [Add my own authentication method](#) (Enterprise MFT)
- [Request a CA certificate](#) (All editions)
- [Install a CA certificate](#) (All editions)
- [Activate CompleteFTP free](#) (Free edition only)
- [Activate the paid editions](#) (Standard edition and greater)
- [Set up a cluster](#) (Enterprise MFT only)

# Add a non-Windows user

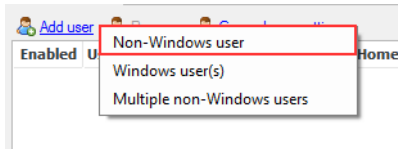
1. Select the 'Users' panel from the side-bar menu.



2. Click the 'Add user' link at the top of the panel. Free edition users: skip to step 4.



3. Select 'Non-Windows user' from the menu.



4. Enter the username and, optionally, enter any other information in the appropriate dialog boxes. Click 'Next'.

A screenshot of the 'Add User' dialog box, Step 1: Set Username. The dialog has a title bar 'Add User' and a subtitle 'Add a non-Windows user'. It contains four text input fields: 'Username' (filled with 'JoeSmith'), 'Full name (optional)' (filled with 'Joe Smith'), 'Description (optional)' (empty), and 'E-mail (optional)' (filled with 'joesmith@domain.com'). At the bottom, there are three buttons: 'Previous', 'Next' (highlighted with a red rectangle), and 'Cancel'.

5. Enter the password or click 'Random' to generate one randomly. In Standard Edition (or higher) you can also add public keys. Click 'Next'.

A screenshot of the 'Add User' dialog box, Step 2: Set Authentication. The dialog has a title bar 'Add User' and a subtitle 'Add a non-Windows user'. It contains two main sections: 'Password' and 'Public keys (SFTP/SCP/SSH)'. The 'Password' section has a 'Password' field with a 'Random' link, a 'Password again' field, and a 'Show characters' checkbox. The 'Public keys' section has an 'Add public key(s)' button. At the bottom, there are three buttons: 'Previous', 'Next' (highlighted with a red rectangle), and 'Cancel'.

Note for Standard, Professional and Enterprise MFT users: if the user is going to be authenticating with a public key then give them a random password

6. Set the home directory for this user by creating a new one (default) or selecting an existing one. Click 'Next'.

Add User

### Add a non-Windows user

#### Step 3: Select home folder

Home folder:

☒ Create default home folder

/Home/JoeSmith (C:\ProgramData\Enterprise Distributed Technologies\Comp)

☐ Select non-default home folder

> Home

> Public

[Add root folder](#) [Add sub-folder](#)

Previous **Next** Cancel

7. Click the 'Apply' button.

Add User

### Add a non-Windows user

#### Step 4: Verify Details

Summary:

Username:  
JoeSmith

Full name:  
(none)

Description:  
(none)

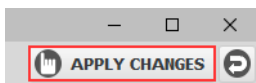
E-mail:  
(none)

Home folder:  
/Home/JoeSmith (C:\ProgramData\Enterprise Distributed Technologies\Comp)

< >

Previous **Apply** Cancel

8. Finally, click the 'Apply changes' button at the top right of the CompleteFTP Manager.



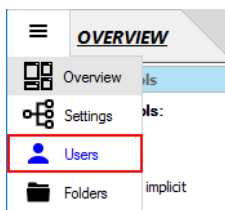
The user is now successfully added on the CompleteFTP server and displayed in the user list. Try it out by logging in from a client application.

<a href="#">Add user</a> <a href="#">Remove</a> <a href="#">General user settings</a>					<b>User Properties (JoeSmith):</b>	
Enabled	Username	Type	Home Folder	Full Name	Username	JoeSmith
<input checked="" type="checkbox"/>	JoeSmith	Non-Windows	/Home/JoeSmith	Joe Smith	User type	Non-Windows
					Home folder	/Home/JoeSmith
					Enabled	<input checked="" type="checkbox"/>

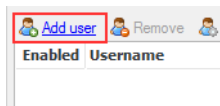


# Add a Windows user (Standard edition only)

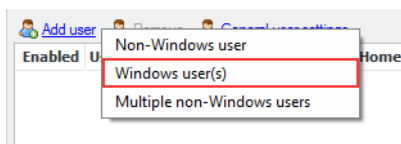
1. Select the 'Users' panel from the side-bar menu.



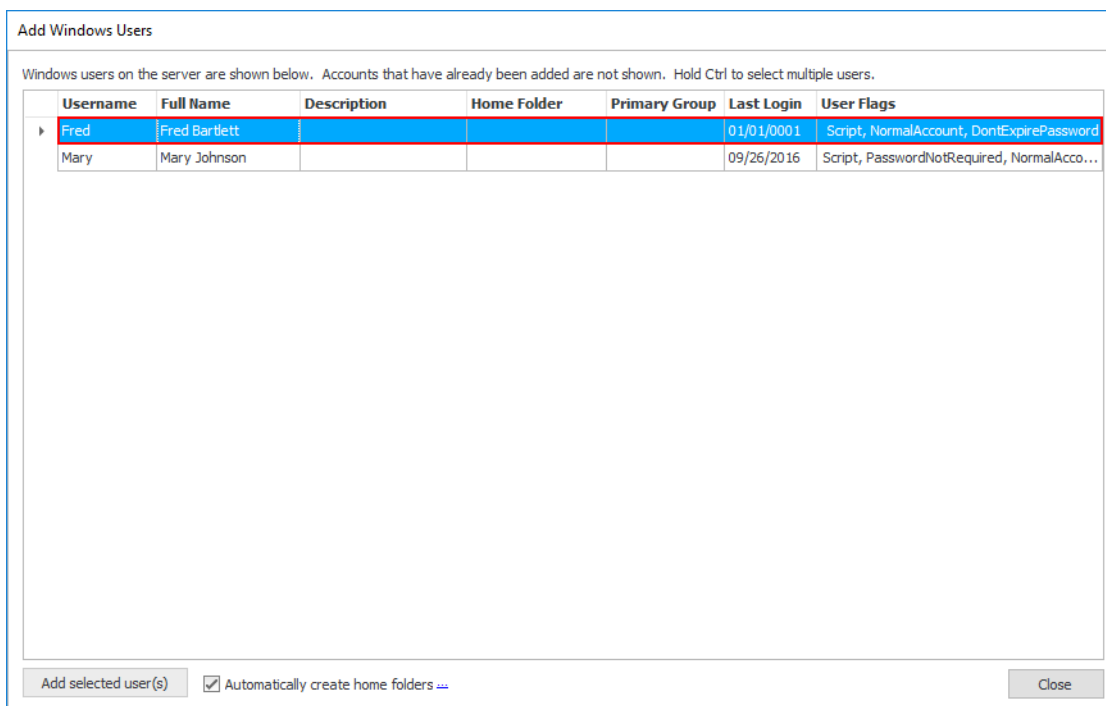
2. Click the 'Add user' link at the top of the panel.



3. Select 'Windows user(s)' from the menu.



4. In the 'Add Windows Users' dialog box, select the user, e.g. 'Fred', to be added from the list.



5. Select the way you want the user's home folder to be created. There are two options here:
  - (1) Automatically create the user's home folder by checking the 'Automatically create home folders' checkbox (It's selected by default).

☒ Automatically create home folders

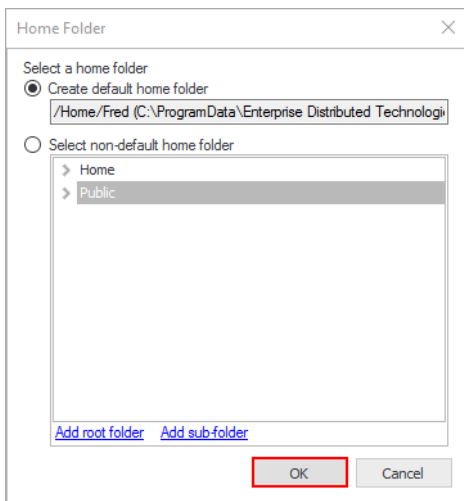
- (2) Manually create the user's home folder by un-checking the 'Automatically create home folders' checkbox.

☐ Automatically create home folders

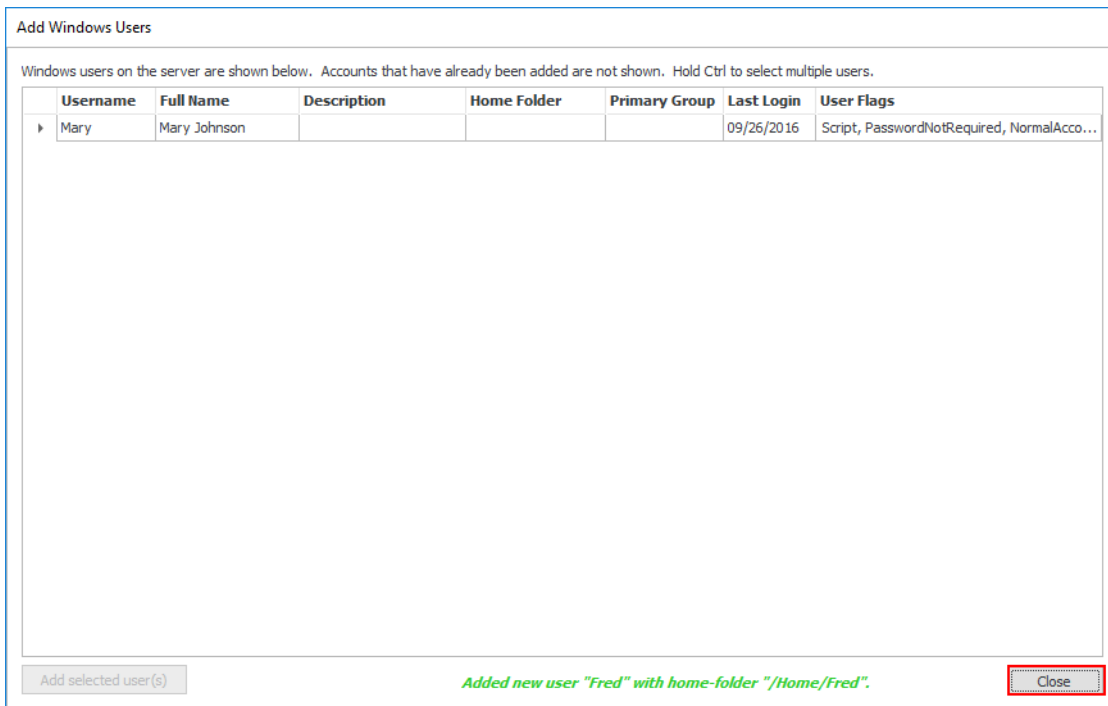
6. Click the 'Add selected user(s)' button. If you checked 'Automatically create home folders' then skip to step 8.

Add selected user(s)

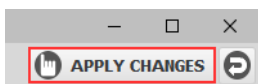
7. In 'Home Folder' dialog box, select the user's home folder (default or non-default one) manually. Then click 'OK'.



8. Click the 'Close' button if you don't want to add another Windows user.



9. Click the 'Apply changes' button at the top right of the CompleteFTP Manager.

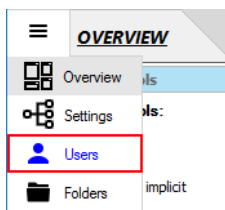


The users are now successfully added to the CompleteFTP server and displayed in the user list.

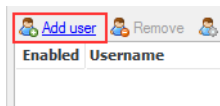


# Add a Windows user (Professional edition and greater)

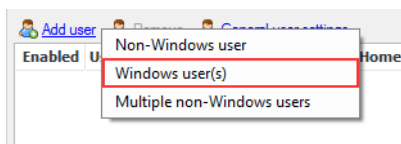
1. Select the 'Users' panel from the side-bar menu.



2. Click the 'Add user' link at the top of the panel.



3. Select 'Windows user(s)' from the menu.



4. In the 'Add Windows Users' dialog box, you can choose either to:

(1) Add server's local users.

A screenshot of the 'Add Windows Users' dialog box. The title bar says 'Add Windows Users'. Inside, there's a message: 'Windows users on the server are shown below. Accounts that have already been added are not shown. Hold Ctrl to select multiple users.' Below this is a table with columns 'Username', 'Full Name', 'Login', and 'User Flags'. An orange callout box with a red arrow points to a button labeled 'List server's local users'. Below this button is the text 'OR'. Then there are input fields for 'Domain name', a checkbox for 'Login when querying domain', an input field for 'User-name filter', and a 'List size-limit' dropdown set to '10000'. At the bottom of this section is a button labeled 'List domain users'. At the very bottom of the dialog box are two buttons: 'Add selected user(s)' and 'Close'. There is also a checkbox labeled 'Automatically create home folders ...'.

OR (2) add domain users. Enter the name of the domain.

**Add Windows Users**

Windows users on the server are shown below. Accounts that have already been added are not shown. Hold Ctrl to select multiple users.

Username	Full Name	Description	Home Folder	Primary Gro...	Last Login	User Flags
<div style="border: 1px solid gray; padding: 10px; margin: 10px auto; width: 300px;"> <div style="text-align: center; margin-bottom: 10px;">List server's local users</div> <div style="text-align: center; margin-bottom: 10px;"><b>OR</b></div> <div>Domain name <input type="text" value="domain_name"/></div> <div><input type="checkbox"/> Login when querying domain</div> <div>User-name filter <input type="text"/></div> <div>List size-limit <input type="text" value="10000"/></div> <div style="border: 1px solid red; padding: 2px; text-align: center; margin-top: 5px;">List domain users</div> </div>						

If you want to add a Windows domain user, enter the domain name then click 'List domain users'

Add selected user(s) ☒ Automatically create home folders ... Close

Note: for domains with a very large number of users, the size of the list can be restricted and/or a filter can be provided to reduce the number of users listed.

5. In the 'Add Windows Users' dialog box, select the user, e.g. 'Fred', to be added from the list.

**Add Windows Users**

Windows users on the server are shown below. Accounts that have already been added are not shown. Hold Ctrl to select multiple users.

Username	Full Name	Description	Home Folder	Primary Group	Last Login	User Flags
Fred	Fred Bartlett				01/01/0001	Script, NormalAccount, DontExpirePassword
Mary	Mary Johnson				09/26/2016	Script, PasswordNotRequired, NormalAcco...

Add selected user(s) ☒ Automatically create home folders ... Close

6. Select the way you want the user's home folder to be created. There are two options here:

(1) Automatically create the user's home folder by checking the 'Automatically create home folders' checkbox (It's selected by default).

☒ Automatically create home folders ...

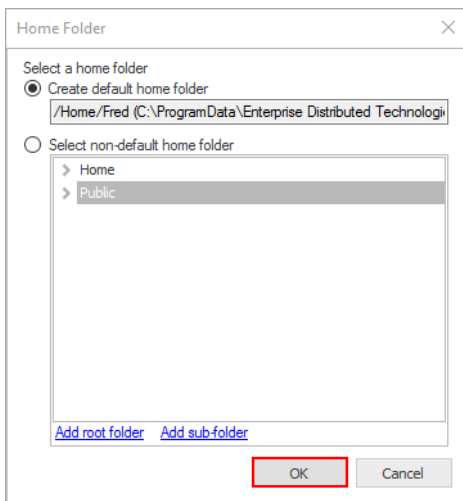
(2) Manually create the user's home folder by un-checking the 'Automatically create home folders' checkbox.

☐ Automatically create home folders ...

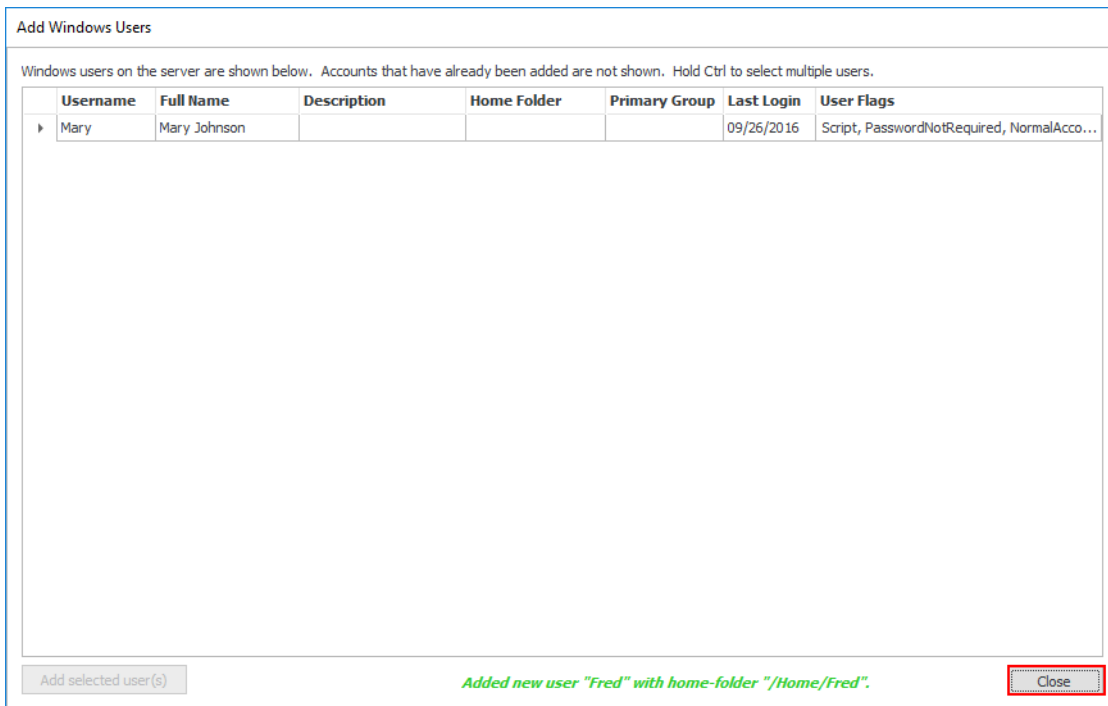
7. Click the 'Add selected user(s)' button, however if you checked 'Automatically create home folders' then skip to step 9.

Add selected user(s)

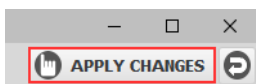
8. In 'Home Folder' dialog box, select the user's home folder (default or non-default one) manually. Then click 'OK'.



9. Click the 'Close' button.



10. Click the 'Apply changes' button at the top right of the CompleteFTP Manager.



The users are now successfully added to the CompleteFTP server and displayed in the user list.



# Set up public-key authentication for a user

Before getting started, ensure that the user who you want to set up the public-key authentication for, has already been created. In this guide, the 'JoeSmith' user will be used.

To set up public-key authentication for a user, you need to:

- I. Enable public key authentication.
- II. Set up the user's keys.

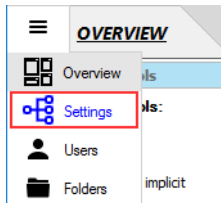
Here are the detailed steps:

## I. Enable public key authentication

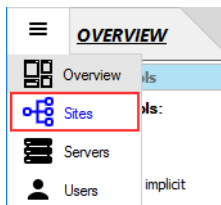
Public authentication needs to be enabled in two places as shown below. But if you are using the Standard/Professional Edition, you just need to go through the first one.

### 1. Enable public key authentication method in the 'Settings'/'Sites' panel.

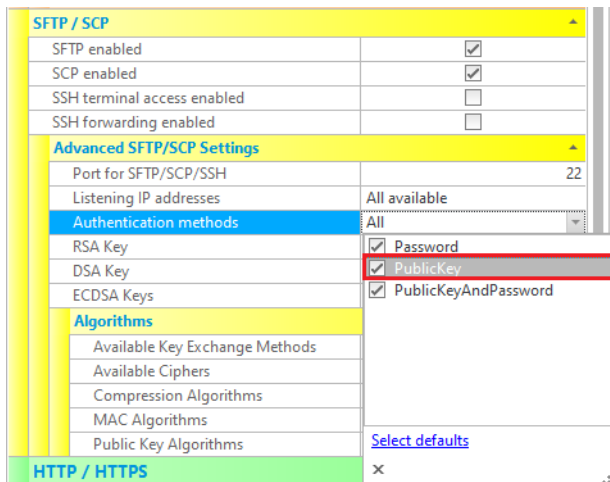
- a. From the side-bar menu, select the 'Settings' panel (Standard/Professional Edition)



OR the 'Sites' panel (Enterprise MFT).

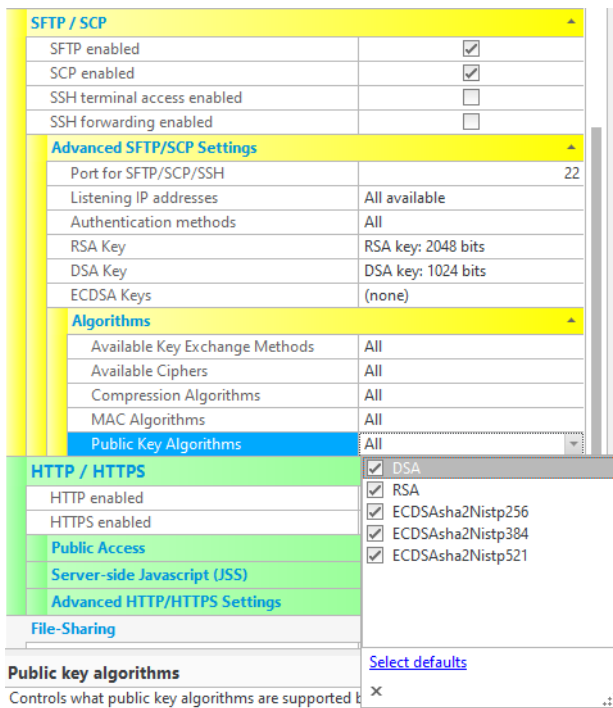


- b. Enable 'PublicKey' authentication method in SFTP settings (It is enabled by default).



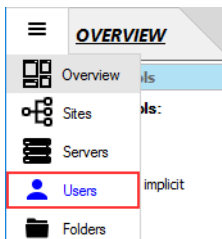
- c. Set the public key algorithms (optional).

Note: public key authentication uses either the RSA or DSA algorithm. Both of them are enabled by default.



2. Enabled public key authentication method in the user's properties (Enterprise MFT only).

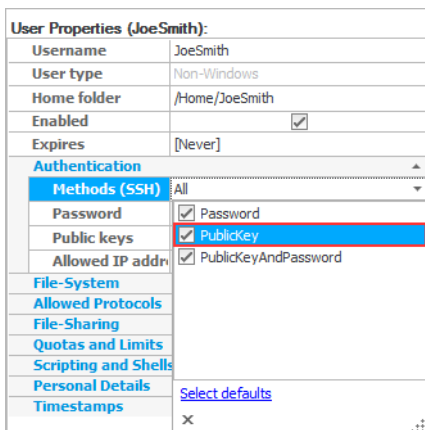
- a. Select the 'Users' panel from the side-bar menu.



- b. In the user list, select the user who you want to set up the public key authentication for.

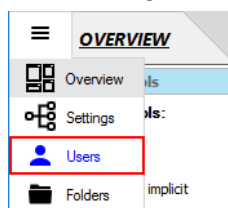
Add user Remove General user settings					User Properties (JoeSmith):	
Enabled	Username	Type	Home Folder	Groups	Username	JoeSmith
<input checked="" type="checkbox"/>	FredJones	Non-Windows	/Home/FredJones	users	User type	Non-Windows
<input checked="" type="checkbox"/>	JoeSmith	Non-Windows	/Home/JoeSmith	users	Home folder	/Home/JoeSmith
					Enabled	<input checked="" type="checkbox"/>

- c. In the 'User Properties' window, enable 'PublicKey' authentication method for this user in 'Authentication' section.



## II. Set up the user's keys

1. Select the 'Users' panel from the side-bar menu.



2. In the user list, select the user you want to set up the public key authentication for.

<a href="#">Add user</a> <a href="#">Remove</a> <a href="#">General user settings</a>					<b>User Properties (JoeSmith):</b>	
Enabled	Username	Type	Home Folder	Groups	Username	JoeSmith
<input checked="" type="checkbox"/>	FredJones	Non-Windows	/Home/FredJones	users	User type	Non-Windows
<input checked="" type="checkbox"/>	JoeSmith	Non-Windows	/Home/JoeSmith	users	Home folder	/Home/JoeSmith
					Enabled	<input checked="" type="checkbox"/>

- Open the 'Manage Public Keys' dialog box by clicking the ellipsis button (...) on the right of "Public keys" in the 'User Properties' window.

<b>User Properties (JoeSmith):</b>	
Username	JoeSmith
User type	Non-Windows
Home folder	/Home/JoeSmith
Enabled	<input checked="" type="checkbox"/>
Expires	[Never]
<b>Authentication</b>	
Password	*****
Public keys	No keys ...

- Click 'Generate a new keypair for this user and add it to the list' to create a new keypair. You can also use "Import a public key and add it to the list", to import existing RSA or DSA public keys for this user. Public keys can also be deleted or exported.

Manage Public Keys

Public keys:

Type
Fingerprint
Encoded

[Import a public key and add it to the list](#)  
Remove currently selected key from the list  
Export the selected key in OpenSSH format  
Export the selected key in SECSH format  
[Generate a new keypair for this user and add it to the list](#)

OK
Cancel

- In the 'Generate SSH Keypair' dialog box, select the desired SSH key type and set the key's size if needed (RSA only). Then click 'OK'.

Generate SSH Keypair

☒ RSA Keypair
2,048 bits
☐ ECDSA P-256 Keypair
☐ ECDSA P-384 Keypair
☐ ECDSA P-521 Keypair
☐ DSA Keypair

OK
Cancel

- In 'Save User's Private Key' dialog box, navigate to the directory where you want to save the private key securely. Then enter the file's name and click 'Save'.

Save User's Private Key

OneDrive

Organize
New folder

Dropbox
OneDrive
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
DATA (D:)

No items match your search.

File name:
user\_private\_key

Save as type:
All Files (\*.\*)

Hide Folders
Save
Cancel

- Enter the key's password manually or click 'Random' to generate one randomly, then click 'OK'.



Enter Password

☐ Show characters

Password [Random](#)  
 \*\*\*\*\*

Password again  
 \*\*\*\*\*

8. Now, the new public key is added to the list. Click 'OK' to close the 'Manage Public Keys' dialog.

Manage Public Keys

Public keys: (any of these may be used by the user for authentication)

Type	Fingerprint	Encoded
▶ RSA	89:90:0e:2f:31:b0:00:4a:14:d8:d9:fa:8b:e...	AAAAB3NzaC1yc2EAAAABIwAAAQBAACFAAB...

[Import a public key and add it to the list](#)  
[Remove currently selected key from the list](#)  
[Export the selected key in OpenSSH format](#)  
[Export the selected key in SECSH format](#)  
[Generate a new keypair for this user and add it to the list](#)

9. Click the 'Apply changes' button at the top right of the CompleteFTP Manager.

Now, the new keypair has been generated successfully and the public key has been registered with the CompleteFTP server.

[Add user](#)
[Remove](#)
[General user settings](#)

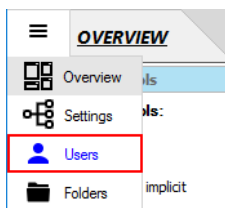
Enabled	Username	Type	Home Folder	Groups
<input checked="" type="checkbox"/>	FredJones	Non-Windows	/Home/FredJones	users
<input checked="" type="checkbox"/>	JoeSmith	Non-Windows	/Home/JoeSmith	users

**User Properties (JoeSmith):**

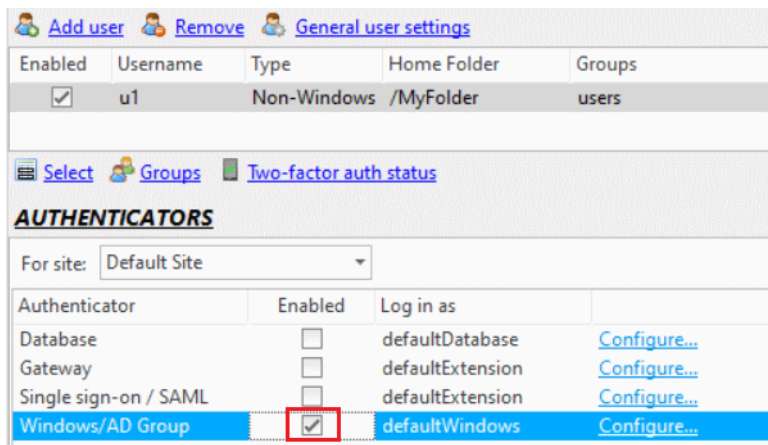
Username	JoeSmith
User type	Non-Windows
Home folder	/Home/JoeSmith
Home web-app	[Site default]
Enabled	<input checked="" type="checkbox"/>
Expires	[Never]
<b>Authentication</b>	
Methods (SSH)	All
Password	*****
Public keys	RSA: 1, DSA: 0, ECDSA: 0

# Allow all users from a Windows group to log in

1. Select the 'Users' panel from the side-bar menu.

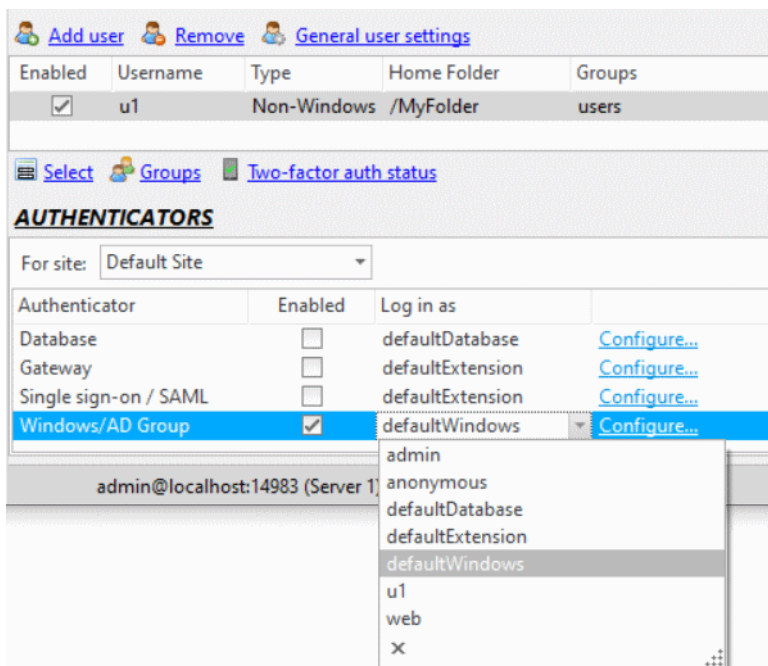


2. In the 'Authenticators' section, enable the Automatic Windows authentication method by checking its 'Enabled' checkbox.

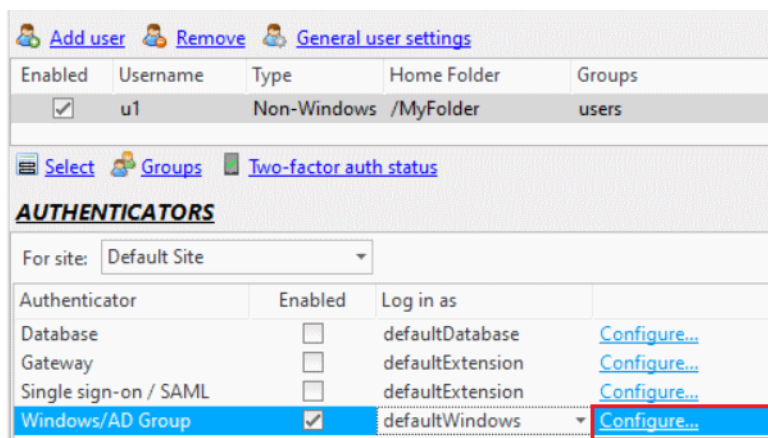


3. Set the 'log-in-as' user for this authentication method via the drop-down list (optional).

Note: users connecting via this method are subject to the properties of the 'log-in-as' user. By default, the 'defaultWindows' user is selected



4. Click the 'Configure' link to the right, in order to enable logins for users who belong to a particular Windows group.



5. Click 'Add' and enter the name of either an Active Directory (AD) or a local group in the first column. Then click 'OK'.

Windows/AD Group Configuration

Add allowed AD or Windows group

Remove

AD or Windows Group	CompleteFTP Groups	Log-In-As User	Home Folder

AD/Windows Group

Name of the AD or Windows group. AD groups are usually of the pattern domain\groupname and Windows groups are specified just be groupname. If you're unsure about the exact name of an AD group then enter what you think it should be have the user attempt to log in. If the user entered the correct user-name and password, but was disallowed because they're not in an allowed group then a list of all the groups that they're a member of will be shown in the logs (including the realtime logs).

OK

Cancel

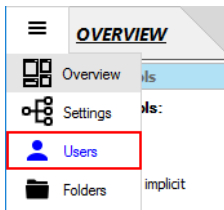
6. Finally, click the 'Apply changes' button at the top right of the CompleteFTP Manager.

APPLY CHANGES

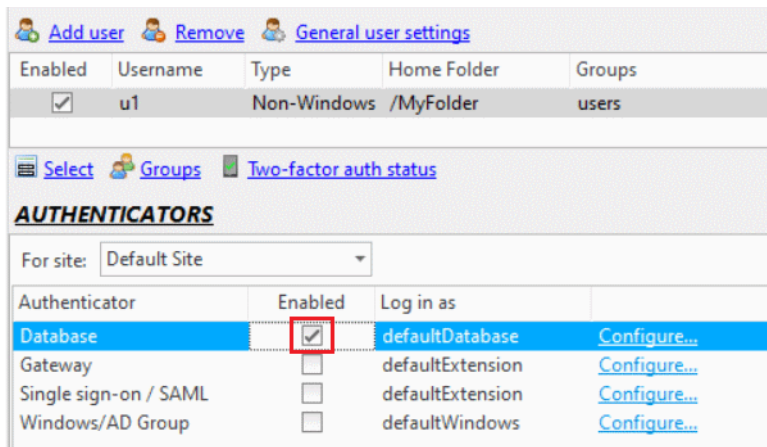
# Authenticate users from a database

Before starting with this guide, you need to create a data-source corresponding to the database-type you will select in step 6 of this guide. This data-source may be a database table, spreadsheet or file, containing a list of usernames, passwords or password-hashes, and optionally home directories, groups, password-salts, RSA and/or DSA public keys.

1. Select the 'Users' panel from the side-bar menu.

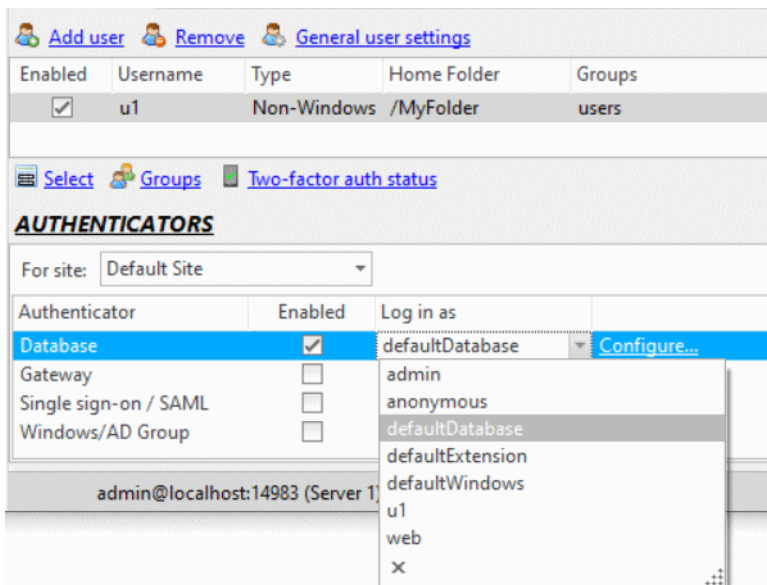


2. In the 'Authenticators' section, enable Database authentication method by checking its 'Enabled' checkbox.

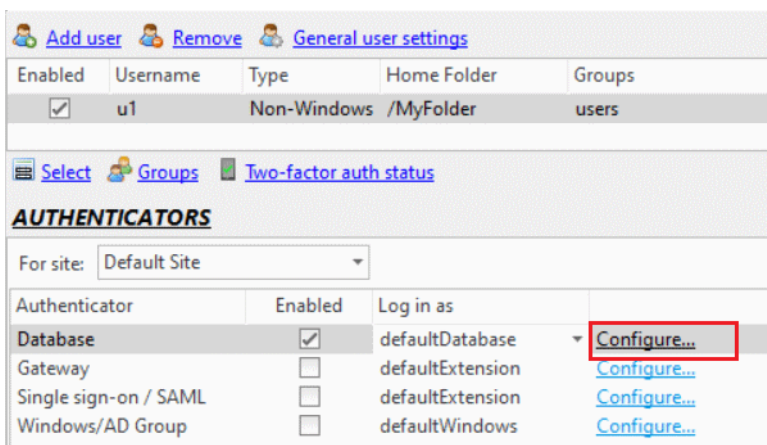


3. Set the 'log-in-as' user for this authentication method via the 'Log in as' drop-down list (optional).

Note: users connecting via this method are subject to the properties of the 'log-in-as' user. By default, the 'defaultDatabase' user is selected.



4. Click the 'Configure' link to open the 'Database Configuration' dialog box.



5. Select the desired database type from the 'Database type' drop-down list. Note that new database types can be added (see [here](#)).

Database Configuration

Database type

SQL Server Compact 4.0

Odbc

OleDb

Oracle

SQL Server

SQL Server Compact 4.0

Test connection

Query for authenticating a user

Given a username (specified by %UserName%), it must return the password or password-hash. It may also return the Windows path of the user's home-folder; a comma-separated list of groups; the password salt; an RSA public key and a DSA public key. E.g. `SELECT Password, HomeFolderPath, Groups, Salt, RSAKey, DSAKey FROM Users WHERE UserName='%UserName%'`

Test login

User-name

Password

Test login

Password encoding

☒ Plain text ☐ MD5/SHA1 hash

Database type

The type of database to connect to when authenticating. This also determines the format of the connection-string.

By default, three specific database types (Oracle, SQL Server and SQL Server Compact) and two generic data-source types (ODBC and OLEDB) are supported. Support for new database types may be also added (see below).

If an ODBC data-source is being used then a driver for the specific data-source must be installed and the specific data-

OK Cancel

6. Enter the connection string into the 'Connection-string' text box. For connection string examples, click the 'examples' link or visit the website [connectionstrings.com](http://connectionstrings.com).

Database Configuration

Database type

SQL Server Compact 4.0

Connection-string [examples](#)

Data Source=E:\Data\myDatabase.sdf

Test connection

Query for authenticating a user

Given a username (specified by %UserName%), it must return the password or password-hash. It may also return the Windows path of the user's home-folder; a comma-separated list of groups; the password salt; an RSA public key and a DSA public key. E.g. `SELECT Password, HomeFolderPath, Groups, Salt, RSAKey, DSAKey FROM Users WHERE UserName='%UserName%'`

Test login

User-name

Password

Test login

Password encoding

☒ Plain text ☐ MD5/SHA1 hash

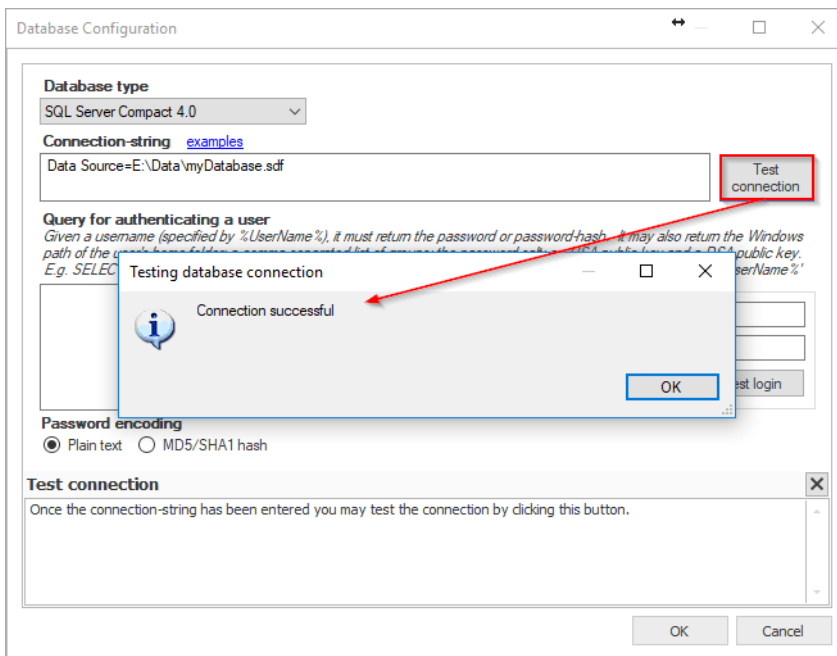
Query for authenticating a user

This is the SQL query that retrieves the user's authentication data from the data-source. It must return at least the password or password-hash. It may also optionally define a Windows home-folder; a comma-separated list of groups to which the user belongs; the password 'salt'; an RSA public key and a DSA public key.

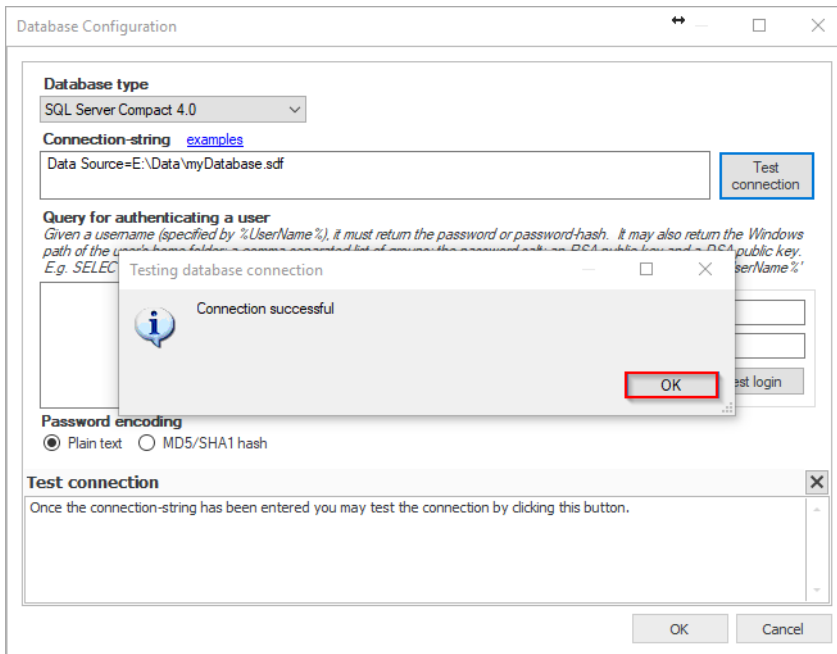
The password or password-hash must be the first field returned. It is far more secure to store password hashes rather than passwords, and even better to use a 'salt' when hashing (see the fourth field below).

OK Cancel

7. Click the 'Test connection' button to verify that it works.



8. Click the 'OK' button.



9. Enter the SQL query for retrieving the user's password/password-hash into the 'Query for authenticating a user' text box. It may also optionally define a Windows home-folder, a comma-separated list of groups to which the user belongs and the password-salt.

Database Configuration

Database type  
SQL Server Compact 4.0

Connection-string [examples](#)  
Data Source=E:\Data\myDatabase.sdf Test connection

**Query for authenticating a user**  
Given a username (specified by %UserName%), it must return the password or password-hash. It may also return the Windows path of the user's home-folder; a comma-separated list of groups; the password salt; an RSA public key and a DSA public key. E.g. `SELECT Password, HomeFolderPath, Groups, Salt, RSAKey, DSAKey FROM Users WHERE UserName=%UserName%`

`SELECT Password, HomeFolderPath FROM [Users] WHERE UserName=%UserName%`

Test login  
User-name   
Password  Test login

Password encoding  
☒ Plain text ☐ MD5/SHA1 hash

**Query for authenticating a user**  
This is the SQL query that retrieves the user's authentication data from the data-source. It must return at least the password or password-hash. It may also optionally define a Windows home-folder; a comma-separated list of groups to which the user belongs; the password 'salt'; an RSA public key and a DSA public key.  
The password or password-hash must be the first field returned. It is far more secure to store password hashes rather than passwords, and even better to use a 'salt' when hashing (see the fourth field below).

OK Cancel

10. To test, enter the user-name and password of a user who exists in the user database, then click 'Test login' to verify that the SQL query works.

Database Configuration

Database type  
SQL Server Compact 4.0

Connection-string [examples](#)  
Data Source=E:\Data\myDatabase.sdf Test connection

**Query for authenticating a user**  
Given a username (specified by %UserName%), it must return the password or password-hash. It may also return the Windows path of the user's home-folder; a comma-separated list of groups; the password salt; an RSA public key and a DSA public key. E.g. `SELECT Password, HomeFolderPath, Groups, Salt, RSAKey, DSAKey FROM Users WHERE UserName=%UserName%`

Testing database connection  
User authenticated successfully  
Home folder is C:\Temp\maryjohnson OK

Test login  
User-name   
Password  Test login

**Test login**  
Once the connection-string and user-query have been entered you may test your settings by entering a user-name and a password and clicking Test.

OK Cancel

11. Click the 'OK' button.

Database Configuration

Database type  
SQL Server Compact 4.0

Connection-string [examples](#)  
Data Source=E:\Data\myDatabase.sdf Test connection

**Query for authenticating a user**  
Given a username (specified by %UserName%), it must return the password or password-hash. It may also return the Windows path of the user's home-folder; a comma-separated list of groups; the password salt; an RSA public key and a DSA public key. E.g. `SELECT Password, HomeFolderPath, Groups, Salt, RSAKey, DSAKey FROM Users WHERE UserName=%UserName%`

Testing database connection  
User authenticated successfully  
Home folder is C:\Temp\maryjohnson OK

Test login  
User-name   
Password  Test login

**Test login**  
Once the connection-string and user-query have been entered you may test your settings by entering a user-name and a password and clicking Test.

OK Cancel

12. In the 'Password encoding' section, you can select either 'Plain text' option

The screenshot shows the 'Database Configuration' dialog box. The 'Database type' is set to 'SQL Server Compact 4.0'. The 'Connection-string' is 'Data Source=E:\Data\myDatabase.sdf'. The 'Query for authenticating a user' is 'SELECT Password, HomeFolderPath, Groups, Salt, RSAKey, DSAKey FROM Users WHERE UserName= '%UserName%'''. The 'Password encoding' section has two radio buttons: 'Plain text' (selected) and 'MD5/SHA1 hash'. An orange callout box points to the 'Plain text' radio button with the text: 'Select 'Plain text' if the password in the user database is stored in plain text'. The 'Test login' section has 'User-name' as 'myJohnson' and 'Password' as '\*\*\*\*\*'. The 'Test login' button is highlighted with a blue dashed border. The 'OK' and 'Cancel' buttons are at the bottom.

OR 'MD5/SHA1 hash' option.

The screenshot shows the 'Database Configuration' dialog box. The 'Database type' is set to 'SQL Server Compact 4.0'. The 'Connection-string' is 'Data Source=E:\Data\myDatabase.sdf'. The 'Query for authenticating a user' is 'SELECT Password, HomeFolderPath, Groups, Salt, RSAKey, DSAKey FROM Users WHERE UserName= '%UserName%'''. The 'Password encoding' section has two radio buttons: 'Plain text' and 'MD5/SHA1 hash' (selected). An orange callout box points to the 'MD5/SHA1 hash' radio button with the text: 'Select 'MD5/SHA1 hash' if the password is stored as an MD5 or SHA1 hash'. The 'Test login' section has 'User-name' as 'myJohnson' and 'Password' as '\*\*\*\*\*'. The 'Test login' button is highlighted with a blue dashed border. The 'Database user configuration' section is expanded, showing a list of steps: (1) selecting the database type, (2) entering the connection-string of the user database, and (3) entering a query to retrieve the authentication data from that database. The 'OK' and 'Cancel' buttons are at the bottom.

13. Click the 'OK' button to close the 'Database Configuration' dialog box.



Database Configuration

**Database type**  
SQL Server Compact 4.0

**Connection-string** [examples](#)  
Data Source=E:\Data\myDatabase.sdf

**Test connection**

**Query for authenticating a user**  
*Given a username (specified by %UserName%), it must return the password or password-hash. It may also return the Windows path of the user's home-folder; a comma-separated list of groups; the password salt; an RSA public key and a DSA public key. E.g. SELECT Password, HomeFolderPath, Groups, Salt, RSAKey, DSAKey FROM Users WHERE UserName=%UserName%'*

SELECT Password, HomeFolder FROM [Users] WHERE UserName=%UserName%'

**Test login**  
User-name: myJohnson  
Password: \*\*\*\*\*  
**Test login**

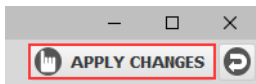
**Password encoding**  
☒ Plain text ☐ MD5/SHA1 hash

**Database user configuration**  
Configuration of the database users feature involves:  
(1) selecting the database type  
(2) entering the connection-string of the user database  
(3) entering a query to retrieve the authentication data from that database.

If the database type you need isn't listed then find its ADO.NET driver (e.g. MySQL Connector/NET). After you restart the

**OK** **Cancel**

14. Finally, click the 'Apply changes' button at the top right of the CompleteFTP Manager.



Now, all users from the data-source can be authenticated without having to be explicitly added to the CompleteFTP's internal user list.

# Change the home folder of a user

Before getting started, ensure that the user whose home folder you want to change has already been created. In this guide, the 'JoeSmith' user will be used.

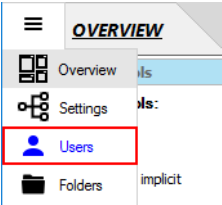
In order to change the home folder of a user, you need to:

- I. Change the user's home folder to another folder.
- II. Set the permissions for the user's new home folder.

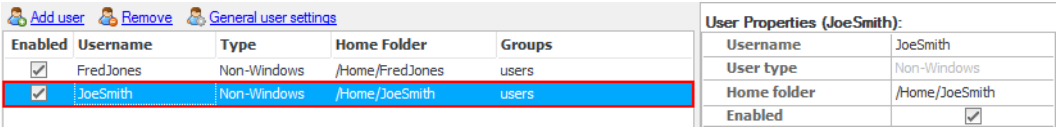
Here are the detailed steps:

I. **Change the user's home folder to another folder.**

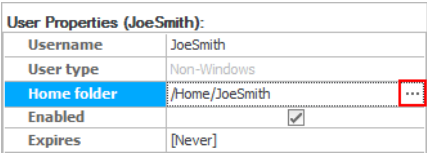
1. Select the 'Users' panel from the side-bar menu.



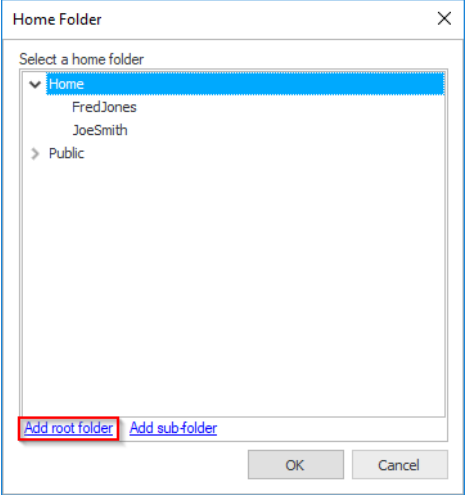
2. In the user list, select the user whose home folder you want to change, e.g. 'JoeSmith'.



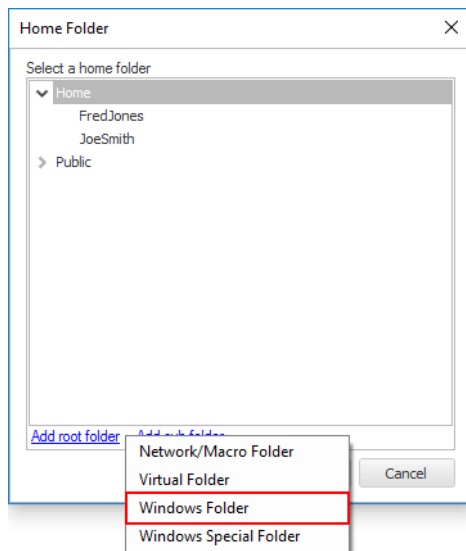
3. In the 'User Properties' window, click the ellipsis button (...) on the right of 'Home folder' to change the user's home folder to another folder.



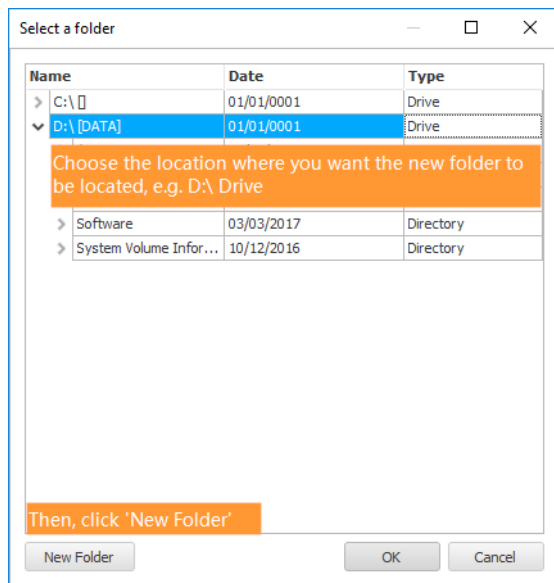
4. In the 'Home Folder' dialog box, create a new folder via the 'Add root folder' or 'Add sub-folder' link. In this tutorial, the 'Add root folder' link is selected.



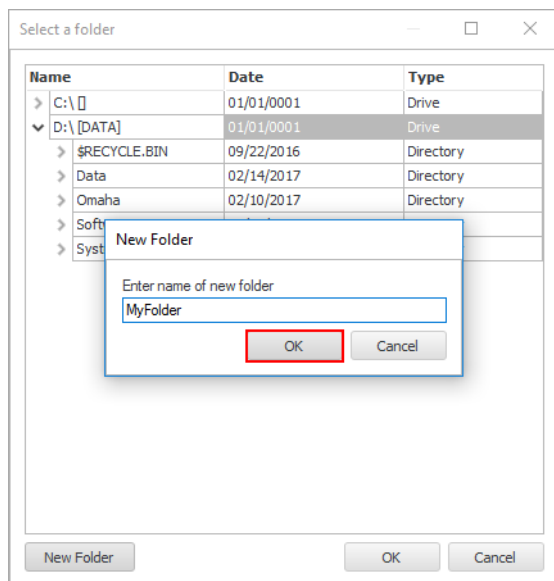
5. Select the 'Windows Folder' from the menu.



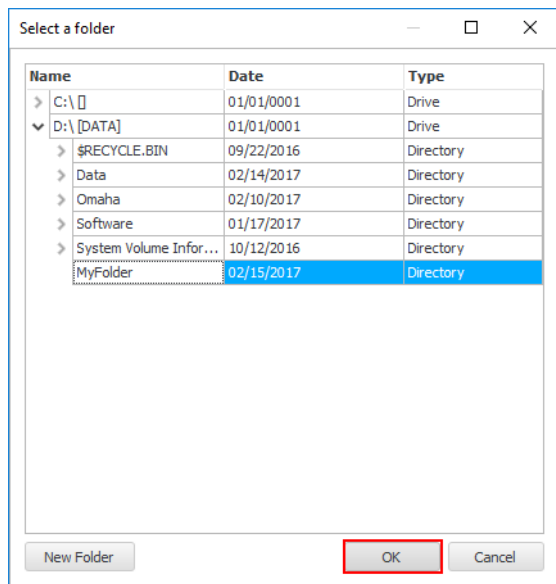
6. In the 'Select a folder' dialog box, do the following.



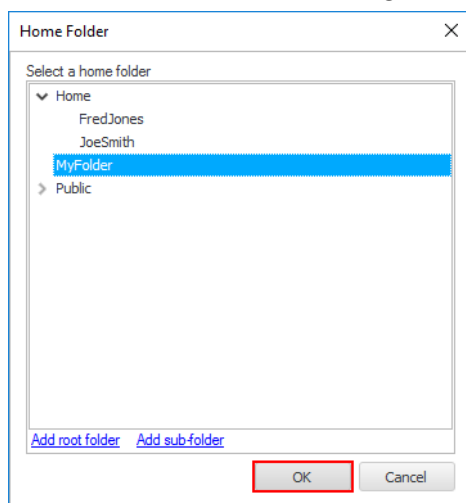
7. Enter the folder name, e.g. 'MyFolder' then click 'OK'.



8. Click the 'OK' button to close the 'Select a folder' dialog box.



9. Then click 'OK' to close the 'Home Folder' dialog box.

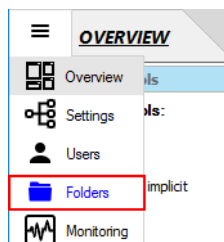


Verify that the user's home folder has been changed successfully.

<a href="#">Add user</a> <a href="#">Remove</a> <a href="#">General user settings</a>					User Properties (JoeSmith):	
Enabled	Username	Type	Home Folder	Full Name	Username	JoeSmith
<input checked="" type="checkbox"/>	FredJones	Non-Windows	/Home/FredJones		User type	Non-Windows
<input checked="" type="checkbox"/>	JoeSmith	Non-Windows	/MyFolder		Home folder	/MyFolder ...
					Enabled	<input checked="" type="checkbox"/>
					Expires	[Never]

## II. Set the permissions for the user's new home folder.

1. Select the 'Folders' panel from the side-bar menu.



2. Select the newly added folder in the list, i.e. 'MyFolder'.

<a href="#">Add root folder</a> <a href="#">Add sub-folder</a> <a href="#">Remove</a>			Folder Properties:	
Name	Type	Owner	Name	MyFolder
Home	Windows Folder	admin	Type	Windows Folder
FredJones	Windows Folder	FredJones	Path	D:\MyFolder
JoeSmith	Windows Folder	JoeSmith	Hidden	<input type="checkbox"/>
MyFolder	Windows Folder	admin	Access (non-Windows users)	
Public	Windows Folder	anonymous	Access (Windows users)	

3. In the 'Folder Properties' window, set the folder's permissions to allow the user to access it. This depends on the user's type.

a. If the user is a non-Windows user:

In the 'Access (non-Windows users)' section, set the folder's owner to the user, i.e. 'JoeSmith' and give full access to the owner.

**Folder Properties:** Free/Standard Edition

Name	MyFolder
Type	Windows Folder
Path	D:\MyFolder
Hidden	<input type="checkbox"/>

**Access (non-Windows users)**

Owner	JoeSmith
Permissions	<input checked="" type="radio"/> Owner (full) <input type="radio"/> All users (read) <input type="radio"/> All users (full)

**Folder Prop** Professional/Enterprise Edition

Name	MyFolder
Type	Windows Folder
Path	D:\MyFolder
Hidden	<input type="checkbox"/>

**Access (non-Windows users)**

Owner	JoeSmith
Group	users
Permissions	Click to view

**Access (Windows users)**

**Timestamps**

Owner can...

- ☒ download files
- ☒ upload files
- ☒ rename files
- ☒ delete files
- ☒ list folder contents
- ☒ change into folder
- ☒ create subfolders
- ☒ rename subfolders
- ☒ delete subfolders
- ☒ innore filters

b. If the user is a Windows user (This is not available in Free Edition):

Windows file-system permission may be viewed in the 'Access (Windows users)' section but can't be modified from here, you must use Windows Explorer to edit Windows permissions.

**Folder Properties:**

Name	MyFolder
Type	Windows Folder
Path	D:\MyFolder
Hidden	<input type="checkbox"/>

**Access (non-Windows users)**

**Access (Windows users)**

**Permissions** Click to view

**Timestamps**

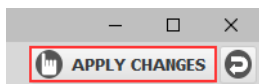
Group or user names:

- BUILTIN\Administrators
- BUILTIN\Users
- NT AUTHORITY\Authenticated Users
- NT AUTHORITY\SYSTEM

Permissions for BUILTIN\Administrators:

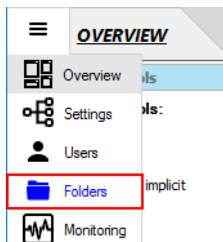
Permission	Allow	Deny
Full Control	yes	
Modify	yes	
Read & Execute	yes	
List Folder Contents	yes	
Read	yes	
Write	yes	

4. Finally, click the 'Apply changes' button at the top right of the CompleteFTP Manager.

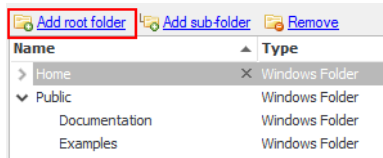


# Add a network folder

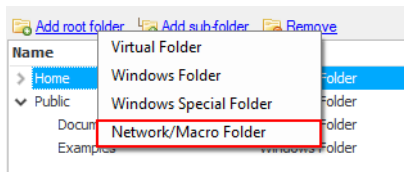
1. Select the 'Folders' panel from the side-bar menu.



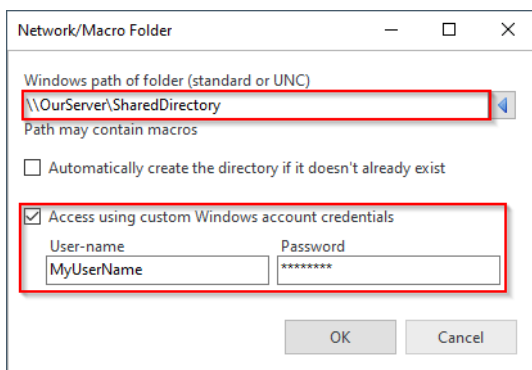
2. Select the 'Add root folder' link at the top of the panel. Note that you can also right-click on an existing folder to create a subfolder.



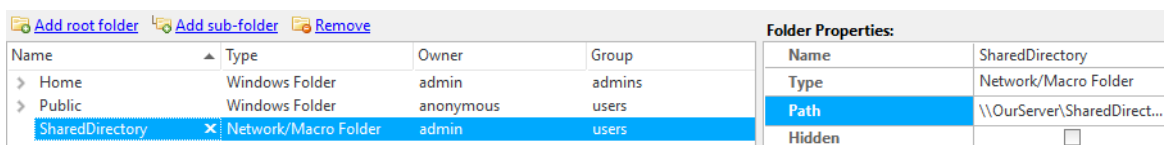
3. Choose the 'Network/Macro Folder' to create a network folder.



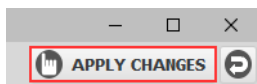
4. Enter the Windows path of the folder (in standard format or in UNC format), e.g. '\\OurServer\SharedDirectory'. Then click 'OK'. If the connecting user is a Windows user, or the account under which the CompleteFTP service is running (SYSTEM by default), has permission to access the folder then it's not necessary to provide custom credentials, otherwise these should be entered.



Now, the new network folder is displayed in the folder list.



5. Click the 'Apply changes' button at the top right of the CompleteFTP Manager, in order to save the changes.



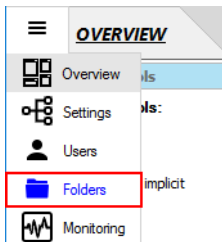
# Allow users to share a folder (Method 1)

This guide illustrates how to allow multiple users to access a single Windows folder, by creating a subfolder under each user's home folder which maps to that Windows folder.

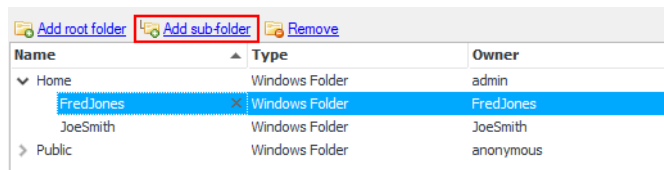
In order to achieve that, you simply need to create a sub-folder in the virtual file-system underneath the home folder of each user, which all map to the same folder in the Windows file-system.

Firstly, ensure that the users have already been created in the 'Users' panel. In this guide, 'FredJones' and 'JoeSmith' non-Windows users will be used.

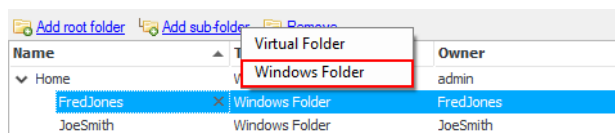
1. Select the 'Folders' panel from the side-bar menu.



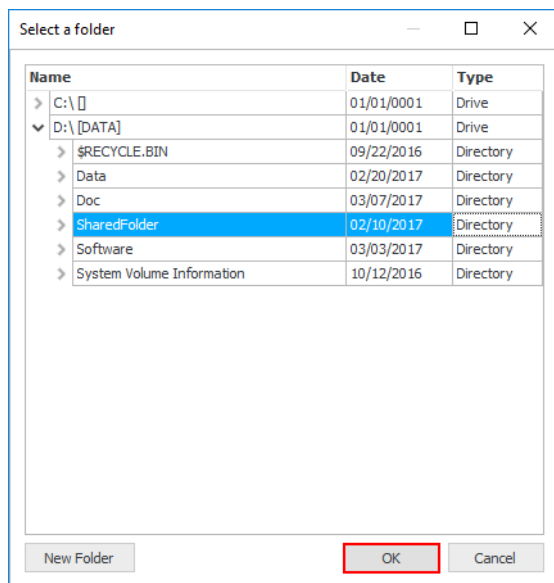
2. Add a Windows subfolder underneath the home folder of each user, that maps to the same folder in the Windows file-system.
  - a. Select the 'FredJones' user from the list then click 'Add sub-folder'.



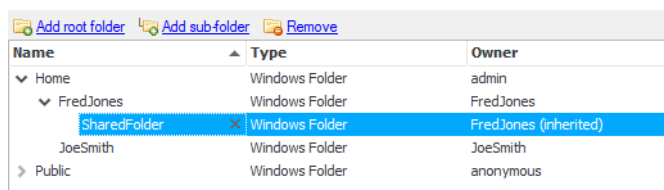
- b. Select the 'Windows Folder' menu item.



- c. Select a folder which will be shared among the users, e.g. 'SharedFolder'. You can also create a new one via the 'New Folder' button.



- d. The result will look like this.



- e. Repeat steps 2.a to 2.c for 'JoeSmith' user.

<a href="#">Add root folder</a> <a href="#">Add sub-folder</a> <a href="#">Remove</a>		
Name	Type	Owner
Home	Windows Folder	admin
▼ FredJones	Windows Folder	FredJones
SharedFolder	Windows Folder	FredJones (inherited)
▼ JoeSmith	Windows Folder	JoeSmith
SharedFolder	Windows Folder	JoeSmith (inherited)
Public	Windows Folder	anonymous

3. Give full access permissions to the folder to the owner.

a. In Free/Standard edition, select both users, then select 'Owner (full)' option in the 'Folder Properties' window.

<a href="#">Add root folder</a> <a href="#">Add sub-folder</a> <a href="#">Remove</a>		
Name	Type	Owner
Home	Windows Folder	admin
▼ FredJones	Windows Folder	FredJones
SharedFolder	Windows Folder	FredJones (inherited)
▼ JoeSmith	Windows Folder	JoeSmith
SharedFolder	Windows Folder	JoeSmith (inherited)
Public	Windows Folder	anonymous

**Folder Properties:**  

Name	SharedFolder
Type	Windows Folder
Path	D:\SharedFolder
Hidden	<input type="checkbox"/>
<b>Access</b>	
Owner	FredJones (inherited)
Permissions	<input checked="" type="radio"/> Owner (full) <input type="radio"/> All users (read) <input type="radio"/> All users (full)
<b>Timestamps</b>	

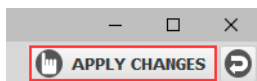
b. In Professional/Enterprise MFT, select both users, then check all permission checkboxes in the 'Owner' tab.

<a href="#">Add root folder</a> <a href="#">Add sub-folder</a> <a href="#">Remove</a>			
Name	Type	Owner	Group
Home	Windows Folder	admin	admins
▼ FredJones	Windows Folder	FredJones	users
SharedFolder	Windows Folder	FredJones (inherited)	users (inherited)
▼ JoeSmith	Windows Folder	JoeSmith	users
SharedFolder	Windows Folder	JoeSmith (inherited)	users (inherited)
Public	Windows Folder	anonymous	users

**Folder Properties:**  

Name	SharedFolder
Type	Windows Folder
Path	D:\SharedFolder
Hidden	<input type="checkbox"/>
<b>Access (non-Windows users)</b>	
Owner	FredJones (inherited)
Group	users (inherited)
Permissions	Click to view (inheri... X)
<b>Access (Windows users)</b>	
Permissions	<div> Owner can... <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> download files</li> <li><input checked="" type="checkbox"/> upload files</li> <li><input checked="" type="checkbox"/> rename files</li> <li><input checked="" type="checkbox"/> delete files</li> <li><input checked="" type="checkbox"/> list folder contents</li> <li><input checked="" type="checkbox"/> change into folder</li> <li><input checked="" type="checkbox"/> create subfolders</li> <li><input checked="" type="checkbox"/> rename subfolders</li> <li><input checked="" type="checkbox"/> delete subfolders</li> <li><input checked="" type="checkbox"/> ignore filters</li> </ul> </div>
<b>Timestamps</b>	

4. Finally, click the 'Apply changes' button at the top right of the CompleteFTP Manager, in order to save the changes.



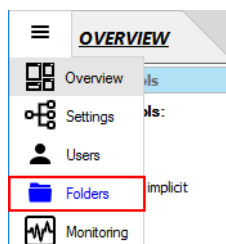


## Allow users to share a folder (Method 2)

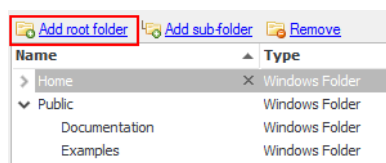
This guide illustrates how to allow multiple users, with different Windows home folders, to share a common sub-folder. This is done by using a %UserName% macro in the Windows path of the home folder so that, even though they will all have the same home folder (network/macro folder) in the virtual file-system, this folder will map to their different Windows home folders.

Before getting started, ensure that the users have already been created in the Users panel. In this guide, 'FredJones' and 'JoeSmith' non-Windows users will be used.

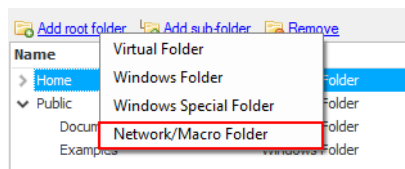
1. Create a root network/macro folder that contains the %UserName% macro.
  - a. Select the 'Folders' panel from the side-bar menu.



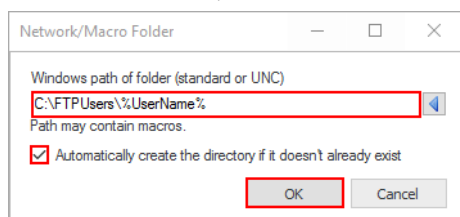
- b. Click the 'Add root folder' link at the top of the panel.



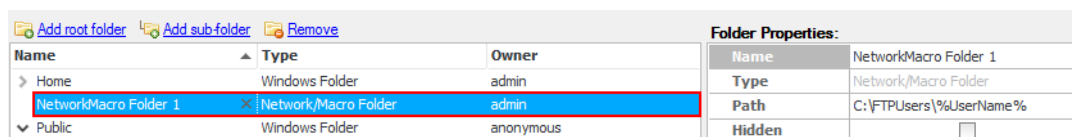
- c. Select the 'Network/Macro' folder from the menu.



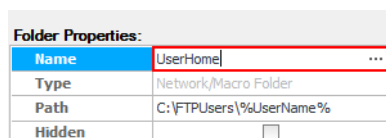
- d. In the 'Network/Macro Folder' dialog box, enter the Windows path of the folder, e.g. 'C:\FTPUsers\%UserName%', then check the 'Automatically create the directory' checkbox and click 'OK'. (Please note the User's Home Folder must be listed under this directory - C:\FTPUsers- for this example to work)



- e. Select the folder you've just created, it's probably called 'NetworkMacro Folder 1'.

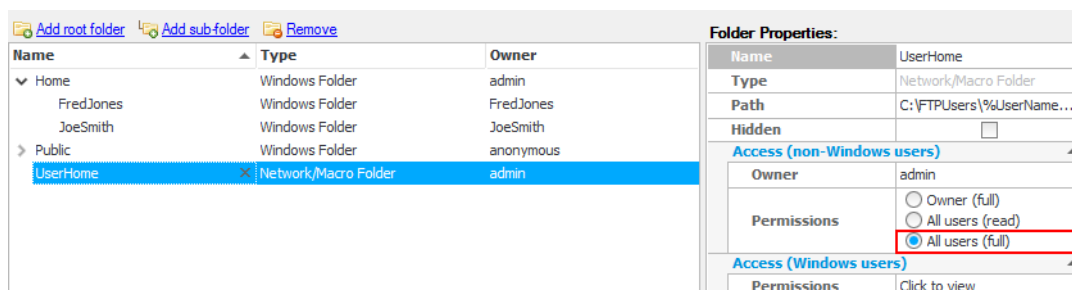


- f. In the 'Folder Properties' window, change the folder's name to a meaningful one, e.g. 'UserHome'.



This will be the home folder in the virtual file-system for the users, although it will map to a different Windows directory for each user.

2. Give full access permissions to /UserHome folder to all users. In other words, all users who have an account on the server will be able to access to their own unique Windows folder through /UserHome, even if their home folder is elsewhere
  - a. If you're using Standard Edition, select the 'All users (full)' option in 'Folder Properties' window (It's selected by default).



- b. If you're using Professional/Enterprise MFT, select 'All users' tab and then check all permission checkboxes.

The screenshot shows the 'Folder Properties' dialog box for the 'UserHome' folder. The 'Permissions' tab is selected, and the 'All users' group is chosen. All permissions are checked, including 'download files', 'upload files', 'rename files', 'delete files', 'list folder contents', 'change into folder', 'create subfolders', 'rename subfolders', 'delete subfolders', and 'inheritance filters'.

Name	Type	Owner	Group
Home	Windows Folder	admin	admins
FredJones	Windows Folder	FredJones	users
JoeSmith	Windows Folder	JoeSmith	users
Public	Windows Folder	anonymous	users
UserHome	Network/Macro Folder	admin	users

In these higher editions, permissions can be more tightly controlled using groups.

3. Add the folder that will be shared among the users beneath /UserHome.

- a. Select the 'UserHome' folder then click the 'Add sub-folder' link.

The screenshot shows the 'Add sub-folder' dialog box. The 'UserHome' folder is selected in the list of folders to be added as a sub-folder.

Name	Type	Owner
Home	Windows Folder	admin
Public	Windows Folder	anonymous
Documentation	Windows Folder	web
Examples	Windows Folder	web
UserHome	Network/Macro Folder	admin

- b. Select the 'Windows Folder' from the menu.

The screenshot shows the 'Add sub-folder' dialog box. The 'Windows Folder' is selected in the menu of folder types.

- c. In the 'Select a folder' dialog box, select the folder that users will share with one another, e.g. 'SharedFolder'. You can also create a new one via the 'New Folder' button.

The screenshot shows the 'Select a folder' dialog box. The 'SharedFolder' is selected in the list of folders.

Name	Date	Type
C:\ [ ]	01/01/0001	Drive
D:\ [DATA]	01/01/0001	Drive
\$RECYCLE.BIN	09/22/2016	Directory
Data	02/20/2017	Directory
Doc	03/07/2017	Directory
SharedFolder	02/10/2017	Directory
Software	03/03/2017	Directory
System Volume Information	10/12/2016	Directory

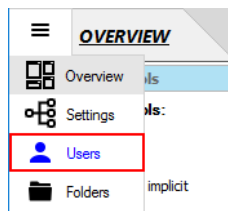
Now, the 'SharedFolder' is added underneath the 'UserHome' folder.

The screenshot shows the 'Folder Properties' dialog box for the 'SharedFolder' folder. The 'UserHome' folder is selected in the list of folders.

Name	Type	Owner
Home	Windows Folder	admin
FredJones	Windows Folder	FredJones
JoeSmith	Windows Folder	JoeSmith
Public	Windows Folder	anonymous
UserHome	Network/Macro Folder	admin
SharedFolder	Windows Folder	admin (inherited)

4. Set the home folders of the users to /UserHome folder.

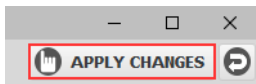
- a. Select the 'Users' panel from the side-bar menu.



- b. Select both 'JoeSmith' and 'FredJones' and set their 'Home Folder' to '/UserHome'.

<a href="#">Add user</a> <a href="#">Remove</a> <a href="#">General user settings</a>					<b>User Properties (FredJones):</b>	
Enabled	Username	Type	Home Folder	Full Name	Username	FredJones
<input checked="" type="checkbox"/>	FredJones	Non-Windows	/UserHome		User type	Non-Windows
<input checked="" type="checkbox"/>	JoeSmith	Non-Windows	/UserHome		Home folder	/UserHome
					Enabled	<input checked="" type="checkbox"/>

5. Finally, click the 'Apply changes' button at the top right of the CompleteFTP Manager.

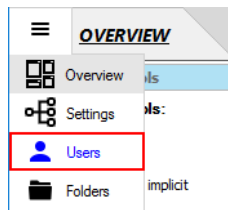


# Allow users to share a folder (Method 3)

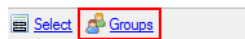
This guide illustrates how to allow a group of users, who have their own home folder, to share access to a single folder which is not underneath their home folder. This is done by giving full access permissions to the folder to the group that these users belong to.

Before getting started, ensure that the users have already been created in the Users panel. In this guide, 'FredJones' and 'JoeSmith' non-Windows users will be used.

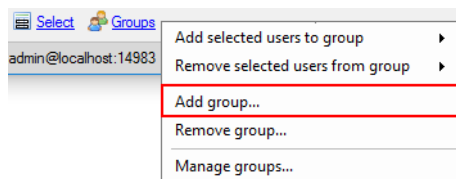
1. Create a new group.
  - a. Select the 'Users' panel from the side-bar menu.



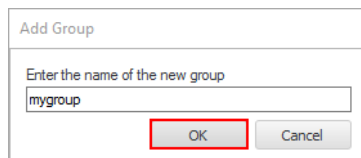
- b. Click the 'Groups' link at the bottom of the panel.



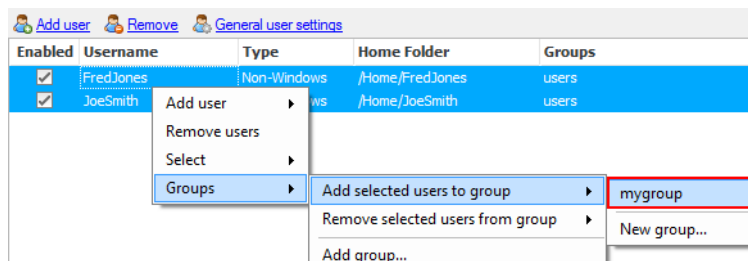
- c. Select the 'Add group' menu item.



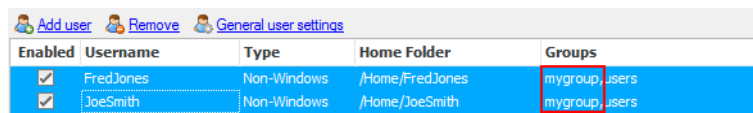
- d. In the 'Add Group' dialog box, enter the group name, e.g. 'mygroup' and click 'OK'.



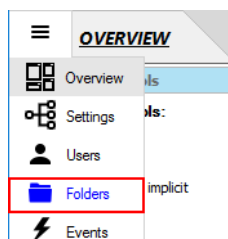
2. Add the users to the newly created group.
  - a. Select both 'FredJones' and 'JoeSmith' from the user list, then right click and select 'Groups' -> 'Add selected users to group' -> 'mygroup'.



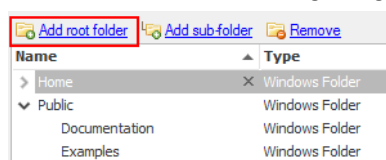
- b. Now, both 'FredJones' and 'JoeSmith' users have been added to 'mygroup'.



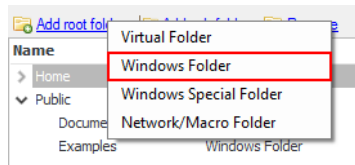
3. Create a folder that will be shared among the users.
  - a. Select the 'Folders' panel from the side-bar menu.



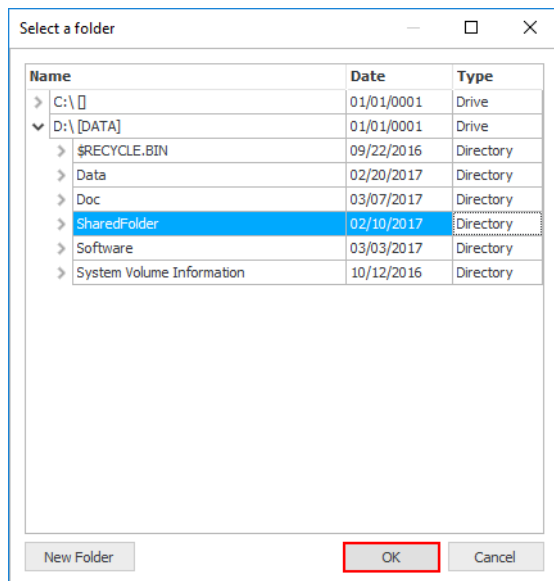
- b. Click the 'Add root folder' link at the top of the panel.



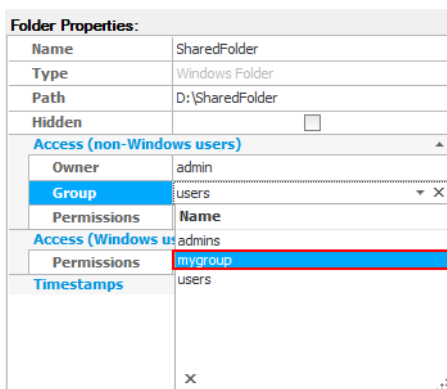
c. Select the 'Windows Folder' menu item.



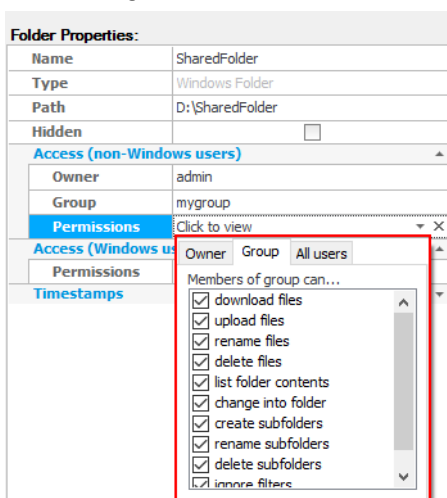
d. Select a folder in the list then click 'OK'. You can also create a new one via the 'New Folder' button.



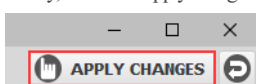
4. Select the 'SharedFolder' folder and set its group property to 'mygroup' in the 'Folder Properties' window.



5. Give full access permissions to the 'SharedFolder' to the 'mygroup'.



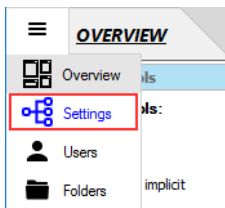
6. Finally, click the 'Apply changes' button at the top right of the CompleteFTP Manager to save the changes.



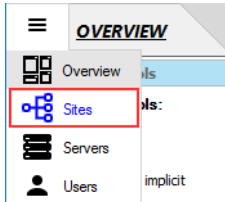
# Block connections from specific IP addresses (blacklisting)

This guide illustrates how to allow access from anywhere except specific (blacklisted) IP addresses.

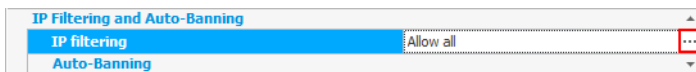
1. From the side-bar menu, select the 'Settings' panel (Professional Edition)



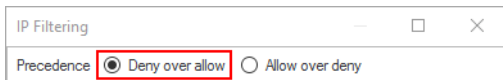
OR the 'Sites' panel (Enterprise MFT).



2. Click the ellipsis button of the 'IP filtering' property of the 'IP filtering and Auto-Banning' category.

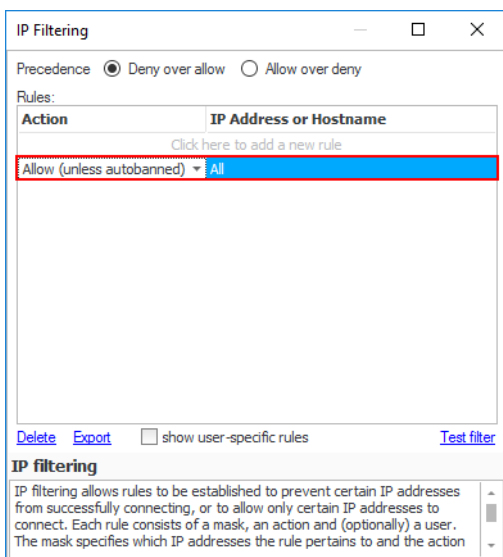


3. In the 'IP Filtering dialog', make sure the 'Deny over allow' option is selected.



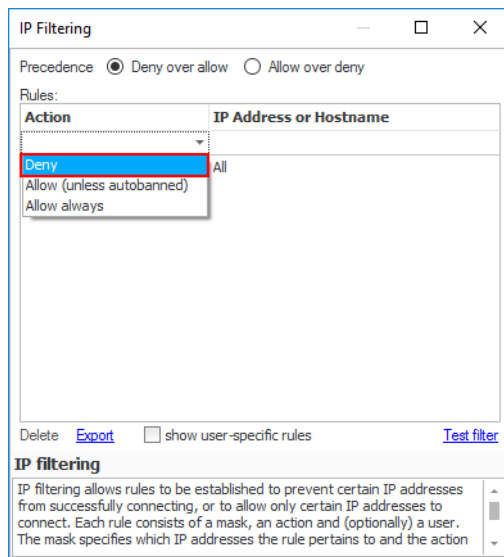
4. Create a rule to allow connection from all IP addresses.

By default there's a rule that allows connections from all IP address. If you have changed/removed it you'll need to add it again.

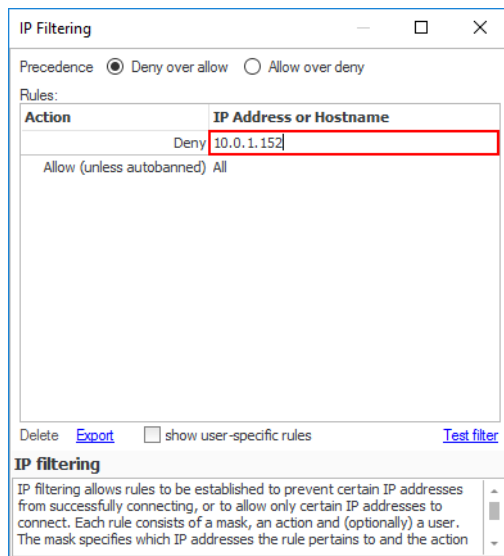


Once this is done, all connections will be permitted from any IP address.

5. Create a rule that denies connections from a specific IP address.
  - a. Add a 'Deny' rule.



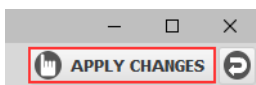
b. Enter the IP address you wish to block, e.g. 10.0.1.152.



Note that it's also possible to deny connections from:

- The first one, two or three parts of an IP address, such as 192.168, which specifies all the IP addresses that begin with those parts.
- A specific numeric range of IP addresses, such as 192.168.2.100-150.
- An Internet host-name.
- A LAN network name.

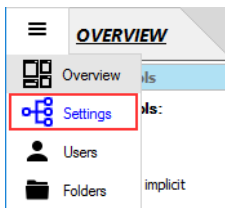
6. Close the 'IP Filtering' dialog then click the 'Apply changes' button.



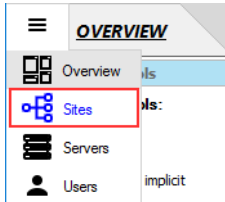
# Accept connections only from specific IP addresses (whitelisting)

This guide illustrates how to deny access from everywhere except specific (whitelisted) IP addresses.

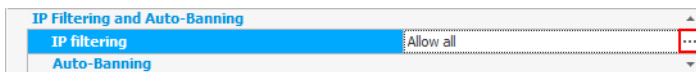
1. From the side-bar menu, select the 'Settings' panel (Professional Edition)



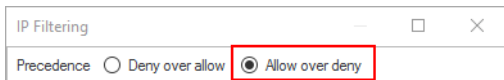
OR the 'Sites' panel (Enterprise MFT).



2. Click the ellipsis button of the 'IP filtering' property of the 'IP filtering and Auto-Banning' category.

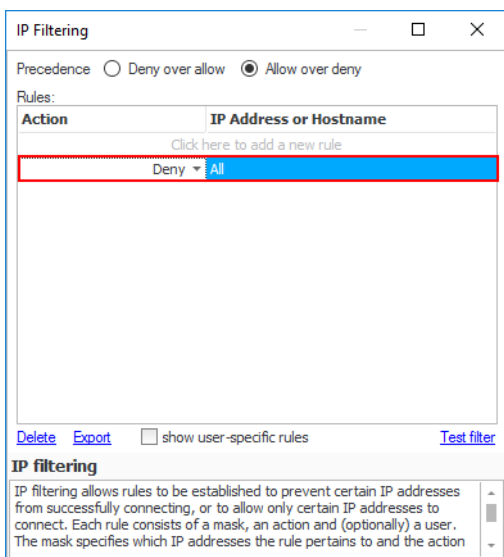


3. In the 'IP Filtering dialog', select the 'Allow over deny' radio button.



4. Create a rule to deny access from all IP addresses.

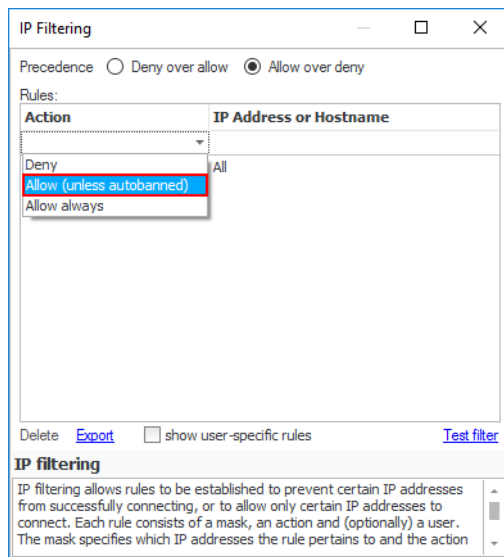
If there's currently a rule that allows all connections, change this to deny all.



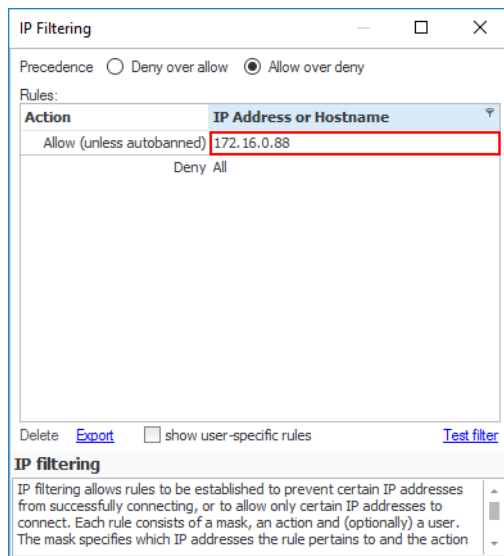
Now, no connections will be allowed from any IP address.

5. Create a rule that allows connections from a specific IP address.
  - a. Add an 'Allow (unless autobanned)' rule.





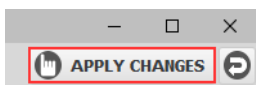
b. Enter the IP address that is permitted to connect to the CompleteFTP server, e.g. 172.16.0.88.



Note that it's also possible to allow connections from:

- The first one, two or three parts of an IP address, such as 192.168, which specifies all the IP addresses that begin with those parts.
- A specific numeric range of IP addresses, such as 192.168.2.100-150.
- An Internet host-name.
- A LAN network name.

6. Close the 'IP Filtering' dialog then click the 'Apply changes' button.



# Accept connections only from specific users at specific IP addresses

It's possible to configure CompleteFTP to deny access from all users, except some specific users at specific IP addresses.

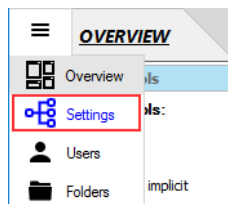
The main steps will be:

- I. Denying access from other IP addresses
- II. Allowing specific IP addresses for specific users

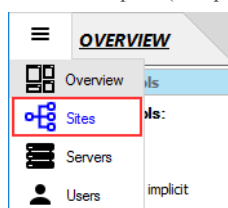
Here are the detailed steps:

## I. Denying access from other IP addresses

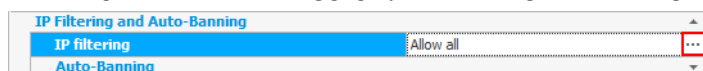
1. From the side-bar menu, select the 'Settings' panel (Professional Edition)



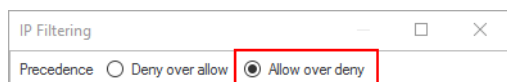
OR the 'Sites' panel (Enterprise MFT).



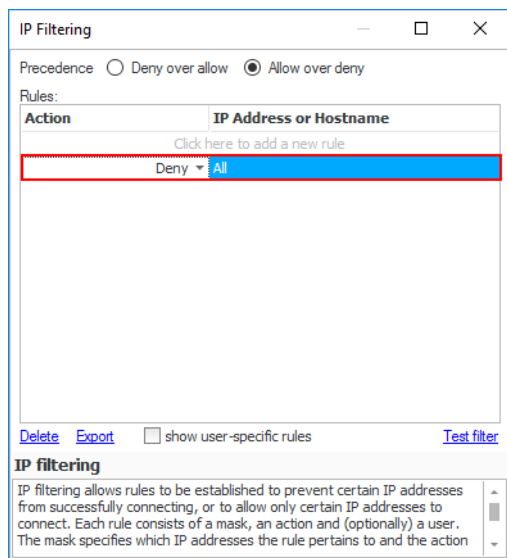
2. Click the ellipsis button of the 'IP filtering' property of the 'IP filtering and Auto-Banning' category.



3. In the 'IP Filtering dialog', select the 'Allow over deny' radio button.



4. If there's currently a rule that allows all connections, change this to deny all.

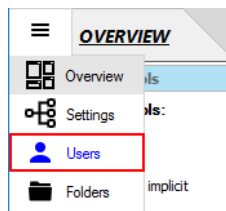


Once this is done, no connections will be allowed from any IP addresses.

## II. Allowing specific IP addresses for specific users

Now that CompleteFTP has been configured to deny connections by default, we can go ahead and allow specific users to connect from specific IP addresses, as follows:

1. Select the 'Users' panel from the side-bar menu.



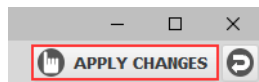
2. Select one or more users, e.g. select 'JoeSmith'.

<a href="#">Add user</a> <a href="#">Remove</a> <a href="#">General user settings</a>					<b>User Properties (JoeSmith):</b>	
Enabled	Username	Type	Home Folder	Groups	Username	JoeSmith
<input checked="" type="checkbox"/>	FredJones	Non-Windows	/Home/FredJones	users	User type	Non-Windows
<input checked="" type="checkbox"/>	JoeSmith	Non-Windows	/Home/JoeSmith	users	Home folder	/Home/JoeSmith
					Enabled	<input checked="" type="checkbox"/>

3. Enter the IP address(es) from which the user is allowed to connect. Separate multiple IP addresses with commas, e.g. '10.0.1.138, 10.0.1.123'. Domain-names may also be used.

<a href="#">Add user</a> <a href="#">Remove</a> <a href="#">General user settings</a>					<b>User Properties (JoeSmith):</b>	
Enabled	Username	Type	Home Folder	Groups	Username	JoeSmith
<input checked="" type="checkbox"/>	FredJones	Non-Windows	/Home/FredJones	users	User type	Non-Windows
<input checked="" type="checkbox"/>	JoeSmith	Non-Windows	/Home/JoeSmith	users	Home folder	/Home/JoeSmith
					Enabled	<input checked="" type="checkbox"/>
					Expires	[Never]
					<b>Authentication</b>	
					Password	*****
					Public keys	No keys
					Allowed IP addresses	10.0.1.138, 10.0.1.123

4. Finally, click the 'Apply changes' button at the top right of the CompleteFTP Manager.



Now, only the 'JoeSmith' user from the specified IP addresses can connect to the CompleteFTP server (check the 'show user-specific rules' checkbox in the 'IP Filtering' dialog to show JoeSmith's rules).

IP Filtering

Precedence
☐ Deny over allow
☒ Allow over deny

Rules:

Action	IP Address or Hostname	User
Click here to add a new rule		
Deny	All	None
Allow (unless autob...	10.0.1.138	JoeSmith
Allow (unless autob...	10.0.1.123	JoeSmith

[Delete](#)
[Export](#)
☒ show user-specific rules
[Test filter](#)

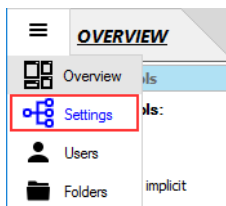
**IP filtering**

IP filtering allows rules to be established to prevent certain IP addresses from successfully connecting, or to allow only certain IP addresses to connect. Each rule consists of a mask, an action and (optionally) a user. The mask specifies which IP addresses the rule pertains to and the action

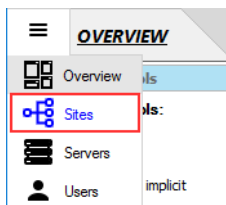
# Restrict some users to connect only from specific IP addresses

Sometimes it's desirable to allow some users, to connect only from specific IP addresses, while allowing all other users to connect from any IP address. This guide will show you how to do this.

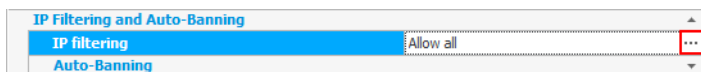
1. From the side-bar menu, select the 'Settings' panel (if you're using Professional Edition)



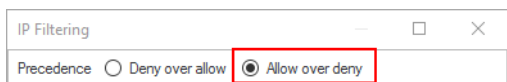
OR select the 'Sites' panel (if you're using Enterprise MFT)



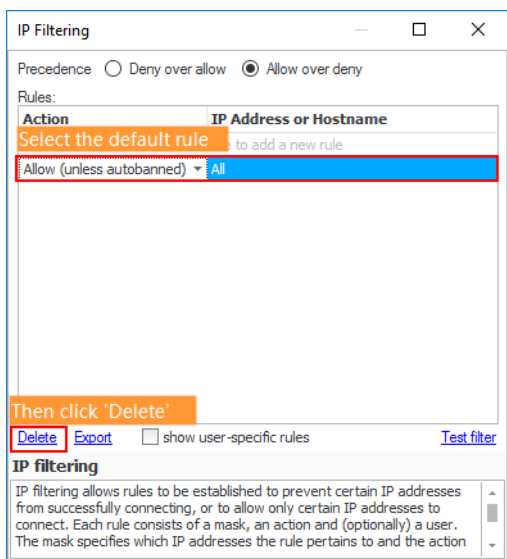
2. Select the 'IP filtering' property in the 'IP filtering and Auto-banning' category. Then click the ellipsis (...) button.



3. In the 'IP Filtering' dialog, set the precedence to 'Allow over deny' option.

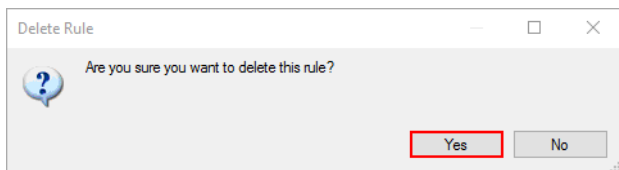


4. Then delete the default rule.



Why? If no rule applies to an incoming connection, then the first part of the precedence is used (i.e. 'Deny' for 'Deny over allow' and 'Allow' for 'Allow over deny'). Therefore, in this example, unless the user-specific rules that will be defined below apply, then an incoming connection will be allowed (Remember, in this example we are using the 'Allow over Deny' precedence).

5. Click the 'Yes' button when the confirmation dialog appears.



6. Check the 'show user-specific rules' checkbox.

IP Filtering

Precedence ☐ Deny over allow ☒ Allow over deny

Rules:

Action	IP Address or Hostname	User
		None

Delete [Export](#) ☒ show user-specific rules [Test filter](#)

**IP filtering**

IP filtering allows rules to be established to prevent certain IP addresses from successfully connecting, or to allow only certain IP addresses to connect. Each rule consists of a mask, an action and (optionally) a user. The mask specifies which IP addresses the rule pertains to and the action

7. Now, add a new rule to deny access of the 'JoeSmith' user from all IP addresses.
- Click the empty cell under the 'Action' column then select the 'Deny' action.

IP Filtering

Precedence ☐ Deny over allow ☒ Allow over deny

Rules:

Action	IP Address or Hostname	User
Deny		None

Delete [Export](#) ☒ show user-specific rules [Test filter](#)

**IP filtering**

IP filtering allows rules to be established to prevent certain IP addresses from successfully connecting, or to allow only certain IP addresses to connect. Each rule consists of a mask, an action and (optionally) a user. The mask specifies which IP addresses the rule pertains to and the action

- After the action is selected, the rule is automatically set for all IP addresses. Let it be!

IP Filtering

Precedence ☐ Deny over allow ☒ Allow over deny

Rules:

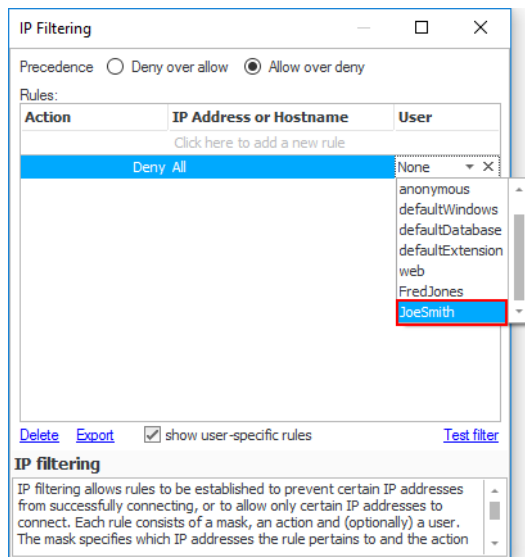
Action	IP Address or Hostname	User
Deny	All	None

Delete [Export](#) ☒ show user-specific rules [Test filter](#)

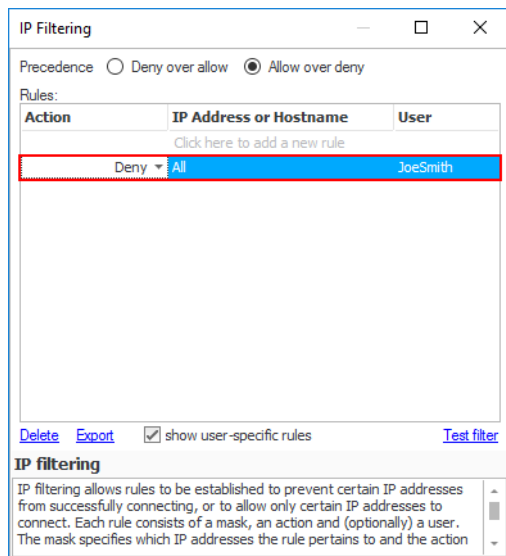
**IP filtering**

IP filtering allows rules to be established to prevent certain IP addresses from successfully connecting, or to allow only certain IP addresses to connect. Each rule consists of a mask, an action and (optionally) a user. The mask specifies which IP addresses the rule pertains to and the action

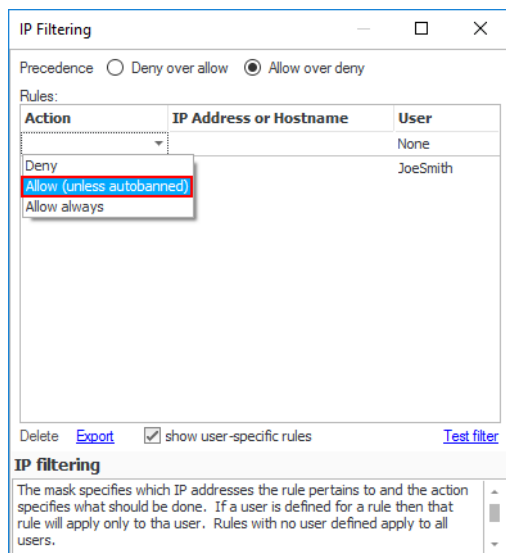
- Select the 'JoeSmith' user from the list.



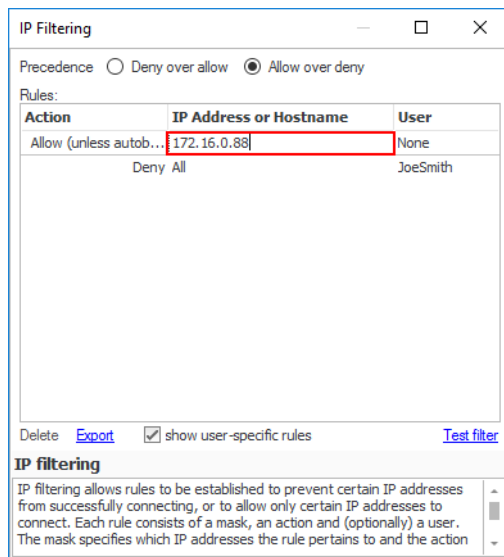
d. Now, the rule that prevents the 'JoeSmith' user from connecting to the server from all IP addresses will look like this:



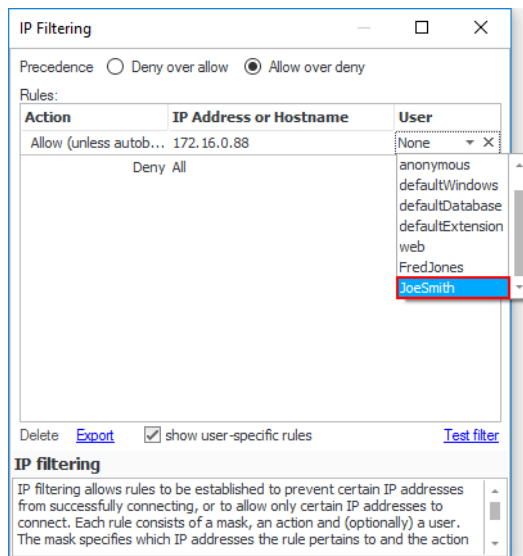
8. Add another rule to allow the 'JoeSmith' user to be able to access from specified IP addresses.
  - a. Add an 'Allow (unless autobanned)' rule.



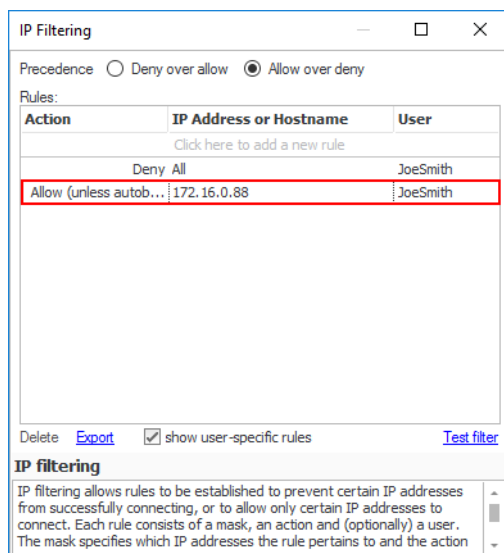
- b. Enter the IP address from which the 'JoeSmith' user is allowed to connect, e.g. 172.16.0.88.



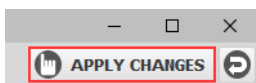
c. Then Select the 'JoeSmith' user from the list, which is the one who this rule will apply to.



d. Now, the rule that allows the 'JoeSmith' user to access from the specified IP address will look like this:

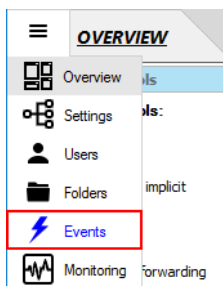


9. Close the 'IP Filtering' dialog, then click the 'Apply changes' button.



# E-mail notification when a file is uploaded

1. Select the 'Events' panel from the side-bar menu.



2. Select the 'Email Notifications' tab at the top of the panel.



3. Enter the mail server (SMTP server) settings.
  - a. Specify the server's hostname/IP address, e.g. 'smtp.myserver.com'.

A screenshot of the 'SMTP Server' configuration form. The 'Host-name / IP address' field contains the text 'smtp.myserver.com' and is highlighted with a red rectangular box. Other fields include 'Port', 'Username', 'Password', and a checkbox for 'SSL/TLS'. There is also a checkbox for 'Show password characters'.

- b. Specify the port number (normally port 25)

A screenshot of the 'SMTP Server' configuration form. The 'Port' field is a dropdown menu showing '25' and is highlighted with a red rectangular box. The 'Host-name / IP address' field contains 'smtp.myserver.com'.

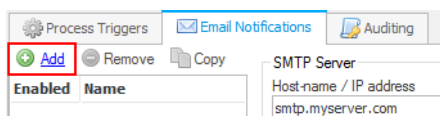
If you enable SSL connections by checking the "Enabled SSL/TLS" checkbox, the port number will most likely need to be changed - the default SMTP SSL port is 465.

A screenshot of the 'SMTP Server' configuration form. The 'Port' field is a dropdown menu showing '465' and is highlighted with a red rectangular box. The 'SSL/TLS' checkbox is checked. The 'Host-name / IP address' field contains 'smtp.myserver.com'.

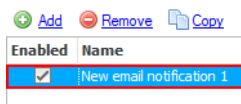
- c. Specify the SMTP username and password.

A screenshot of the 'SMTP Server' configuration form. The 'Username' field contains 'myuser' and the 'Password' field contains 'password'. Both fields are highlighted with a red rectangular box. The 'Port' field shows '25' and the 'SSL/TLS' checkbox is unchecked.

4. Click the 'Add' button to create a new email notification.



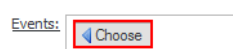
5. Select the email notification you just created, it's probably called 'New email notification 1'.



6. Enter an appropriate name for this email notification, e.g. 'New file upload'.

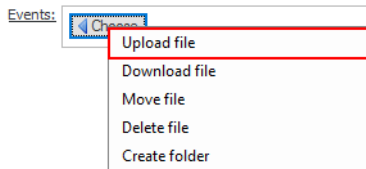
A screenshot of the email notification configuration form. The 'Name' field contains the text 'New file upload' and is highlighted with a red rectangular box. To the left is a table with columns 'Enabled' and 'Name', showing the selected notification. To the right are fields for 'Host-name / IP address', 'Port', 'Username', 'Password', and a checkbox for 'SSL/TLS'.

7. In the 'Events' section, click the 'Choose' button.



8. Then select the 'Upload file' from the menu.





Once the event type is selected, extra fields appear below.

Errors: ☒ Trigger on success ☐ Trigger on error

Folder filter:   ☐ Inverse

User filter:   ☐ Inverse

From:

To:

Subject:

Message:

9. Select the 'Trigger on success' checkbox (it is selected by default).

Errors: ☒ Trigger on success ☐ Trigger on error

10. Enter the folder- or file-path filter, which will then select the files/folders for which the event should be triggered. E.g. the filter 'Home/\*' will match all files/folders under the /Home folder.

Folder filter:   ☐ Inverse

Macros can be selected via the folder filter's left arrow button.

\* - Wildcard: Match zero or more characters  
 ? - Wildcard: Match one character  
 regex - Interpret filter as a regular expression (.NET-style)

11. Select users for which an event should occur, from the 'User Filter' dropdown list. By default, all users are selected.

User filter:   ☐ Inverse

From: ☒ anonymous

To: ☒ defaultWindows

Subject: ☒ defaultDatabase

Message: ☒ defaultExtension

From: ☒ web

To: ☒ JoeSmith

Subject: ☒ FredJones

OK Cancel

12. Select sites for which an event should occur, from the 'Site Filter' dropdown list (Enterprise MFT only). By default, all sites are selected.

Site filter:   ☐ Inverse

From: ☒ (Select All)

To: ☒ Default Site

Subject: ☒ Admin

OK Cancel

13. Enter the email address for which the email notification is to be sent from, e.g. 'from@myserver.com'.

From:

Macros can be used and selected with the left arrow button.

%OwnerEmail% - Email address of owner of folder  
 %LoginEmail% - Email address of logged-in user

14. Enter the email address the email notification is to be sent to, e.g. 'to@myserver.com'. Macros can also be used here. The list of macros is the same as the 'From' field's.

To:

15. Enter the subject of the notification message. E.g. 'A new file has been uploaded' %FileName%.

Subject:

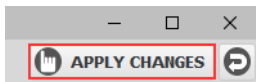
Various macros can be used here. Click the Subject's left arrow button to select them.

%FileName% - Name of file  
 %WindowsFolder% - Windows path to folder  
 %WindowsPath% - Full Windows path of file  
 %VirtualFolder% - Virtual path to folder  
 %VirtualPath% - Full virtual file-system path of file  
 %PathFilter% - File filter for this event  
 %SiteName% - Name of site  
 %TransferStatus% - Transfer status (success/failure)  
 %Date% - Date of event  
 %DateAndTime% - Date and time of event  
 %Time% - Time of event  
 %Timestamp:format% - Formatted timestamp of event  
 %Type% - Type of event  
 %ClientIP% - IP address of client  
 %LoginEmail% - Email address of logged-in user  
 %LoginFullName% - Full name of logged-in user  
 %LoginUserName% - Username of logged-in user  
 %OwnerEmail% - Email address of owner of folder  
 %OwnerFullName% - Full name of owner of folder  
 %OwnerUserName% - Username of owner of folder  
 %Notification% - Name of this notification

16. Enter the body of the notification message. E.g. 'A new file has been uploaded: User = %LoginUserName%, Path = %VirtualPath%'. Various macros can also be used here. The list of macros is same as the Subject's.

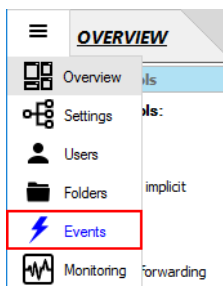
Message: A new file has been uploaded:  
 User = %LoginUserName%  
 Path = %VirtualPath%

17. Finally, click the 'Apply changes' button at the top right of the CompleteFTP Manager.



# Move a file to another directory after it's been uploaded

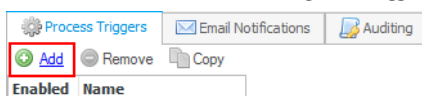
1. Select the 'Events' panel from the side-bar menu.



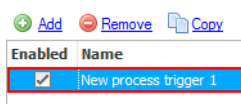
2. Select the 'Process Triggers' tab at the top of the panel.



3. Click the 'Add' button to create a new process trigger.



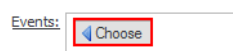
4. Select the process trigger you just created, it's probably called 'New process trigger 1'.



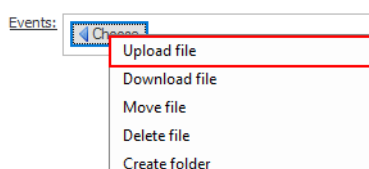
5. Enter an appropriate name for this process trigger, e.g. 'Move on upload'.



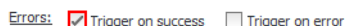
6. In the 'Events' section, click the 'Choose' button.



7. Then select the 'Upload file' event.



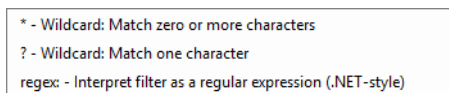
8. After the event is selected, some extra fields are added on the panel. Now, select the 'Trigger on success' checkbox (It is selected by default).



9. Enter the folder- or file-path filter to select the files/folders for which the event should be triggered. E.g. the filter 'Home/\*' will match all files/folders under the /Home folder.



Macros can be selected via the folder filter's left arrow button.



10. Select users for which an event should occur from the 'User Filter' dropdown list. By default, all users are selected.



11. Select sites for which an event should occur from the 'Site Filter' dropdown list (Enterprise MFT only). By default, all sites are selected.

Site filter: (All sites) ☐ Inverse

Type: ☒ (Select All)

Program file: ☒ Default Site ☐ ...

Arguments: ☒ Admin ☐ ...

OK Cancel

12. Select the 'Batch script' radio button.

Type: ☐ Program ☒ Batch script ☐ Powershell script ☐ JSS script ☐ FTP script

13. Enter the following script which will move uploaded files to the directory, 'C:\FTP\Incoming\username\' while username is the name of the logged in user.

Script: `move "%WindowsPath%" "C:\FTP\Incoming\%LoginUserName%\%FileName%"`

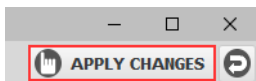
Various macros can be used here. Click the script's left arrow button to select them.

%FileName% - Name of file  
 %WindowsFolder% - Windows path to folder  
 %WindowsPath% - Full Windows path of file  
 %VirtualFolder% - Virtual path to folder  
 %VirtualPath% - Full virtual file-system path of file  
 %SiteName% - Name of site  
 %ServerName% - Name of server  
 %TransferStatus% - Transfer status (success/failure)  
 %Time% - Time of event  
 %Type% - Type of event  
 %ClientIP% - IP address of client  
 %LoginEmail% - Email address of logged-in user  
 %LoginFullName% - Full name of logged-in user  
 %LoginUserName% - Username of logged-in user  
 %OwnerEmail% - Email address of owner of folder  
 %OwnerFullName% - Full name of owner of folder  
 %OwnerUserName% - Username of owner of folder  
 %ProcessRule% - Name of this process-rule

14. Specify the working directory (optional).

Directory:  ...

15. Finally, click the 'Apply changes' button at the top right of the CompleteFTP Manager to save the changes.



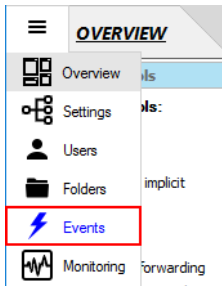
Now, every time a file is uploaded to the CompleteFTP server, the uploaded file will be moved to C:\FTP\Incoming.

# Add a scheduled task

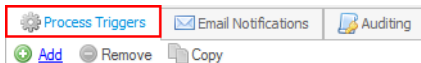
This guide demonstrates how to create an automated task with the CompleteFTP Manager. You can schedule a program to run, or execute a batch/Powershell/JSS/FTP script according to a specified scheduled or at specific times.

This guide will show you how to create a scheduled task in order to upload a file to another server, at a specific time, using FTP script. Here are the steps:

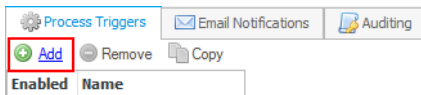
1. Select the 'Events' menu item from the side-bar menu.



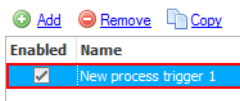
2. Select the 'Process Triggers' tab at the top of the panel.



3. Click the 'Add' button to create a new process trigger.



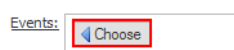
4. Select the process trigger you just created, it's probably called 'New process trigger 1'.



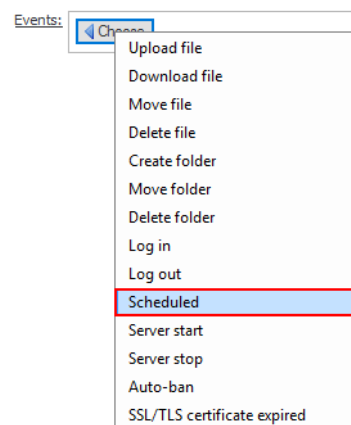
5. Enter an appropriate name for this process trigger, e.g. 'Upload schedule'.



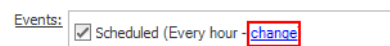
6. In the 'Events' section, click the 'Choose' button.



7. Select the 'Scheduled' event.



8. Click the 'change' link next to the 'Scheduled' checkbox to open the 'Set Schedule' dialog box.



9. Select the 'Specific dates/times' radio button.

Set Schedule

**Set schedule as:**

- ☒ Specific dates/times
- ☐ Even time intervals
- ☐ Days of the week
- ☐ Cron expression (advanced)

Trigger events at:

Time

Add

Delete

Next 50 events:

OK Cancel

10. Select the 'Add' button.

Set Schedule

**Set schedule as:**

- ☒ Specific dates/times
- ☐ Even time intervals
- ☐ Days of the week
- ☐ Cron expression (advanced)

Trigger events at:

Time

Add

Delete

Next 50 events:

OK Cancel

11. Set a date time when you want the file to be uploaded.

Set Schedule

**Set schedule as:**

- ☒ Specific dates/times
- ☐ Even time intervals
- ☐ Days of the week
- ☐ Cron expression (advanced)

Trigger events at:

Time

2017-02-28 12:30:00

Add

Delete

Next 50 events:

Tue 2017-02-28 12:30:00

OK Cancel

12. Close the 'Set Schedule' dialog box.

Set Schedule

**Set schedule as:**

☒ Specific dates/times

☐ Even time intervals

☐ Days of the week

☐ Cron expression (advanced)

Trigger events at:

Time

2017-02-28 12:30:00

Add

Delete

**Next 50 events:**

Tue 2017-02-28 12:30:00

OK

Cancel

13. Select the 'FTP script' radio button.

Type: ☐ Program ☐ Batch script ☐ Powershell script ☐ JSS script ☒ FTP script

14. Enter the following script which will upload the file, 'Process.docx', to the remote server (10.0.1.134).

Script:

```
set remotehost=10.0.1.134
set user=user1
set password=123456
set protocol=sftp

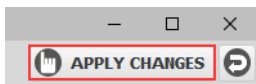
# connect to the server
open

# upload the file to the server
put C:\FTP\Upload\Process.docx Process.docx
# disconnect
close
```

15. Specify the working directory (optional).

Directory:

16. Finally, click the 'Apply changes' button in the top right of the CompleteFTP Manager to save the changes.

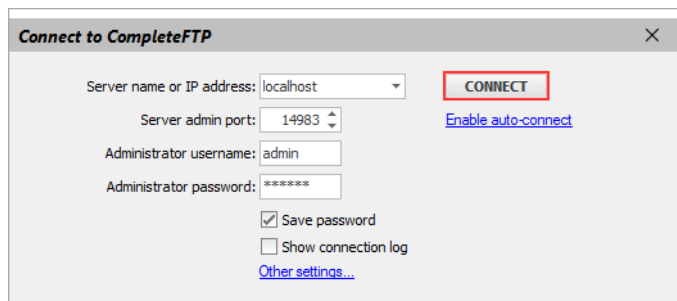


# Accepting connections from the Internet

Normally, when CompleteFTP is initially installed, it can only accept connections from users who are within the local network. This is because (1) the computer on which CompleteFTP is installed is behind a router, so connections requests that are made from the Internet never reach it, and (2) the Windows Firewall may be blocking incoming connections. These are normally desirable as they protect your computer, but sometimes you may need to allow users outside of the local network to connect. Let's say a CompleteFTP server has been set up at your workplace and you want to access it from anywhere via the internet to share files.

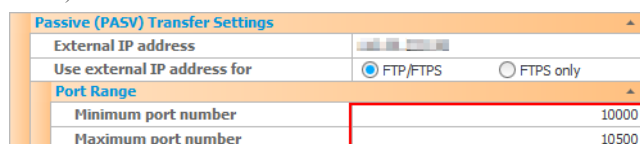
In order to do that, you may need to configure the router and the Windows firewall of the server's machine to allow external connections to your server. Here are the steps:

1. Open the CompleteFTP Manager and connect to the server.

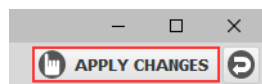


2. Make sure the required protocols are enabled in both 'Settings'/Sites' panel and 'Users' panel -> 'User Properties' of each user.
3. Configure the 'Passive (PASV) transfer settings'. Skip this step if neither FTP nor FTPS are enabled.
  - a. In 'Settings'/Sites' panel, go to 'FTP/FTPS' -> 'Advanced FTP/FTPS Settings' -> 'Passive (PASV) Transfer Settings'.
  - b. Limit the number of ports for passive mode FTP.

The number of ports in the range should be of the same order of magnitude as the maximum total number of files that clients will be transferring at any one time. If that maximum is, say, 10, then a port-range of 20 would suffice, but since there are over 60,000 ports available we may as well make it 1000. The port range should be chosen so that it doesn't clash with those used by other services on the computer. This can be checked in the 'TCP Port Usage' tab of the 'Monitoring' section (look in the 'Local Port' column).



- c. Click the 'Apply changes' button.



4. Configure port forwarding on the router.

If your CompleteFTP server is behind a router, you need to access your router's configuration page and configure your router to forward the following ports to your CompleteFTP server.

- FTP/Explicit FTPS - port 21.
- Legacy or implicit FTPS - port 990.
- SFTP/SCP/SSH - port 22.
- HTTP - port 80.
- HTTPS - port 443.

In addition, if FTP or FTPS are enabled then connections to all ports in the range chosen in step 3b) must also be forwarded to the server.

The way to configure port forwarding on the router depends on the type of router installed, so you should consult your router's documentation for appropriate steps.

5. Configure the Windows firewall on the server's machine.

You need to create an inbound firewall rule to allow connections to the ports on which CompleteFTP server is listening (these ports are same as ports in step 4). And also, don't forget to set the passive (or PASV) port range in the firewall to permit users to connect in this range, in case FTP or FTPS are enabled.

The steps to configure the firewall depend on the specific firewall being used. Refer to the documentation of the firewall for specific instructions.

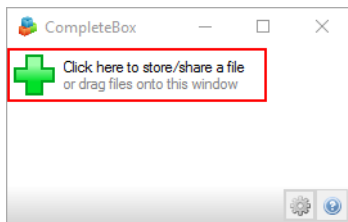


# Share a file using CompleteBox

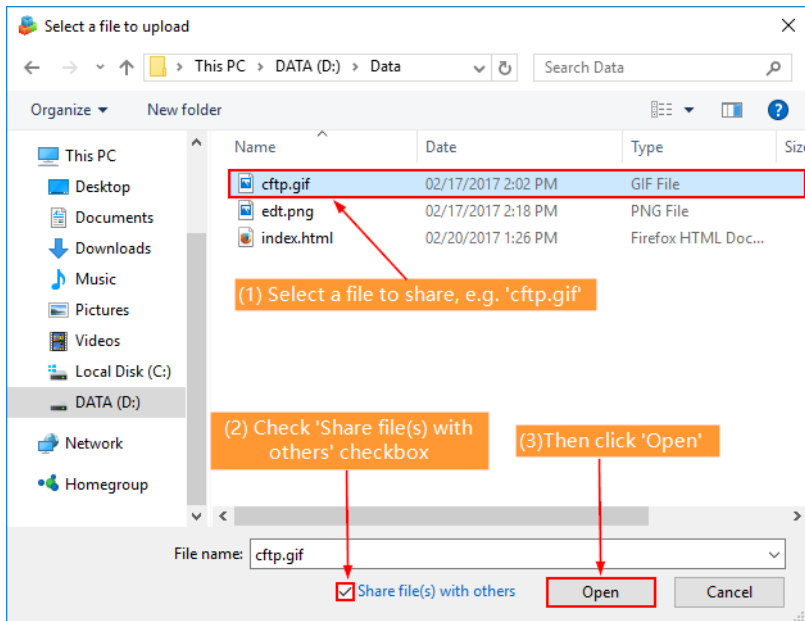
Before getting started, ensure that CompleteBox has been **installed** on the machine that files are being shared from, and that it has been **enabled** on your CompleteFTP server.

To share a file follow these steps:

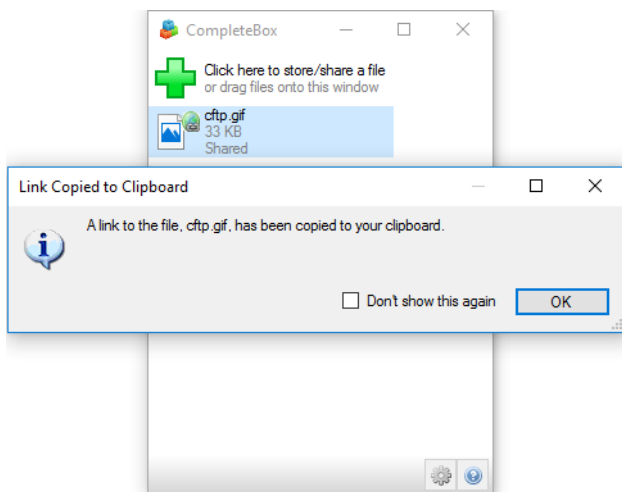
1. Open the CompleteBox dialog by selecting the CompleteBox icon in the Windows system tray. Then click the green cross to start sharing a file.



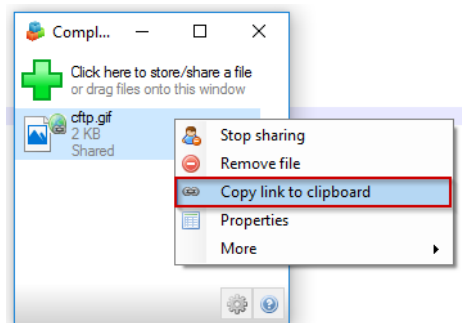
2. In the 'Select a file to upload' dialog box, do the following.



3. Once selected, the files will be shared on the file sharing server via a secure connection. A public link to the file is generated and automatically copied to the clipboard. You can now paste the link into an e-mail or a chat client, as required.



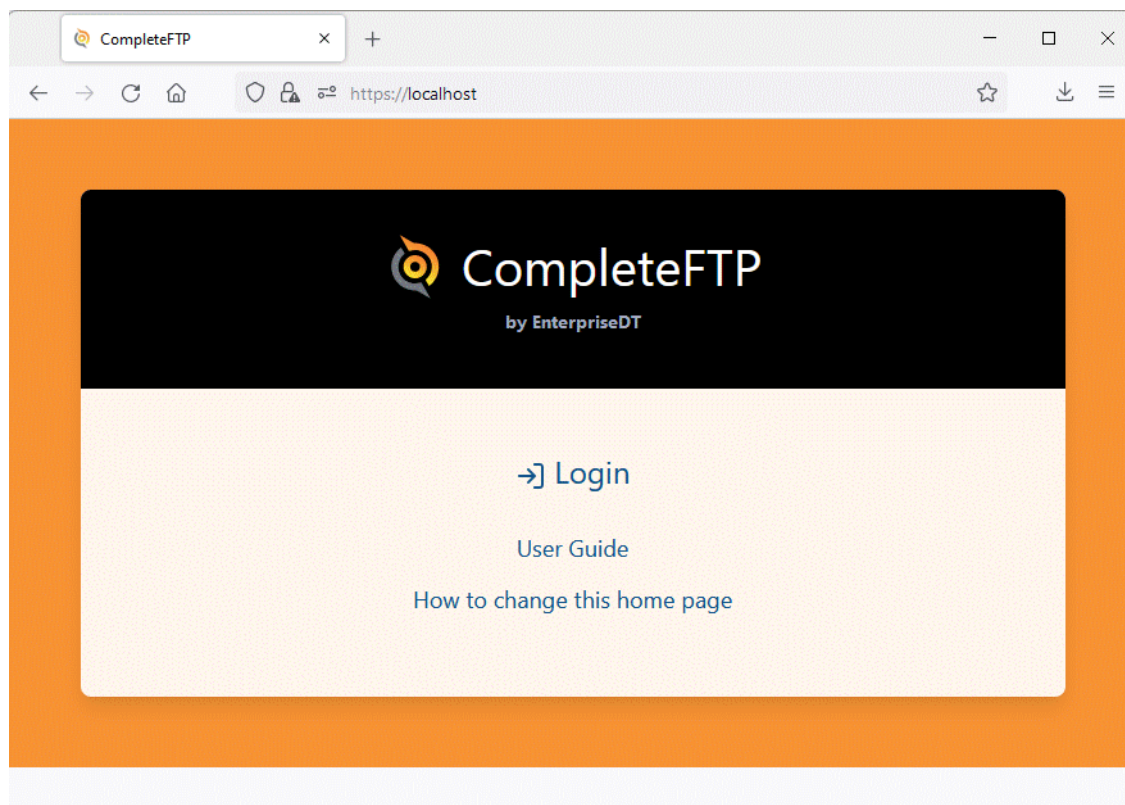
4. If, at a later time, you need to retrieve the link you can click on the file in CompleteBox and select 'Copy link to clipboard'.





# Customize the welcome page

CompleteFTP's default home page (below) is displayed when a user first navigates to the server via a web browser.



To modify the text or replace the home page, follow these steps:

1. Create or locate the customization instructions file at the path:

```
C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Include\custom\customizations.js
```

2. To modify the text and logo of the existing home page, use the variable assignments as illustrated below:

```
publicPageContent.homeURL = "https://mycompany.com";
publicPageContent.logo = "https://mycompany.com/logo.jpg";
publicPageContent.title = "My Company";
publicPageContent.subTitle = "";
publicPageContent.login = "Sign In";
publicPageContent.links = [
  { url: "https://mycompany.com/instructions.html", text: "Usage Instructions" },
  { url: "https://mycompany.com/privacy.html", text: "Privacy Policy" }
];
```

3. Alternatively, replace the home page entirely by setting the variable `publicPageTemplate` to the path of your custom HTML file:

```
publicPageTemplate = "../Include/custom/homePage.html";
```

The resulting file path will be:

```
C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Include\custom\homePage.html
```

Ensure this HTML file has a link to `/Login` or use the provided Javascript redirect code:

```
<html>
<body>
  <script type = "text/javascript">
    window.location = "/Login";
  </script>
</body>
</html>
```

4. The login and logout pages can be customized using `loginPageTemplate` and `logoutPageTemplate` variables, respectively. Ensure that customized templates work reliably,

especially after CompleteFTP updates.

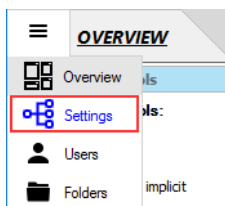
For more detailed instructions, please refer to Step-by-step guide: [Customize the web home page](#)

# Host a web-app in CompleteFTP

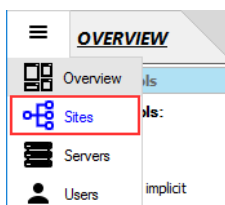
This tutorial will give you two simple examples on how to host web-applications in CompleteFTP.

Before getting started, you need to enable the 'Server-side Javascript (JSS)' property, on the site which is serving up the page.

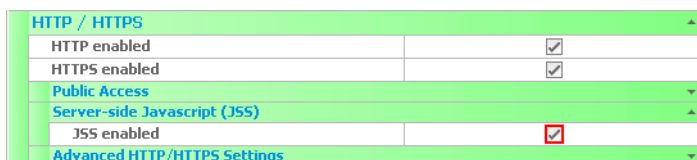
1. From the side-bar menu, select the 'Settings' panel (Professional Edition)



OR select the 'Sites' panel (Enterprise MFT).

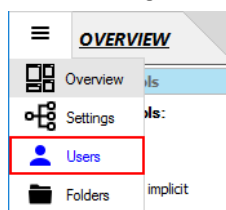


2. Enable JSS for the site by checking the 'JSS enabled' checkbox in the Site Settings HTTP/HTTPS category.



3. Enable JSS for the user who is the owner of the folder, into which the JSS file will be placed, as follows:

- a. Select the 'Users' panel from the side-bar menu.

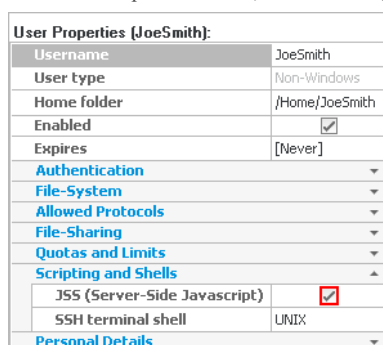


- b. Select the user whose home folder will contain the jss file, e.g. 'JoeSmith'.

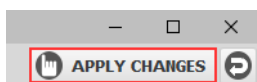
<a href="#">Add user</a> <a href="#">Remove</a> <a href="#">General user settings</a>					<b>User Properties (JoeSmith):</b>	
Enabled	Username	Type	Home Folder	Groups	Username	JoeSmith
<input checked="" type="checkbox"/>	FredJones	Non-Windows	/Home/FredJones	users	User type	Non-Windows
<input checked="" type="checkbox"/>	JoeSmith	Non-Windows	/Home/JoeSmith	users	Home folder	/Home/JoeSmith
					Enabled	<input checked="" type="checkbox"/>

Note that the owner of the /Public folder is the user called 'anonymous', which is hidden by default. To show it, check the 'Show system users/folder/sites' item in the View menu at the bottom left.

- c. In the 'User Properties' window, check the 'JSS(Server-side Javascript)' checkbox.



4. Click the 'Apply changes' button at the top right of the CompleteFTP Manager.



Now, let's create our own web-application in Javascript.

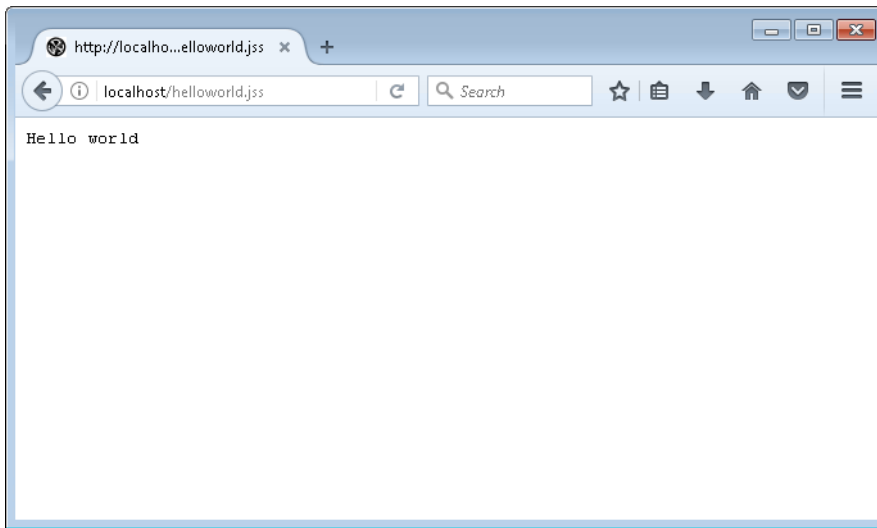
## Example 1

In this example, we create a web-app that simply displays 'Hello world' and is available publicly.

1. Create a text file called `helloworld.jss` with the following content.

```
response.write("Hello world");
```

2. Save it in `C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Public`.
3. Since 'anonymous' is the owner of `/Public`, JSS must be enabled for that user, as described above.
4. Now navigate to <http://localhost/helloworld.jss>. A web-page simply saying 'Hello world' will show up.



## Example 2

In this example, we're going to create a `.jss` file in a user's home directory, and execute it as a JSS web application. This web-app will be available only to the user.

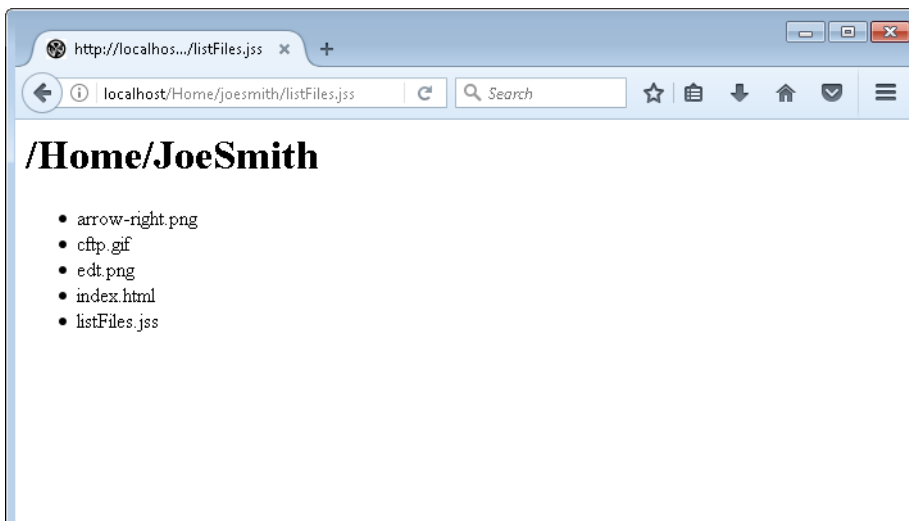
1. Create a file called `listFiles.jss` with the following content:

```
response.write("<h1>" + system.user.homeFolder + "</h1>");
response.write("");

var homeFolder = system.getFile(system.user.homeFolder)
var files = homeFolder.GetFiles();
files.forEach(function(file) {
    response.write("" + file.name + "")
});

response.write("");
```

2. Save it in JoeSmith's home directory, i.e. `C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Users\JoeSmith`.
3. Enable JSS for the JoeSmith user, as described above.
4. Then go to the URL <http://localhost/Home/joesmith/listFiles.jss>, where you should see a listing of all the files in that directory.

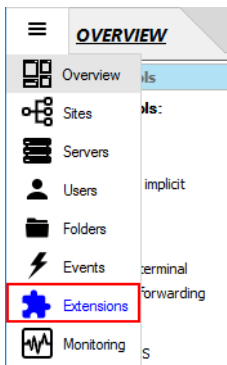


For more information on developing the JSS web-app, please refer to [JSS Web-App Basics](#) and the [JSS API Guide](#).

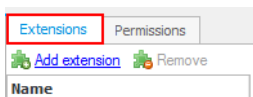
# Add my own authentication method

This example will show you how to create your own authentication method using **JSS**. It's also possible to implement **authentication extensions in .NET**.

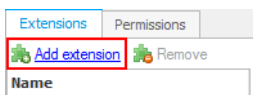
1. Select the 'Extensions' panel from the side-bar menu.



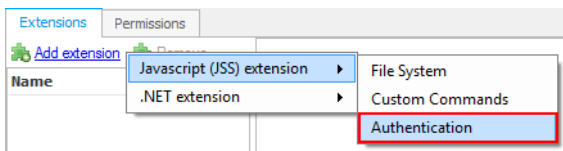
2. Select the 'Extensions' tab at the top of the panel.



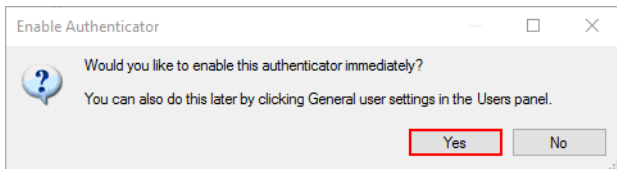
3. Click the 'Add extension' button.



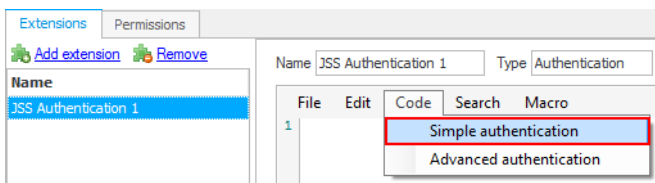
4. Select 'Javascript (JSS) extension' then 'Authentication' menu item.



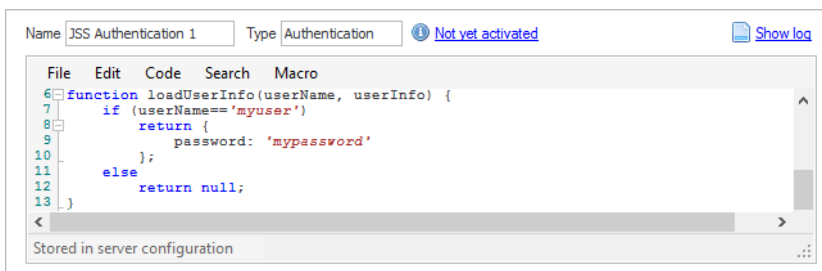
5. In the 'Enable Authenticator' dialog. Click the 'Yes' button to enabled this authenticator.



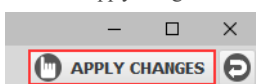
6. In the code editor, select 'Code' then 'Simple authentication' menu item.



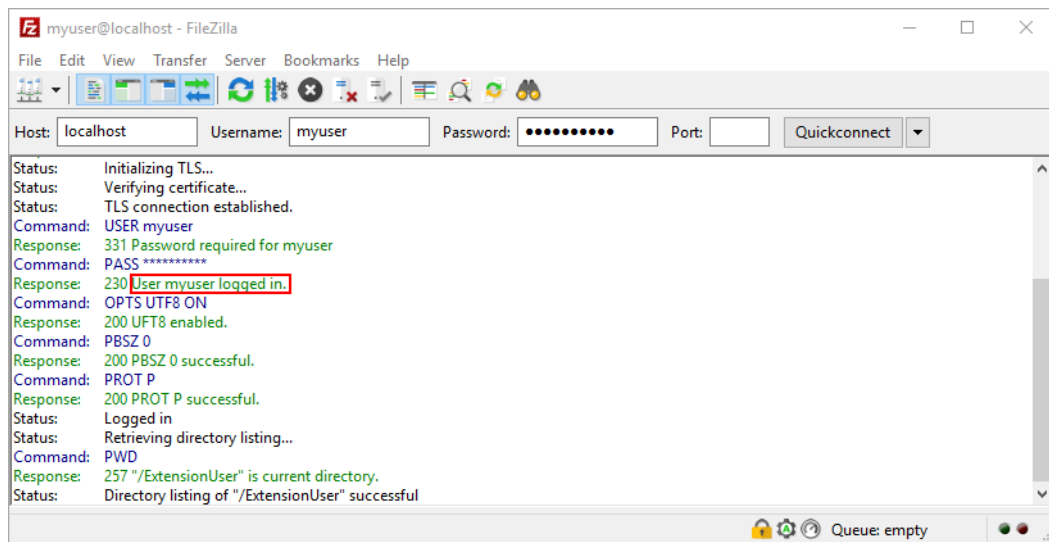
7. Then enter your **JSS** code. The example is shown below.



8. Click the 'Apply changes' button in the top right of the CompleteFTP Manager.



Now, try to log in using the username 'myuser' and the password 'mypassword', you should be able to log in successfully.

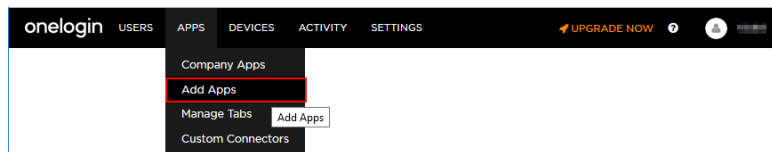




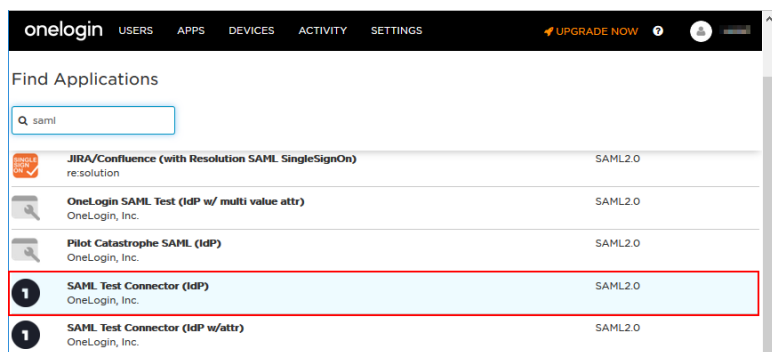
# Configure SAML Single Sign On with OneLogin

This guide illustrates how to configure **CompleteFTP** as a SAML Service Provider (SP) with **OneLogin** as an Identity Provider (IDP). Please note that this requires Enterprise MFT.

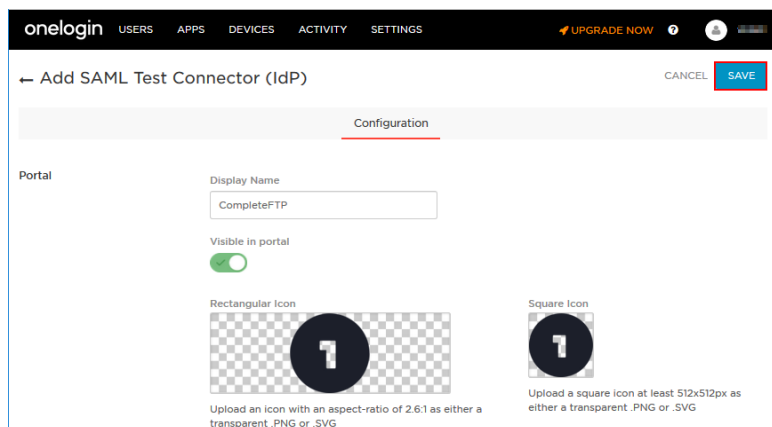
1. Configure OneLogin as the Identity Provider (IDP).
  - a. Create a free account at **OneLogin** and log in.
  - b. Navigate to Apps → Add Apps.



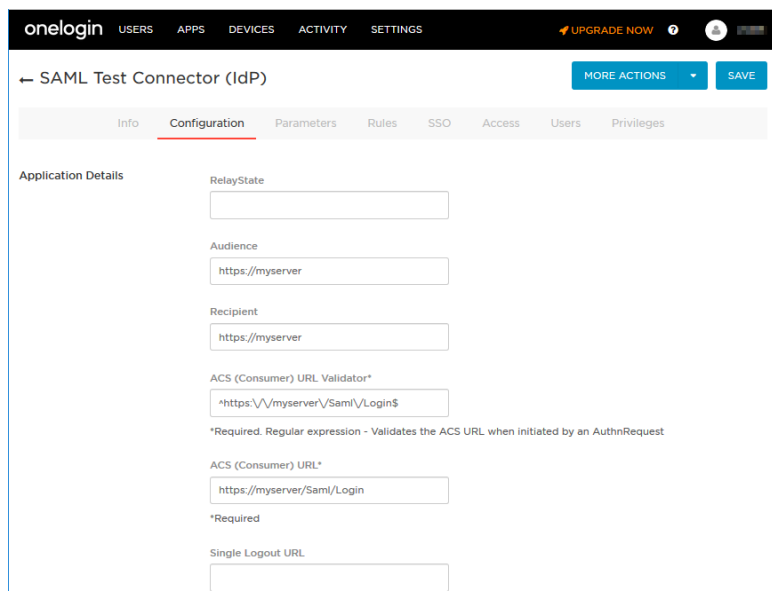
- c. Enter 'saml' into the search box, then pick 'SAML Test Connector (IdP)' from the list.



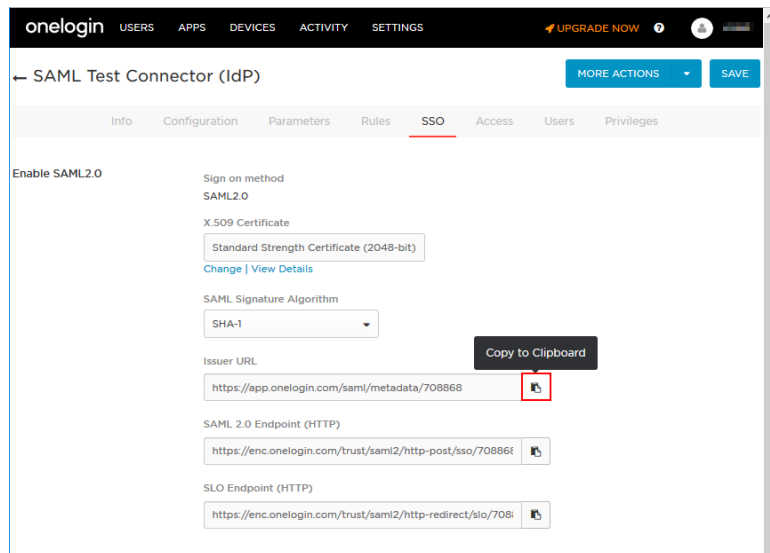
- d. Enter the app's name into the 'Display Name' field (e.g. CompleteFTP) then click the 'Save' button.



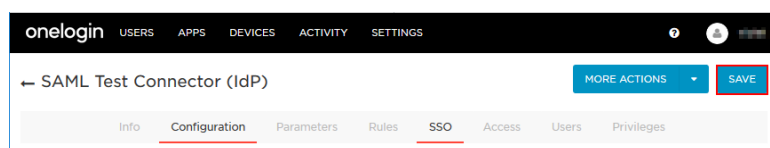
- e. Go to the 'Configuration' tab and...
    - i. Enter your CompleteFTP server's address into 'Audience' and 'Recipient' fields. This URL will also be used in **step 2e** below.
    - ii. Append '/Saml/Login' to the site's URL and put the whole URL, i.e. `https://myserver/Saml/Login`, into the 'ACS (Consumer) URL' field.
    - iii. Convert the 'ACS (Consumer) URL' into a regular expression and put that into 'ACS (Consumer) URL Validator' field. Note that all dots and backslashes in the URL must be preceded by backslashes. For more information, please visit [this link](#).



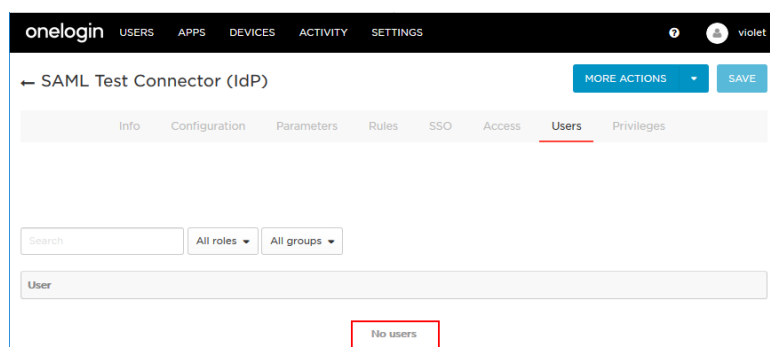
f. Next, go to the SSO tab, copy the Issuer URL to the clipboard for later use. This is the URL that the IDP's metadata file can be downloaded from.



g. Click the 'Save' button in the top right corner to save the configuration.

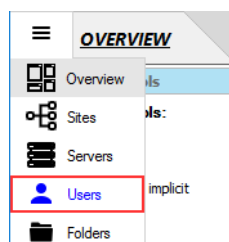


h. The app is now created but no one has been given permissions to access it (see below). Please visit [this link](#) for detailed instructions on how to give users access to the app.



## 2. Configure CompleteFTP as the Service Provider (SP)

a. Open the CompleteFTP Manager, select the 'Users' panel from the side-bar menu.



b. Enable the Single sign-on/SAML authenticator by checking its 'Enabled' checkbox.

[Add user](#)
[Remove](#)
[General user settings](#)

Enabled	Username	Type	Home Folder	Groups
<input checked="" type="checkbox"/>	u1	Non-Windows	/Home/u1	users

[Select](#)
[Groups](#)
[Two-factor auth status](#)

### AUTHENTICATORS

For site: Default Site

Authenticator	Enabled	Log in as	
Database	<input type="checkbox"/>	defaultDatabase	<a href="#">Configure...</a>
Gateway	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Single sign-on / SAML	<input checked="" type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Windows/AD Group	<input type="checkbox"/>	defaultWindows	<a href="#">Configure...</a>

- c. Click the 'Configure' link to open the 'Single sign-on/SAML Configuration' dialog

[Add user](#)
[Remove](#)
[General user settings](#)

Enabled	Username	Type	Home Folder	Groups
<input checked="" type="checkbox"/>	u1	Non-Windows	/Home/u1	users

[Select](#)
[Groups](#)
[Two-factor auth status](#)

### AUTHENTICATORS

For site: Default Site

Authenticator	Enabled	Log in as	
Database	<input type="checkbox"/>	defaultDatabase	<a href="#">Configure...</a>
Gateway	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Single sign-on / SAML	<input checked="" type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Windows/AD Group	<input type="checkbox"/>	defaultWindows	<a href="#">Configure...</a>

- d. Enter the site's URL. This is the URL of the CompleteFTP server's site on which the Single sign-on/SAML authenticator is enabled.  
Note that if the site is configured to use a non-standard port (i.e. 80 for HTTP and 443 for HTTPS) then the site URL must include the port number (e.g. https://myserver:1443).

Single sign-on / SAML Configuration

Identity Providers (IDP)

Name	ID	User attribute	User prefix
<a href="#">Add IDP</a> <a href="#">Remove</a>			

☐ Validate certificates against the Windows Certificate Store

This site's URL

**Administrative contact**

First name: 
 Last name: 
 Company:

Email address: 
 Phone number:

[Save Service Provider \(SP\) metadata](#)
[Edit raw XML](#)

**Identity Providers (IDPs)**

Identity Providers (IDPs) are servers that are responsible for authenticating users and for sending information to CompleteFTP stating that a user should be allowed to log in. Before CompleteFTP can interoperate with an IDP the IDP's metadata must be added to this list. An IDP is added by means of its metadata, which contains required information about the IDP, including its URL. In addition to the metadata, three pieces of information is required by CompleteFTP:

(1) The IDP's name. This is simply a label that's used on CompleteFTP's login form to tell users that they can log in via that particular IDP.

(2) The user attribute. Each time a user has authenticated via an IDP, the IDP sends a message about the user to CompleteFTP. This packet contains one or more attributes that describe the user. The exact attributes that are provided depends on the configuration of the IDP. CompleteFTP uses one of the attributes as the user-name. The user-name attribute field tells CompleteFTP which attribute should be used. Note that if the field is left blank and the server sends only one attribute then that attribute will be used. If more than one attribute is sent but none match the

OK Cancel

- e. Enter the administrative contact details.  
Note that all fields are mandatory.

Single sign-on / SAML Configuration

Identity Providers (IDP)

Name	ID	User attribute	User prefix

Add IDP
 Remove

☐ Validate certificates against the Windows Certificate Store

This site's URL

https://myserver

**Administrative contact**

First name: My first name    Last name: My last name    Company: My company

Email address: email@domain.com    Phone number: +61123456789

Save Service Provider (SP) metadata    Edit raw XML

**Identity Providers (IDPs)**

Identity Providers (IDPs) are servers that are responsible for authenticating users and for sending information to CompleteFTP stating that a user should be allowed to log in. Before CompleteFTP can interoperate with an IDP the IDP's metadata must be added to this list. An IDP is added by means of its metadata, which contains required information about the IDP, including its URL. In addition to the metadata, three pieces of information is required by CompleteFTP:

(1) The IDP's name. This is simply a label that's used on CompleteFTP's login form to tell users that they can log in via that particular IDP.

(2) The user attribute. Each time a user has authenticated via an IDP, the IDP sends a message about the user to CompleteFTP. This packet contains one or more attributes that describe the user. The exact attributes that are provided depends on the configuration of the IDP. CompleteFTP uses one of the attributes as the user-name. The user-name attribute field tells CompleteFTP which attribute should be used. Note that if the field is left blank and the server sends only one attribute then that attribute will be used. If more than one attribute is sent but none match the

OK    Cancel

f. Now, register OneLogin as an IDP in CompleteFTP by clicking 'Add IDP' link.

Single sign-on / SAML Configuration

Identity Providers (IDP)

Name	ID	User attribute	User prefix

Add IDP
 Remove

☐ Validate certificates against the Windows Certificate Store

This site's URL

https://myserver

**Administrative contact**

First name: My first name    Last name: My last name    Company: My company

Email address: email@domain.com    Phone number: +61123456789

Save Service Provider (SP) metadata    Edit raw XML

**Identity Providers (IDPs)**

Identity Providers (IDPs) are servers that are responsible for authenticating users and for sending information to CompleteFTP stating that a user should be allowed to log in. Before CompleteFTP can interoperate with an IDP the IDP's metadata must be added to this list. An IDP is added by means of its metadata, which contains required information about the IDP, including its URL. In addition to the metadata, three pieces of information is required by CompleteFTP:

(1) The IDP's name. This is simply a label that's used on CompleteFTP's login form to tell users that they can log in via that particular IDP.

(2) The user attribute. Each time a user has authenticated via an IDP, the IDP sends a message about the user to CompleteFTP. This packet contains one or more attributes that describe the user. The exact attributes that are provided depends on the configuration of the IDP. CompleteFTP uses one of the attributes as the user-name. The user-name attribute field tells CompleteFTP which attribute should be used. Note that if the field is left blank and the server sends only one attribute then that attribute will be used. If more than one attribute is sent but none match the

OK    Cancel

g. That brings up the 'Import IDP Metadata' dialog. Choose the 'Website' button.

Single sign-on / SAML Configuration

Identity Providers (IDP)

Name	ID	User attri...	User prefix

Add IDP
 Remove

☐ Validate certificates against the Windows Certificate Store

This site's URL

https://myserver

**Administrative contact**

First name: My first name    Last name: My last name    Company: My company

Email address: email@domain.com    Phone number: +61123456789

Save Service Provider (SP) metadata    Edit raw XML

**Add IDP**

Adding an IDP involves importing the IDP's metadata, setting the attribute to use for the user-name and giving it a name that will appear on the CompleteFTP login form.

The IDP metadata may be downloaded from the IDP website, or requested from the IDP server administrators. CompleteFTP Manager is able to load the IDP metadata from a file or from a website. A URL is required. A URL is a string that identifies the location of a resource on the Internet. A URL is a string that identifies the location of a resource on the Internet. A URL is a string that identifies the location of a resource on the Internet.

OK    Cancel

**Import IDP Metadata**

Would you like to load the IDP metadata from a file or download it from a website?

File    Website    Cancel

h. Paste the URL copied in step 1f into the text-field and click 'OK' button.

Single sign-on / SAML Configuration

Identity Providers (IDP)

Name	ID	User attri...	User prefix
<div> Add IDP Remove </div>			

☐ Validate certificates against the Windows Certificate Store  
This site's URL  
https://myserver

Administrative contact  
First name: My first name   Last name: My last name   Company: My company  
Email address: email@domain.com   Phone number: +61123456789

Save Service Provider (SP) metadata   Edit raw XML

**Add IDP**

Adding an IDP involves importing the IDP's metadata, setting the attribute to use for the user-name and giving it a name that will appear on the CompleteFTP login form.

The IDP metadata may be downloaded from the IDP website, or requested from the IDP server administrators. CompleteFTP Manager is able to load it directly from a URL. After the Add IDP link is clicked choose which method you'd

Import IDP Metadata from Website

URL of IDP metadata  
https://app.onelogin.com/saml/metadata/720080

OK   Cancel

- i. A row should have been added to the list. Let's enter a name for it (e.g. OneLogin) into the 'Name' field.

Single sign-on / SAML Configuration

Identity Providers (IDP)

Name	ID	User attri...	User prefix
OneLogin	https://app.onel...		

Add IDP   Remove

☐ Validate certificates against the Windows Certificate Store  
This site's URL  
https://myserver

Administrative contact  
First name: My first name   Last name: My last name   Company: My company  
Email address: email@domain.com   Phone number: +61123456789

Save Service Provider (SP) metadata   Edit raw XML

**Identity Providers (IDPs)**

Identity Providers (IDPs) are servers that are responsible for authenticating users and for sending information to CompleteFTP stating that a user should be allowed to log in. Before CompleteFTP can interoperate with an IDP the IDP's metadata must be added to this list. An IDP is added by means of its metadata, which contains required information about the IDP, including its URL. In addition to the metadata, three pieces of information is required by CompleteFTP:

(1) The IDP's name. This is simply a label that's used on CompleteFTP's login form to tell users that they can log in via that particular IDP.

(2) The user attribute. Each time a user has authenticated via an IDP, the IDP sends a message about the user to CompleteFTP. This packet contains one or more attributes that describe the user. The exact attributes that are provided depends on the configuration of the IDP. CompleteFTP uses one of the attributes as the user-name. The user-name attribute field tells CompleteFTP which attribute should be used. Note that if the field is left blank and the server sends only one attribute then that attribute will be used. If more than one attribute is sent but none match the

OK   Cancel

- j. Close the 'Single sign-on/SAML Configuration' dialog.

Single sign-on / SAML Configuration

Identity Providers (IDP)

Name	ID	User attri...	User prefix
OneLogin	https://app.onel...		

Add IDP   Remove

☐ Validate certificates against the Windows Certificate Store  
This site's URL  
https://myserver

Administrative contact  
First name: My first name   Last name: My last name   Company: My company  
Email address: email@domain.com   Phone number: +61123456789

Save Service Provider (SP) metadata   Edit raw XML

**Identity Providers (IDPs)**

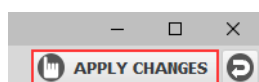
Identity Providers (IDPs) are servers that are responsible for authenticating users and for sending information to CompleteFTP stating that a user should be allowed to log in. Before CompleteFTP can interoperate with an IDP the IDP's metadata must be added to this list. An IDP is added by means of its metadata, which contains required information about the IDP, including its URL. In addition to the metadata, three pieces of information is required by CompleteFTP:

(1) The IDP's name. This is simply a label that's used on CompleteFTP's login form to tell users that they can log in via that particular IDP.

(2) The user attribute. Each time a user has authenticated via an IDP, the IDP sends a message about the user to CompleteFTP. This packet contains one or more attributes that describe the user. The exact attributes that are provided depends on the configuration of the IDP. CompleteFTP uses one of the attributes as the user-name. The user-name attribute field tells CompleteFTP which attribute should be used. Note that if the field is left blank and the server sends only one attribute then that attribute will be used. If more than one attribute is sent but none match the

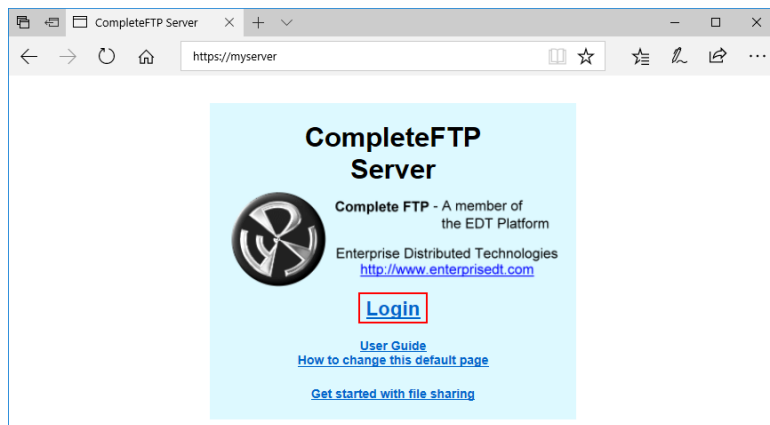
OK   Cancel

- k. Click the apply changes button.

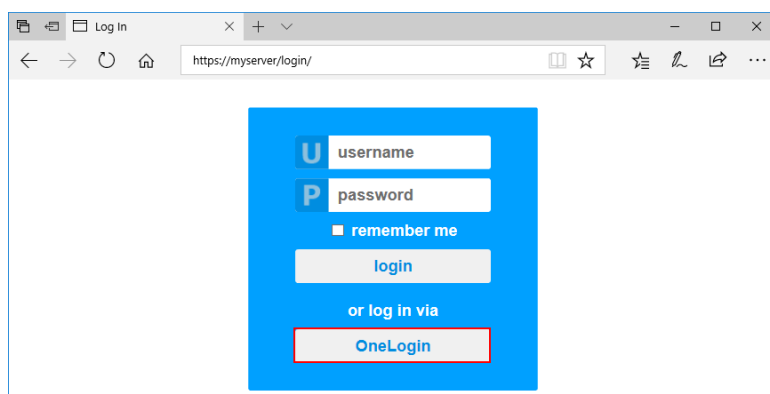


3. Alright, OneLogin should now be configured as an IDP for your server. Now, let's see how it works.

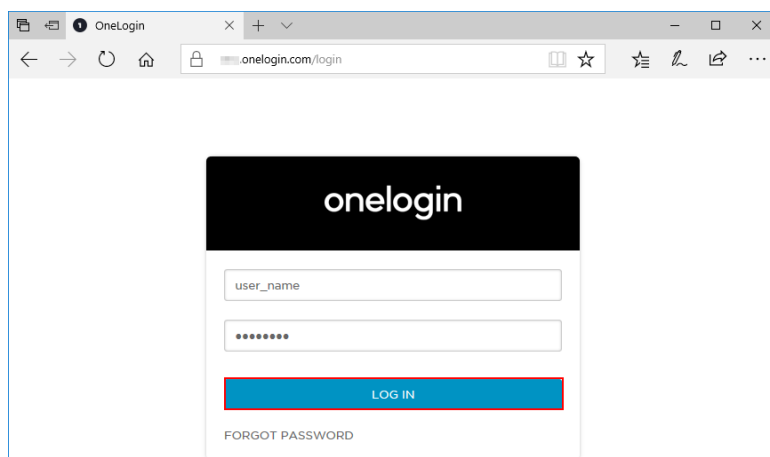
a. Navigate to your CompleteFTP server via HTTPS protocol and click the 'Login' link.



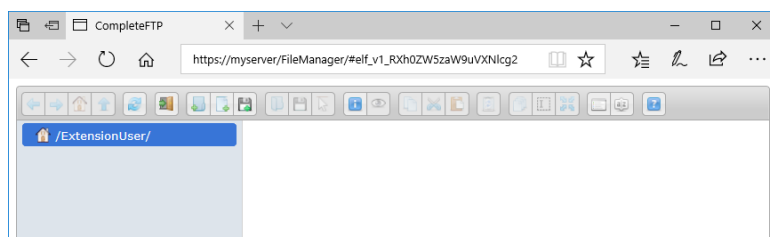
b. The CompleteFTP login page is displayed. Let's login via OneLogin by clicking the OneLogin button.



c. You will be taken to the OneLogin login page. Now, enter the credentials of a OneLogin user who has been given permissions to access the OneLogin's app in [step 1h](#) above. Then click 'Log in'.



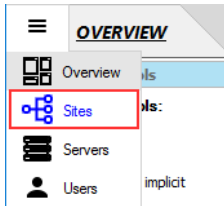
d. Once you've logged in successfully, you'll be redirected back to CompleteFTP where the File Manager will display.



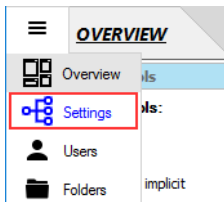
# Request a CA certificate

In order to purchase a CA SSL certificate from a CA (Certificate Authority), a Certificate Signing Request (CSR) must be generated. This guide will demonstrate how to generate a CSR using the CompleteFTP Manager.

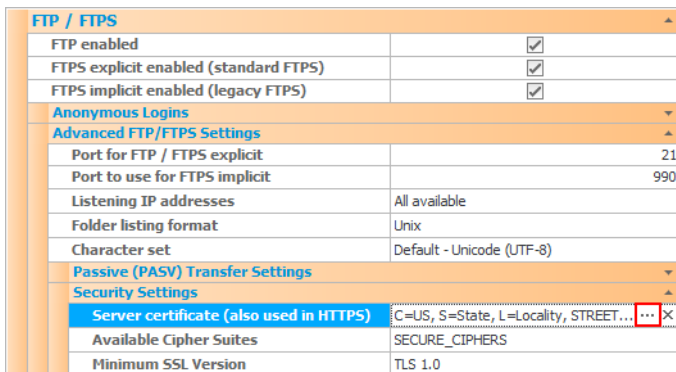
1. From the side-bar menu, select the 'Sites' panel (Enterprise MFT):



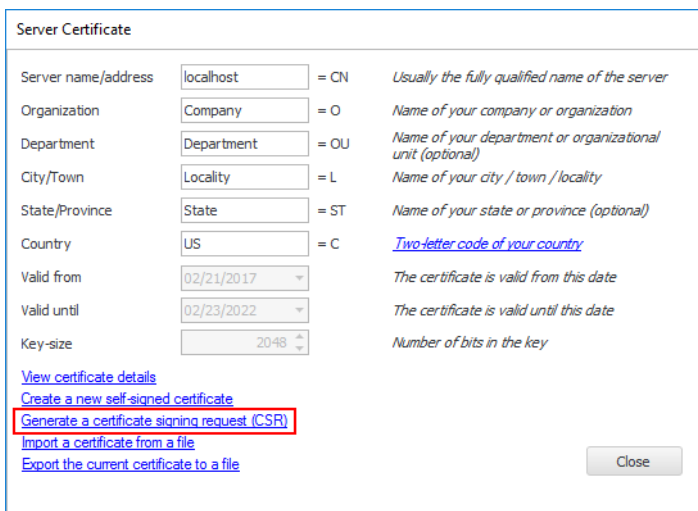
OR the 'Settings' panel (other Editions)



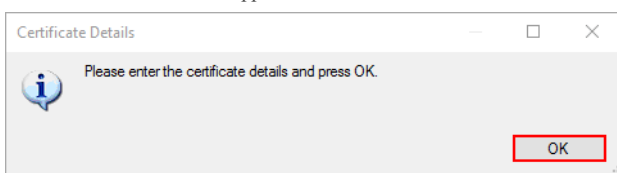
2. Click 'FTP/FTPS' -> 'Advanced FTP/FTPS Settings' -> 'Security Settings' -> 'Server certificate' then select the ellipsis (...) button.



3. In the 'Server Certificate' dialog box, click the 'Generate a certificate signing request (CSR)' link.



4. Choose OK when the notice appears.



5. Enter the certificate details and press OK.

Server Certificate

Server name/address	localhost	= CN	Usually the fully qualified name of the server
Organization	Company	= O	Name of your company or organization
Department	Department	= OU	Name of your department or organizational unit (optional)
City/Town	Locality	= L	Name of your city / town / locality
State/Province	State	= ST	Name of your state or province (optional)
Country	US	= C	<u>Two-letter code of your country</u>

View certificate details  
 Create a new self-signed certificate  
 Generate a certificate signing request (CSR)  
 Import a certificate from a file  
 Export the current certificate to a file

OK Cancel

6. Enter the private key password or click the 'Random' link to generate one randomly. Then press 'OK'.

Enter private key password

☐ Show characters

Password [Random](#)

\*\*\*\*\*

Password again

\*\*\*\*\*

OK Cancel

7. The private key is generated. Please make sure this key is saved to a secure location for use with the issued certificate. Do not lose it! Your certificate is of no use without the corresponding private key.

Enter name of private key file

« DATA ... » Server Certificates

Search Server Certificates

Organize New folder

Documents Downloads Music Pictures Videos

Name Date modified Type

No items match your search.

File name: PrivateKey.pvk

Save as type: RSA Private Key File (\*.pvk)

Hide Folders Save Cancel

8. The CSR file is generated. Enter the name for it and click 'Save'.

Enter name of certificate signing request file

« DATA ... » Server Certificates

Search Server Certificates

Organize New folder

Documents Downloads Music Pictures Videos

Name Date modified Type

No items match your search.

File name: CSR.csr

Save as type: Certificate signing request (\*.csr)

Hide Folders Save Cancel

9. Click OK to confirm the notice about CSR generated.

Certificate Signing Request Generated

The certificate signing request has been written to:  
D:\Server Certificates\CSR.csr

OK

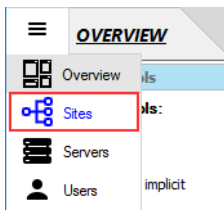


Now your private key and CSR file are ready to use. Your CA will request either the CSR file itself or its content.

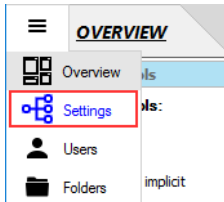
# Install a CA certificate

This guide demonstrates how you should install your SSL Certificate for CompleteFTP. Before installing the SSL Certificate, you first need to [create a Certificate Signing Request \(CSR\)](#).

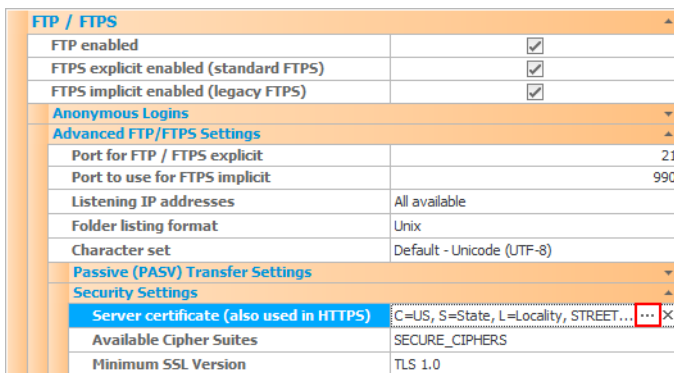
1. From the side-bar menu, select the 'Sites' panel (Enterprise MFT):



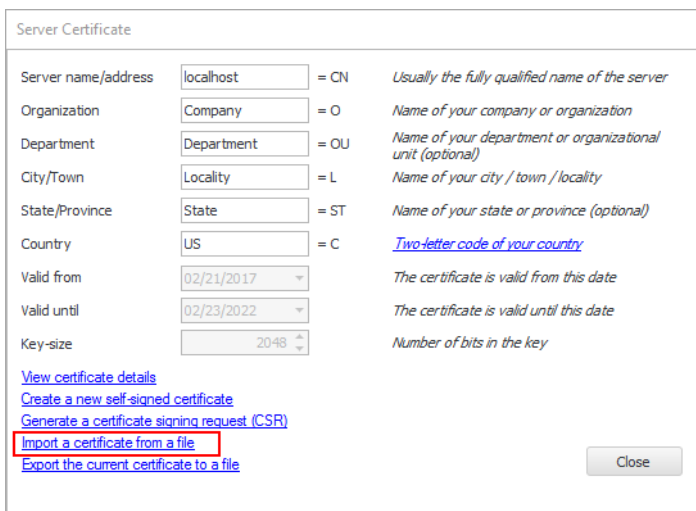
OR the 'Settings' panel (other Editions)



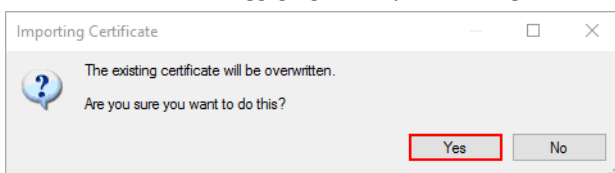
2. Click 'FTP/FTPS' -> 'Advanced FTP/FTPS Settings' -> 'Security Settings' -> 'Server certificate' then select the ellipsis (...) button



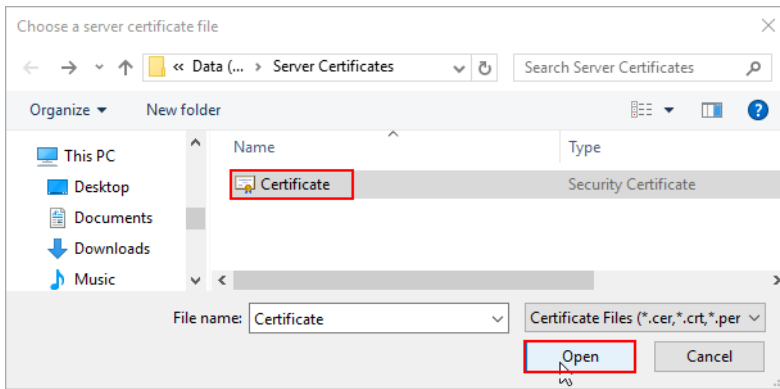
3. In the 'Server Certificate' dialog box, select the 'Import a certificate from a file' link.



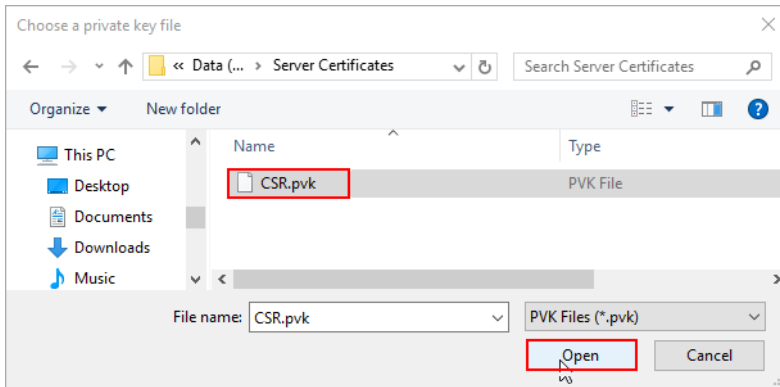
4. Choose 'Yes' when the Warning pops up which says : 'The existing certificate will be overwritten'.



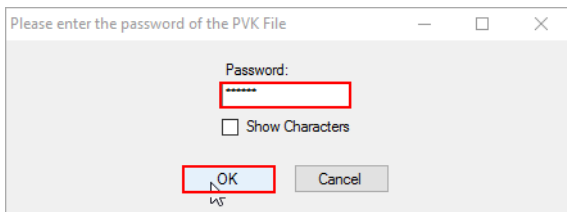
5. Choose the certificate file which contains the server certificate.



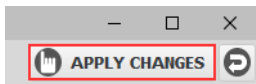
6. Choose the private key which was generated when you created the CSR file.



7. Then enter the password that you previously set when your private key had been generated and click OK.



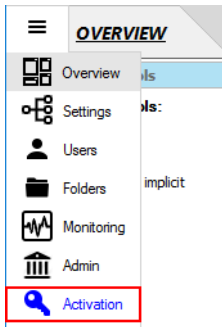
8. Finally, click the 'Apply changes' button to save the changes.



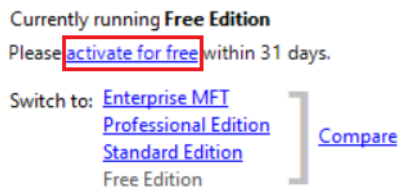
Your SSL Certificate is now installed and ready to use.

# Activate CompleteFTP free

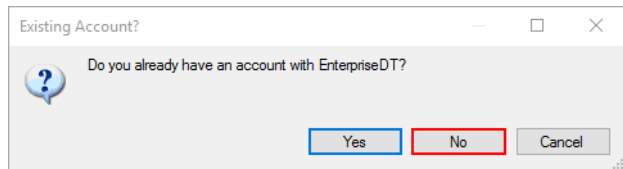
1. Select the 'Activation' panel from the side-bar menu.



2. Click the 'activate for free' link.



3. If you do not have an account with us at EnterpriseDT then take the following steps to create a new one:
  - a. Choose 'No' button.



- b. Enter a valid e-mail address, your name and, optionally, the name of your company.

A screenshot of a 'User Registration' form. It contains several input fields: 'Email address' (with the text 'myemail@domain.com'), 'First name' (with the text 'MyFirstName'), 'Last name' (with the text 'MyLastName'), and 'Company name' (with the text 'My Company'). Each of these four input fields is highlighted with a red rectangular box. There is also a 'Request verification code' button and a 'Verification code' input field at the bottom. The form has 'Register' and 'Cancel' buttons at the very bottom.

- c. Click the 'Request verification code' button. This will tell the EnterpriseDT server to send a verification code to the e-mail address that you entered. It should arrive within a minute or so.

User Registration

Click [here](#) if this computer has no Internet connection.

Please enter your email address, name and company name below. Your free license will be registered with these details.

[Read our privacy policy here.](#)

Email address *Verification code will be sent here*  
myemail@domain.com

First name Last name  
MyFirstName MyLastName

Company name  
My Company

**Request verification code**

Verification code (sent to above e-mail address)

Register Cancel

- d. Copy the verification code from the e-mail to the clipboard.



Dear MyFirstName MyLastName,

Thank you for requesting an account with Enterprise Distributed Technologies Pty Ltd.

Your verification code is **123456**. Please enter this code into CompleteFTP Manager to complete account registration.

Regards

- e. Paste the code into the last field on the 'User Registration' form.

User Registration

Click [here](#) if this computer has no Internet connection.

Please enter your email address, name and company name below. Your free license will be registered with these details.

[Read our privacy policy here.](#)

Email address *Verification code will be sent here*  
myemail@domain.com

First name Last name  
MyFirstName MyLastName

Company name  
My Company

Request verification code

Verification code (sent to above e-mail address)  
**123456**

Register Cancel

- f. Click the 'Register' button.

User Registration

Click [here](#) if this computer has no Internet connection.

Please enter your email address, name and company name below. Your free license will be registered with these details.

[Read our privacy policy here.](#)

Email address *Verification code will be sent here*  
myemail@domain.com

First name Last name  
MyFirstName MyLastName

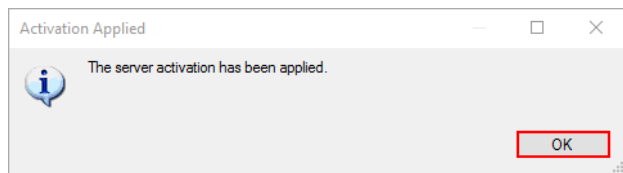
Company name  
My Company

Request verification code

Verification code (sent to above e-mail address)  
123456

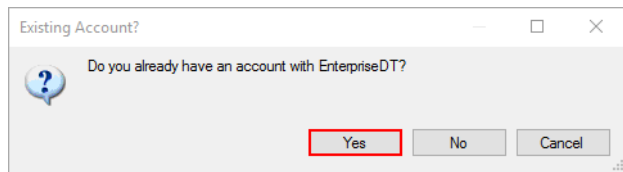
**Register** Cancel

- g. After a few seconds your account will have been registered and your server activated. Press the 'OK' button when the 'Activation Applied' dialog pops up.

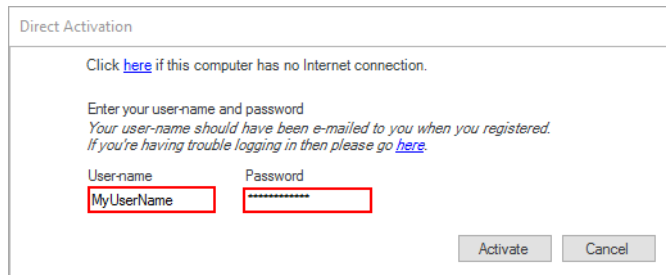


4. If you already have an account then take the following steps to activate a Free Edition server using your existing account.

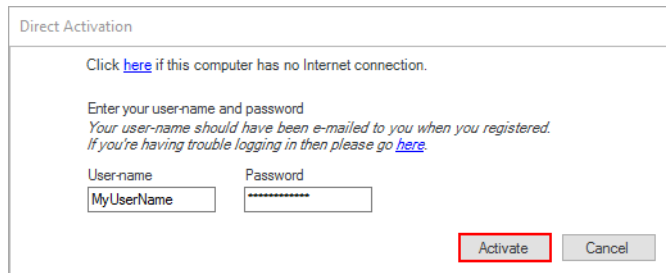
a. Choose the 'Yes' button.



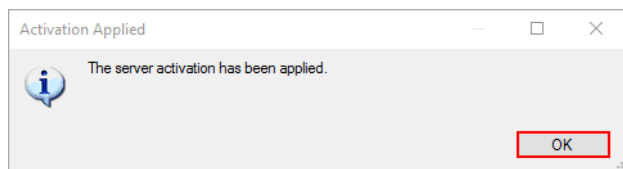
b. Enter the user-name which you received when you first registered your account and use the password which you have now set for this account.



c. Then Click the 'Activate' button. CompleteFTP Manager will automatically activate your server.



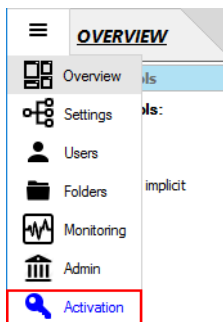
d. Now, your server should have been activated successfully.



## Activating the Free Edition without a direct Internet connection

In case the machine on which CompleteFTP Manager is running, doesn't have an Internet connection, then please follow the steps below:

1. Select the 'Activation' panel from the side-bar menu.



2. Click the 'activate for free' link.

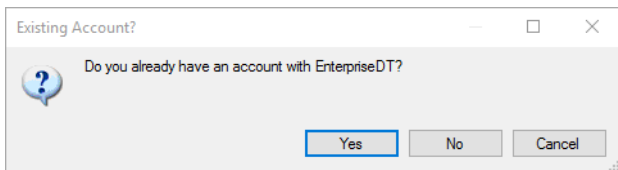
Currently running **Free Edition**

Please **activate for free** within 31 days.

Switch to: [Enterprise MFT](#)  
[Professional Edition](#)  
[Standard Edition](#)  
Free Edition

[Compare](#)

3. You will be asked whether or not you have an account with EnterpriseDT. Click 'Yes' or 'No'.

A dialog box titled "Existing Account?" with a question mark icon. It asks "Do you already have an account with EnterpriseDT?". There are three buttons: "Yes", "No", and "Cancel".

Existing Account?

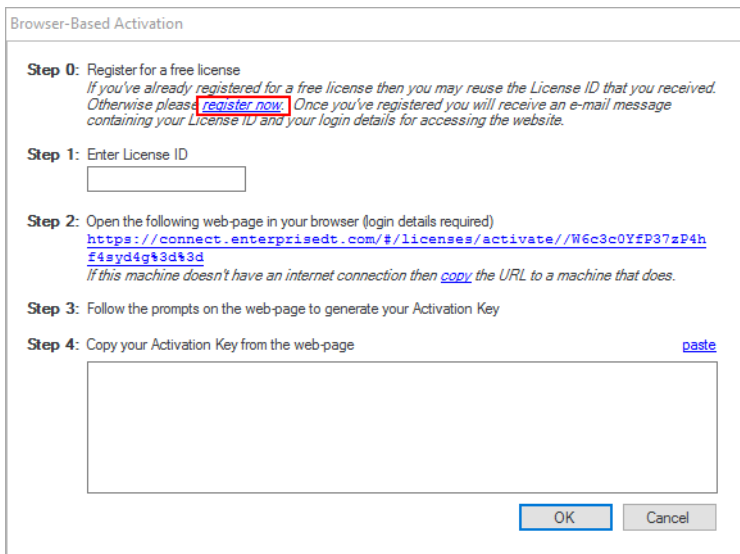
Do you already have an account with EnterpriseDT?

Yes No Cancel

4. In the dialog pops up, Click the 'here' link at the top to open the 'Browser-Based Activation' dialog.

Click **here** if this computer has no Internet connection.

5. If you've previously registered for a free license then use the License ID that was e-mailed to you. If you haven't previously registered, then follow the 'register now' link in step 0 to register you account and receive your License ID (by e-mail).

A dialog box titled "Browser-Based Activation" with a list of steps for license activation. Step 0: Register for a free license. Step 1: Enter License ID. Step 2: Open the following web-page in your browser. Step 3: Follow the prompts on the web-page to generate your Activation Key. Step 4: Copy your Activation Key from the web-page. There are "OK" and "Cancel" buttons at the bottom.

Browser-Based Activation

**Step 0:** Register for a free license  
*If you've already registered for a free license then you may reuse the License ID that you received. Otherwise please [register now](#). Once you've registered you will receive an e-mail message containing your License ID and your login details for accessing the website.*

**Step 1:** Enter License ID

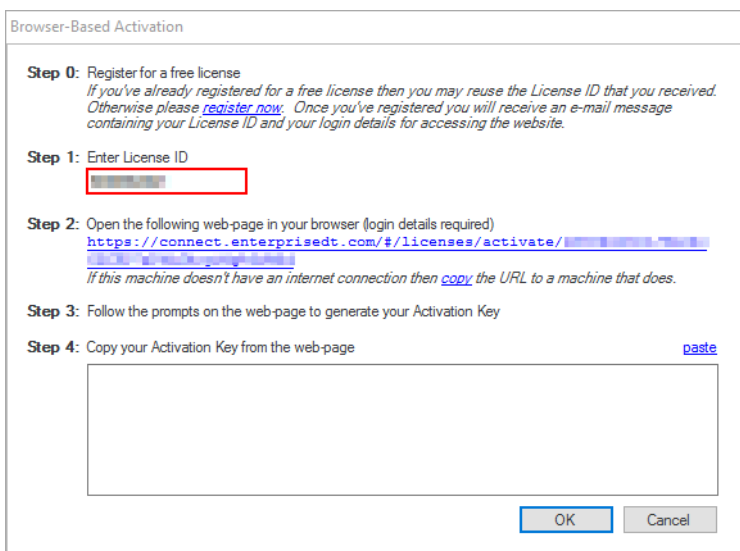
**Step 2:** Open the following web-page in your browser (login details required)  
<https://connect.enterprisedt.com/#/licenses/activate/W6c3c0YfP37zP4hf4syd4g%3d%3d>  
*If this machine doesn't have an internet connection then [copy](#) the URL to a machine that does.*

**Step 3:** Follow the prompts on the web-page to generate your Activation Key

**Step 4:** Copy your Activation Key from the web-page [paste](#)

OK Cancel

6. Enter the License ID in step 1.

The same "Browser-Based Activation" dialog box as above, but the "Enter License ID" field is now filled with a license ID and highlighted with a red box.

Browser-Based Activation

**Step 0:** Register for a free license  
*If you've already registered for a free license then you may reuse the License ID that you received. Otherwise please [register now](#). Once you've registered you will receive an e-mail message containing your License ID and your login details for accessing the website.*

**Step 1:** Enter License ID

**Step 2:** Open the following web-page in your browser (login details required)  
<https://connect.enterprisedt.com/#/licenses/activate/XXXXXXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX>  
*If this machine doesn't have an internet connection then [copy](#) the URL to a machine that does.*

**Step 3:** Follow the prompts on the web-page to generate your Activation Key

**Step 4:** Copy your Activation Key from the web-page [paste](#)

OK Cancel

7. Click the link marked 'copy' in step 2, to copy it to the clipboard.

**Browser-Based Activation**

**Step 0: Register for a free license**  
*If you've already registered for a free license then you may reuse the License ID that you received. Otherwise please [register now](#). Once you've registered you will receive an e-mail message containing your License ID and your login details for accessing the website.*

**Step 1: Enter License ID**

**Step 2: Open the following web-page in your browser (login details required)**  
<https://connect.enterprisedt.com/#/licenses/activate/>  
*If this machine doesn't have an internet connection then **copy** the URL to a machine that does.*

**Step 3: Follow the prompts on the web-page to generate your Activation Key**

**Step 4: Copy your Activation Key from the web-page** [paste](#)

OK Cancel

8. Send the link to a computer that does have Internet access and open it.
9. On the web-page that opens, enter your user-name and password. Then click 'Sign in'.

edtConnect - Sign In

EnterpriseDT Connect  
 SECURE FILE TRANSFER & SHARING

Home > Licenses > Activate

**Sign into your account**

**User-name**

**Password**

**Sign in** →

[Trouble signing in?](#)

10. Copy the Activation Key shown on the web-page, to the clipboard.

edtConnect - Licenses: Ac...

**Activate**

**Purchase ref.**

**Machine key**

**Activation key**

**Instructions**  
 Copy the activation key to the clipboard and paste it to the textbox in Step 4 of the CompleteFTP Manager's Activation form.

[← Manage licenses](#)

11. Send the Activation Key to the computer that's running CompleteFTP.
12. Paste the Activation Key into the field in step 4.



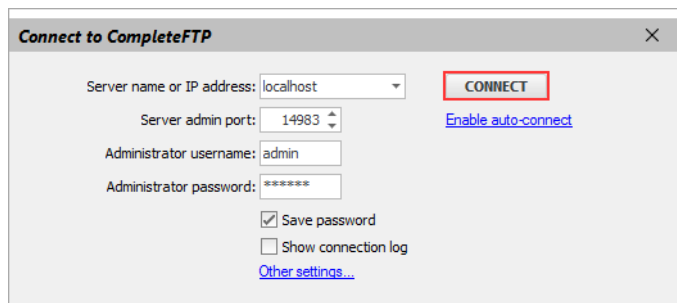


# Activate the paid editions

An installation of CompleteFTP Standard, Professional or Enterprise MFT can be activated via the Licensing panel in the CompleteFTP Manager but first of all, you need to purchase a license. Licenses may be ordered from the [EnterpriseDT web-site](#).

Once your order has been processed, a message will be sent to the technical contact e-mail address which you provided. This message will contain the Purchase Reference of your order, your user-name and a link which you can use to set your password. You will need your password to activate your installation. Once you have your password, please complete the following steps:

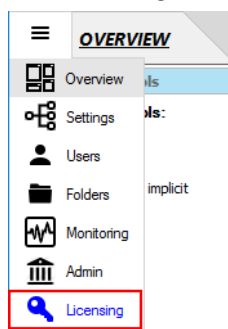
1. Open CompleteFTP Manager and connect to your server.



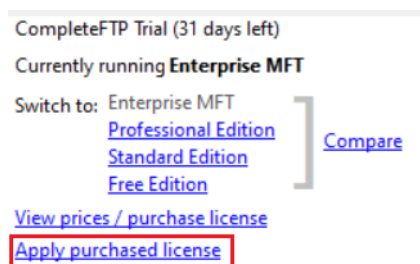
The 'Connect to CompleteFTP' dialog box contains the following fields and options:

- Server name or IP address: localhost
- Server admin port: 14983
- Administrator username: admin
- Administrator password: \*\*\*\*\*
- ☒ Save password
- ☐ Show connection log
- [Other settings...](#)
- [Enable auto-connect](#)
- CONNECT** button

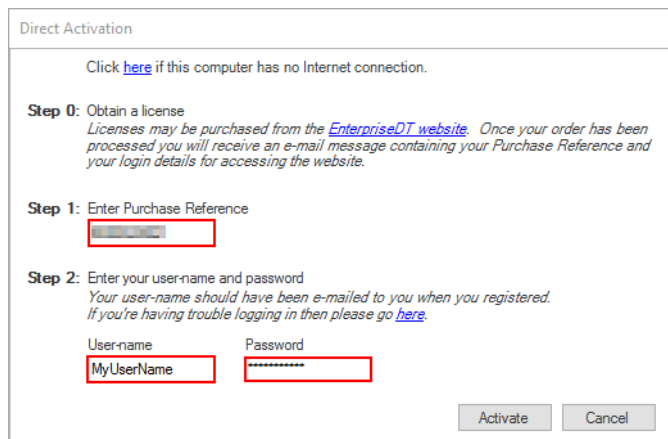
2. Select the 'Licensing' menu item from the side-bar menu.



3. Click the 'Apply purchased license' link.



4. If the computer on which CompleteFTP Manager is running has an Internet connection, then follow the steps below.
  - a. Enter the Purchase Reference, user-name and password in the appropriate text fields.



The 'Direct Activation' dialog box contains the following steps and fields:

- Click [here](#) if this computer has no Internet connection.
- Step 0: Obtain a license**  
Licenses may be purchased from the [EnterpriseDT website](#). Once your order has been processed you will receive an e-mail message containing your Purchase Reference and your login details for accessing the website.
- Step 1: Enter Purchase Reference**  
[Redacted text box]
- Step 2: Enter your user-name and password**  
Your user-name should have been e-mailed to you when you registered. If you're having trouble logging in then please go [here](#).
- User-name: MyUserName
- Password: \*\*\*\*\*
- [Activate](#) button
- [Cancel](#) button

- b. Then Click the 'Activate' button.

Direct Activation

Click [here](#) if this computer has no Internet connection.

**Step 0:** Obtain a license  
*Licenses may be purchased from the [EnterpriseDT website](#). Once your order has been processed you will receive an e-mail message containing your Purchase Reference and your login details for accessing the website.*

**Step 1:** Enter Purchase Reference

**Step 2:** Enter your user-name and password  
*Your user-name should have been e-mailed to you when you registered. If you're having trouble logging in then please go [here](#).*

User-name:  Password:

[Activate](#) [Cancel](#)

- c. Your server will have been activated within a few seconds. Click the 'OK' button when the notice appears.

Activation Applied

The server activation has been applied.

[OK](#)

5. If the computer on which CompleteFTP Manager is running has no Internet connection then follow the steps below.
- a. Click the 'here' link at the top of the 'Direct Activation' form.

Click [here](#) if this computer has no Internet connection.

- b. Enter the Purchase Reference in step 1.

Browser-Based Activation

**Step 0:** Purchase a license  
*Licenses may be purchased from the [EnterpriseDT website](#). Once your order has been processed you will receive an e-mail message containing your Purchase Reference and your login details for accessing the website.*

**Step 1:** Enter Purchase Reference

**Step 2:** Open the following web-page in your browser (login details required)  
<https://connect.enterprisedt.com/#/licenses/activate/>  
*If this machine doesn't have an internet connection then [copy](#) the URL to a machine that does.*

**Step 3:** Follow the prompts on the web-page to generate your Activation Key

**Step 4:** Copy your Activation Key from the web-page [paste](#)

[OK](#) [Cancel](#)

- c. Click the link marked 'copy' in step 2 to copy it to the clipboard.

Browser-Based Activation

**Step 0:** Purchase a license  
*Licenses may be purchased from the [EnterpriseDT website](#). Once your order has been processed you will receive an e-mail message containing your Purchase Reference and your login details for accessing the website.*

**Step 1:** Enter Purchase Reference

**Step 2:** Open the following web-page in your browser (login details required)  
<https://connect.enterprisedt.com/#/licenses/activate/>  
*If this machine doesn't have an internet connection then [copy](#) the URL to a machine that does.*

**Step 3:** Follow the prompts on the web-page to generate your Activation Key

**Step 4:** Copy your Activation Key from the web-page [paste](#)

[OK](#) [Cancel](#)

- d. Send the link to a computer that does have Internet access and open it.
- e. On the web-page that opens, enter your user-name and password. Then click 'Sign in'.

edtConnect - Sign In

EnterpriseDT  
SECURE FILE TRANSFER & SHARING

Connect

Home > Licenses > Activate

Sign into your account

User-name  
MyUserName

Password  
\*\*\*\*\*

Sign in →

[Trouble signing in?](#)

f. Copy the Activation Key shown on the web-page to the clipboard.

edtConnect - Licenses: Ac...

Activate

Purchase ref.  
[Redacted]

Machine key  
[Redacted]

Activation key  
[Redacted]

Instructions  
Copy the activation key to the clipboard and paste it to the textbox in Step 4 of the CompleteFTP Manager's Activation form.

← Manage licenses

g. Send the Activation Key to the computer that's running CompleteFTP.

h. Paste the Activation Key into the field in step 4.

Browser-Based Activation

**Step 0:** Register for a free license  
*If you've already registered for a free license then you may reuse the License ID that you received. Otherwise please [register now](#). Once you've registered you will receive an e-mail message containing your License ID and your login details for accessing the website.*

**Step 1:** Enter License ID  
[Redacted]

**Step 2:** Open the following web-page in your browser (login details required)  
<https://connect.enterprisedt.com/#/licenses/activate/>  
*If this machine doesn't have an internet connection then [copy](#) the URL to a machine that does.*

**Step 3:** Follow the prompts on the web-page to generate your Activation Key

**Step 4:** Copy your Activation Key from the web-page [paste](#)

[Redacted]

OK Cancel

i. Click 'OK' to complete activation.

Browser-Based Activation

**Step 0:** Register for a free license  
*If you've already registered for a free license then you may reuse the License ID that you received. Otherwise please [register now](#). Once you've registered you will receive an e-mail message containing your License ID and your login details for accessing the website.*

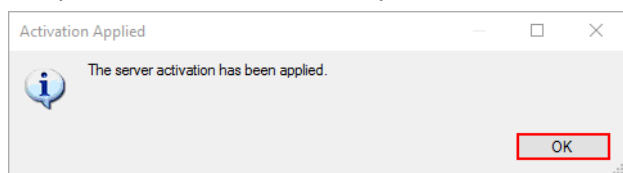
**Step 1:** Enter License ID

**Step 2:** Open the following web-page in your browser (login details required)  
<https://connect.enterprisedt.com/#/licenses/activate/60000000-0000-0000-0000-000000000000>  
*If this machine doesn't have an internet connection then [copy](#) the URL to a machine that does.*

**Step 3:** Follow the prompts on the web-page to generate your Activation Key

**Step 4:** Copy your Activation Key from the web-page [paste](#)

j. Now, your server has been activated successfully. Click the 'OK' button when the notice appears.



## Activating with a Universal Activation Key (UAK)

If you've purchased a [corporate license](#) (for any paid edition) then you're eligible for a UAK. A UAK will activate any machine without any need to connect to the EnterpriseDT website. Please e-mail [sales@enterprisedt.com](mailto:sales@enterprisedt.com) to obtain your UAK.

Follow these steps to activate a server using a UAK:

1. Run the CompleteFTP manager and connect to the CompleteFTP server instance that you wish to activate (usually localhost).

Connect to CompleteFTP

Server name or IP address:

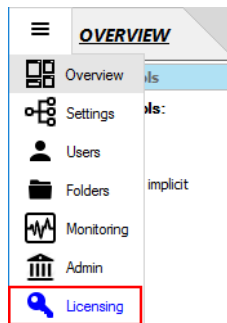
Server admin port:  [Enable auto-connect](#)

Administrator username:

Administrator password:

☒ Save password  
☐ Show connection log  
[Other settings...](#)

2. Select the 'Licensing' menu item from the side-bar menu.



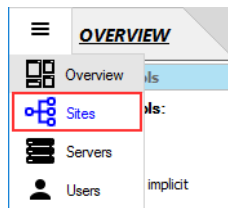
3. Click the 'Apply purchased license' link.



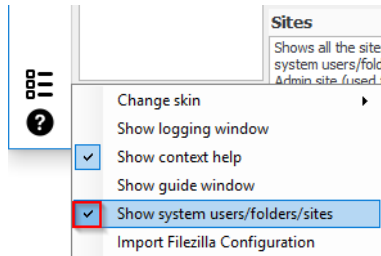
# Set up a cluster

Before setting up a cluster, you need to ensure that:

1. The firewall rules are configured to allow the primary server to communicate with the secondary servers.
2. Each secondary server's IP filtering rules permit the primary server to contact them. You can add a new rule by the steps below:
  - a. Connect to each secondary server.
  - b. Select the 'Sites' menu item from the side-bar menu.



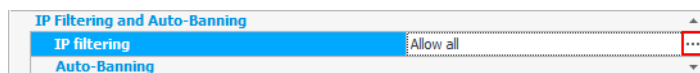
- c. Check 'Show system users/folders/sites' in the Options menu at the bottom left of the window.



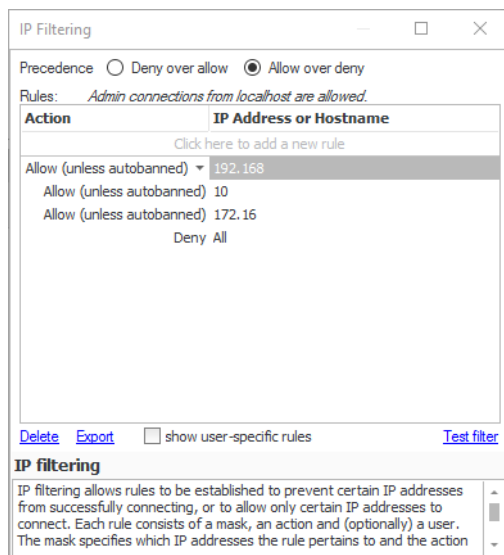
- d. Select the 'Admin' site in the site list.



- e. Select the 'IP filtering' property in the 'IP filtering and Auto-banning' category. Then click the ellipsis (...) button.

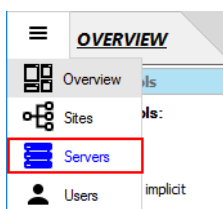


- f. In the 'IP Filtering' window, add a new IP Filter rule that allows the primary server to connect to the secondary if the default rules do not satisfy your needs.



Now, connect to the primary server and start setting up a cluster.

1. Select the 'Servers' menu item from the side-bar menu.



The 'Servers' panel will show up.

Name	Sites	Last Contact
Server 1 (primary)	Default Site	4:27:58 PM

Name	Server 1
Sites	Default Site
Synchronization	
Listening IP Addresses	
Machine Info	
Diagnostic	
License Information	

[Apply new license](#) [Extend update/support period](#)

**Server Properties**

Shows the sites that are enabled on the server (except the admin site, which is enabled on all servers), and permits sites to be enabled or disabled.

[Add server](#) [Remove](#)

- Click the 'Add server' link at the bottom of the panel to add a server as a secondary.

[Add server](#) [Remove](#)

- Enter the secondary server's details.

Add Server

**Add a server**

**Step 1: Connection**

Server name or IP address (use commas to separate multiple)

10.0.1.134

Server admin port

14983

Administrator username

admin

Administrator password

\*\*\*\*\*

[Other settings...](#)

[Previous](#) [Next](#) [Cancel](#)

- Then click 'Next'.

Add Server

**Add a server**

**Step 1: Connection**

Server name or IP address (use commas to separate multiple)

10.0.1.134

Server admin port

14983

Administrator username

admin

Administrator password

\*\*\*\*\*

[Other settings...](#)

[Previous](#) [Next](#) [Cancel](#)

- Click 'Add' to start adding the secondary server to the cluster.



Add Server

### Add a server

#### Step 2: Details

Details:

Ready to add:  
10.0.1.134

Click 'Add' to start:

Previous
Add
Cancel

6. The new server will be added to the cluster as a secondary server. Click the 'Close' button to close the 'Add server' dialog box.

Add Server

### Add a server

#### Step 2: Details

Details:

Adding 10.0.1.134:  
Connecting to remote server  
Querying remote server  
Adding remote server's ID to local configuration  
Adding existing sites to new server  
Applying configuration to remote server. Remote server will restart.  
Saving changes to local configuration  
Reconnecting to remote server - attempt 1 of 10 (0.0s)  
Remote server restarted successfully.  
Success

Successfully added:  
10.0.1.134

Previous
Add
Close

Now, the newly added secondary server is displayed in the list.

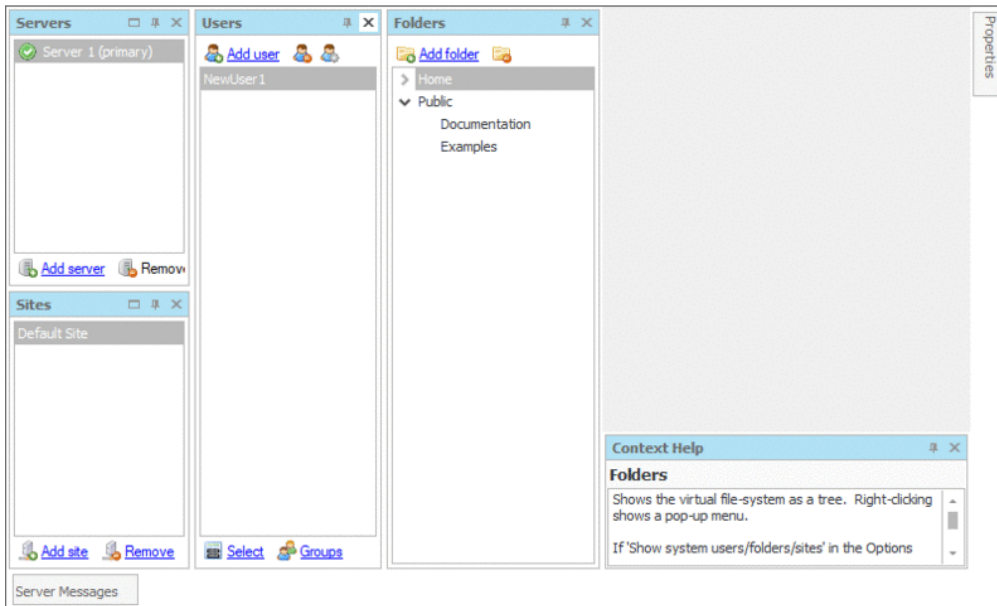
Name	Sites	Last Contact
<div>✓</div> Server 1 (primary)	Default Site	9:37:28 PM
<div>✓</div> Server 2	Default Site	9:36:55 PM

Name	Server 1
Sites	Default Site
Synchronization	▼
Listening IP Addresses	▼
Machine Info	▼
Dagnostic	▼
License Information	▼

# Overview

The Overview panel allows the administrator to combine a series of views within the one panel.

Views may be shown/hidden via the Options menu in the bottom-left of the window. Views can be resized, moved, docked to any edge, combined into tabs and moved into separate "floating" windows. A view may be resized by dragging its edge and moved by dragging its title-bar. While a view is being dragged several icons are shown overlaid on the window; dropping the view onto one of these will dock it to an edge or place it in a tab. The video below illustrates some ways in which panels may be configured:




The properties view shows the properties of the currently selected site, user or folder. If the property view is already visible then it will automatically change as different items are selected. If it is not visible then it may be displayed by right-click on a user or a folder and selected *Properties*.

# Settings/Sites

A CompleteFTP *site* consists of a related group of protocols and port numbers - for example the default ports for FTP on port 21 and SFTP on port 22 make up the "Default site" (below). Sites have their own settings, including public keys and certificates. Every installation of CompleteFTP supports at least two *sites* - the 'Admin site' and the 'Default site'.

## Default site

As described above, the "Default site" consists of a group of protocols on their standard port numbers - FTP/FTPS on port 21 and SSH/SCP/SFTP on port 22 (and optionally HTTP on port 80 and HTTPS on port 443). These are the port numbers and protocols most often used, and generally will not need to be modified, although the port numbers can be changed if required. In the *Standard Edition* and *Professional Edition*, the default site is configured via the *Settings* panel.

**SETTINGS**  Step-by-step guides

Site label	CompleteFTP
<b>FTP / FTPS</b>	
FTP enabled	<input checked="" type="checkbox"/>
FTPS explicit enabled (standard FTPS)	<input checked="" type="checkbox"/>
FTPS implicit enabled (legacy FTPS)	<input checked="" type="checkbox"/>
<b>Anonymous Logins</b>	
<b>Advanced FTP/FTPS Settings</b>	
Port for FTP / FTPS explicit	21
Port to use for FTPS implicit	990
Listening IP addresses	All available
Folder listing format	Unix
Character set	Default - Unicode (UTF-8)
<b>Passive (PASV) Transfer Settings</b>	
<b>Security Settings</b>	
<b>SFTP / SCP</b>	
SFTP enabled	<input checked="" type="checkbox"/>
SCP enabled	<input checked="" type="checkbox"/>
SSH terminal access enabled	<input type="checkbox"/>
SSH forwarding enabled	<input type="checkbox"/>
<b>Advanced SFTP/SCP Settings</b>	
Port for SFTP/SCP/SSH	22
Listening IP addresses	All available
Authentication methods	All
RSA Key	RSA key: 2048 bits
DSA Key	DSA key: 1024 bits
ECDSA Keys	nistp256
<b>Algorithms</b>	
<b>HTTP / HTTPS</b>	
HTTP enabled	<input checked="" type="checkbox"/>
HTTPS enabled	<input checked="" type="checkbox"/>
<b>Advanced HTTP/HTTPS Settings</b>	
Port for HTTP	80
<b>Site Settings</b>	
Protocol-related settings.	

In *Enterprise MFT*, the default site is configured via the *Sites* panel.

**SITES**

Step-by-step guides

Site	Site name	Default Site
Default Site	Site label	CompleteFTP
	Site enabled	<input checked="" type="checkbox"/>
	<b>FTP / FTPS</b>	
	FTP enabled	<input checked="" type="checkbox"/>
	FTPS explicit enabled (standard FTPS)	<input checked="" type="checkbox"/>
	FTPS implicit enabled (legacy FTPS)	<input checked="" type="checkbox"/>
	<b>Anonymous Logins</b>	
	<b>Advanced FTP/FTPS Settings</b>	
	Port for FTP / FTPS explicit	21
	Port to use for FTPS implicit	990
	Listening IP addresses	All available
	Folder listing format	Unix
	Character set	Default - Unicode (UTF-8)
	<b>Passive (PASV) Transfer Settings</b>	
	<b>Security Settings</b>	
	<b>SFTP / SCP</b>	
	SFTP enabled	<input checked="" type="checkbox"/>
	SCP enabled	<input checked="" type="checkbox"/>
	SSH terminal access enabled	<input type="checkbox"/>
	SSH forwarding enabled	<input type="checkbox"/>
	<b>Advanced SFTP/SCP Settings</b>	
	Port for SFTP/SCP/SSH	22
	Listening IP addresses	All available
	Authentication methods	All
	RSA Key	RSA key: 2048 bits
	DSA Key	DSA key: 1024 bits
	ECDSA Keys	nistp256
	<b>Algorithms</b>	
	<b>HTTP / HTTPS</b>	
	HTTP enabled	<input checked="" type="checkbox"/>
	HTTPS enabled	<input checked="" type="checkbox"/>

**Sites**

Shows all the sites in the system, except the administration site. If 'Show system users/folders/sites' in the Options menu at the bottom left of the window is checked then the Admin site (used for server administration) is also shown.

Add site
 Remove

## Admin site

The "Admin site" is what the CompleteFTP manager connects to, to administer the server instance. By default the server listens on port 14983 for SFTP connections from the manager. The Admin site cannot be deleted (the server could not be administered). Settings should be changed with care - if the port numbers are altered administrators must be informed of the changes. In the *Standard Edition* and *Professional Edition*, the Admin site is configured via the [Admin](#) panel.

**ADMIN**

Step-by-step guides

Administrator username  

[Change administrator password](#)

Administrator email address

Administrator full name

[Hide advanced administrator connection settings](#)

Advanced Administrator Connection Settings:

Site label	CompleteFTP
<b>FTP / FTPS</b>	
FTP enabled	<input type="checkbox"/>
FTPS explicit enabled (standard FTPS)	<input type="checkbox"/>
FTPS implicit enabled (legacy FTPS)	<input type="checkbox"/>
<b>Anonymous Logins</b>	
<b>Advanced FTP/FTPS Settings</b>	
Port for FTP / FTPS explicit	14982
Port to use for FTPS implicit	990
Listening IP addresses	All available
Folder listing format	Unix
Character set	Default - Unicode (UTF-8)
<b>Passive (PASV) Transfer Settings</b>	
<b>Security Settings</b>	
<b>SFTP / SCP</b>	
SFTP enabled	<input checked="" type="checkbox"/>
SCP enabled	<input checked="" type="checkbox"/>
SSH terminal access enabled	<input type="checkbox"/>
SSH forwarding enabled	<input type="checkbox"/>
<b>Advanced SFTP/SCP Settings</b>	
<b>HTTP / HTTPS</b>	
HTTP enabled	<input type="checkbox"/>
HTTPS enabled	<input checked="" type="checkbox"/>

**Advanced administrator settings**

These settings apply only to the administrative server, which runs on a different port to that of the standard server accessed by all other user accounts.

In *Enterprise MFT*, the Admin site is displayed in *Sites* by checking 'Show system users/folders/sites' in the Options menu at the bottom left of the window.

Site	Site label	CompleteFTP
<b>Admin</b>		
Default Site		

FTP / FTPS	
FTP enabled	<input type="checkbox"/>
FTPS explicit enabled (standard FTPS)	<input type="checkbox"/>
FTPS implicit enabled (legacy FTPS)	<input type="checkbox"/>
<b>Anonymous Logins</b>	
<b>Advanced FTP/FTPS Settings</b>	
Port for FTP / FTPS explicit	14982
Port to use for FTPS implicit	990
Listening IP addresses	All available
Folder listing format	Unix
Character set	Default - Unicode (UTF-8)
<b>Passive (PASV) Transfer Settings</b>	
<b>Security Settings</b>	
<b>SFTP / SCP</b>	
SFTP enabled	<input checked="" type="checkbox"/>
SCP enabled	<input checked="" type="checkbox"/>
SSH terminal access enabled	<input type="checkbox"/>
SSH forwarding enabled	<input type="checkbox"/>
<b>Advanced SFTP/SCP Settings</b>	
<b>HTTP / HTTPS</b>	
HTTP enabled	<input type="checkbox"/>
HTTPS enabled	<input checked="" type="checkbox"/>
<b>Advanced HTTP/HTTPS Settings</b>	
<b>Server-side Javascript (JSS)</b>	
<b>Public Access</b>	
<b>File System Settings</b>	
<b>IP Filtering and Auto-Banning</b>	
<b>Limits and Timeouts</b>	
<b>Messages</b>	

Settings	
Change skin	
Show logging window	
<input checked="" type="checkbox"/> Show context help	
Show guide window	
<input checked="" type="checkbox"/> Show system users/folders/sites	
Import Filezilla Configuration	

Server settings	
The HTTP and HTTPS protocols.	
Server online	

## Additional sites

CompleteFTP Enterprise MFT supports the creation and administration of **multiple sites**.

# FTP/FTPS

FTP (File Transfer Protocol) is a well established Internet protocol designed to transfer files (and information about files) across networks using TCP (Transmission Control Protocol). FTP is defined in the Request For Comments 959 document ( RFC 959), which can be obtained from the [Internet Engineering Task Force](#).

FTPS is FTP over SSL. There are two incompatible variants of FTPS: explicit and implicit. FTPS explicit is the most recent and most widely used variant. FTPS implicit is older and is generally being phased out. FTP and FTPS explicit can coexist on the same port (usually 21), in fact FTPS explicit connections start out as FTP connections before being switched to SSL. FTPS implicit requires its own dedicated port (usually 990).

FTP / FTPS	
FTP enabled	<input checked="" type="checkbox"/>
FTPS explicit enabled (standard FTPS)	<input checked="" type="checkbox"/>
FTPS implicit enabled (legacy FTPS)	<input checked="" type="checkbox"/>
Anonymous Logins	
Advanced FTP/FTPS Settings	
Port for FTP / FTPS explicit	21
Port to use for FTPS implicit	990
Listening IP addresses	All available IPv4
Folder listing format	Unix
Character set	Default - Unicode (UTF-8)
Passive (PASV) Transfer Settings	
External IP address (IPv4 only)	
Use external IP address for	<input checked="" type="radio"/> FTP/FTPS <input type="radio"/> FTPS only
IP white-list (in addition to client IP)	Only client IP allowed
Port Range	
Minimum port number	
Maximum port number	
Security Settings	
Clients must supply certificates	Don't require valid certificate
Server certificate (also used in HTTPS)	CN=localhost, O=EDT, S=So...
Available cipher suites	DHE_RSA_AES_256_SHA256, E...
Minimum SSL Version	SSL 3.0

The various FTP/FTPS settings that are available to configure *CompleteFTP* are described below. Note that settings must be saved via the *Apply Changes* button, otherwise they will be lost.

Setting	Description
FTP enabled	If checked, the FTP protocol is enabled. If not checked, the FTP protocol will not be supported by the server (although FTPS may be enabled, below).
FTPS explicit enabled (standard FTPS)	If checked, the FTPS protocol (explicit mode) is enabled. If not checked, FTPS explicit mode will not be supported by the server. If "FTP enabled" is not checked and FTPS (either implicit or explicit or both) is, then users are forced to use FTPS (i.e. they must connect securely).
FTPS implicit enabled (legacy FTPS)	If checked, the FTPS protocol (implicit mode) is enabled. This is a legacy form of FTPS that is still widely used. If not checked, FTPS implicit mode will not be supported by the server. If "FTP enabled" is not checked and FTPS (either implicit or explicit or both) is, then users are forced to use FTPS (i.e. they must connect securely).
Anonymous logins enabled	If checked, anonymous access is enabled. This means connections can be made by logging in as an anonymous user rather than as a named user with a password.
Max no. of anonymous connections	The maximum number of simultaneous anonymous connections permitted to the server.
Advanced FTP/FTPS Settings	
Port for FTP/FTPS explicit	Port that is used for FTP and FTPS in explicit mode. The standard port number for FTP and explicit FTPS is port 21.
Port for FTPS implicit	Port that is used for FTPS in implicit mode. The standard port number for implicit FTPS is port 990.
Listening IP addresses	Lists the IP addresses to listen on for this protocol. All network interfaces can be selected, or individual interfaces selected.
Folder listing format	Format of folder listings being sent back to clients. In the FTP protocol folder listings are sent back to the client as plain text. This setting controls the format of this plain text. This is particularly significant for non-console client applications, which usually parse the directory listings automatically.

Character set	Character set that is used for encoding file-names.
<b>Passive (PASV) Transfer Settings</b>	
External IP address	In passive (PASV) mode transfers, the server listens on a random port between the minimum and the maximum port numbers specified, waiting for a connection from the client to this port. The server sends the client a reply to the PASV command that provides the IP address to connect to and the port number. This setting allows a hard-coded IP address to be set. If not set, the IP address of the interface that the server is listening on is provided. This setting is required in cases where the IP address of the server is not reachable by clients. For example, the client may need to connect to another IP address that is accessible externally, and a NAT device may direct the connection to the server.
Use external IP address for	In passive mode transfers, some routers don't cope with a hard-coded external IP address as set in <i>External IP address</i> - they expect to see the local LAN IP address in the PASV reply, and they replace this local address with the external IP address themselves. If the local LAN IP address is not found, some routers will abruptly terminate the connection. The solution in these situations is to only supply the hard-coded external IP address for encrypted (i.e. FTPS) connections, which the router can't inspect (and hence can't know what IP address is in the PASV reply). To do this, select the "FTPS only" setting rather than "FTP/FTPS" (which is the default).
IP white-list	By default, CompleteFTP accepts data-channel connections only from the IP address that the client initially connected from. In some network environments, however, the data-channel connection may come from a different IP address, even though it's actually from the same machine (which is always required). If this is to be allowed then those IP addresses may be white-listed here. This allows CompleteFTP to operate behind active/active firewalls requiring use of SNAT, including the Azure firewall.
Minimum port number	This is the minimum port number used in passive (PASV) mode. In PASV mode transfers, the server listens on a random port between the minimum and the maximum port numbers specified, waiting for a connection from the client to this port.
Maximum port number	This is the maximum port number used in passive (PASV) mode. In PASV mode transfers, the server listens on a random port between the minimum and the maximum port numbers specified, waiting for a connection from the client to this port.
<b>Security Settings</b>	
Clients must supply certificates	This option controls whether or not certificates should be requested from FTPS clients. If they are required then the client must send a certificate that may be validated against the Windows certificate store. This means that its issuer's certificate must be in the Windows 'Trusted root certificate authorities' store.
Server certificate	Displays the dialog for controlling the server certificate. A server certificate must be set if FTPS or HTTPS is enabled. The dialog displays the current certificate's properties, and enables a new self-signed certificate to be generated. Also, a certificate signing request (CSR) can be created to send to a certificate authority (CA) to obtain a CA certificate.  This property is shared with the HTTP/HTTPS settings.
Available cipher suites	This setting controls the cipher suites that can be used. Generally, the default is appropriate unless there are special requirements for a particular algorithm.
Minimum SSL version	This setting controls the minimum version of SSL/TLS supported on this site (for both FTPS and HTTPS). The default is TLS 1.0, which means that SSL 3.0 is not supported by default. This ensures that the server is not vulnerable to the security flaws of SSL 3.0.



# SFTP/SCP/SSH

SFTP is an abbreviation for "SSH File Transfer Protocol", and is exactly that - a protocol for transferring files over an SSH connection. SFTP is not the standard FTP protocol running over SSH. Although SFTP has similar capabilities and even similar commands to standard FTP, these similarities are purely superficial. The protocol is completely different and incompatible with FTP and its secure extension, FTPS. SCP is also a file transfer protocol that runs over SSH connections. It is a precursor to SFTP, and allows the copying of files and directories over SSH.

The various SFTP/SCP/SSH settings that are available to configure *CompleteFTP* are described below. Note that settings must be saved via the *Apply Changes* button, otherwise they will be lost.

Note that SFTP is not supported in the *Free Edition*, and that SCP and SSH terminals are not supported in the *Standard Edition*. The Professional Edition and Enterprise MFT support all features described in this section.

SFTP / SCP	
SFTP enabled	<input checked="" type="checkbox"/>
SCP enabled	<input checked="" type="checkbox"/>
SSH terminal access enabled	<input type="checkbox"/>
SSH forwarding enabled	<input type="checkbox"/>
Advanced SFTP/SCP Settings	
Port for SFTP/SCP/SSH	22
Listening IP addresses	All available
Authentication methods	All
RSA Key	RSA key set
DSA Key	DSA key set
Ed25519 Key	Ed25519 key set
ECDSA Keys	ECDSA key(s) set
Algorithms	
Available Key Exchange Methods	All
Available Ciphers	All
Compression Algorithms	All
MAC Algorithms	All
Public Key Algorithms	All

Setting	Description
SFTP enabled	If checked, the SFTP protocol is enabled. If not checked, the SFTP protocol will not be supported by the server.
SCP enabled (not available in <i>Standard Edition</i> )	If checked, the SCP protocol is enabled. If not checked, the SCP protocol will not be supported by the server.
SSH terminal access enabled (not available in <i>Standard Edition</i> )	If checked, users may (if permitted) log into the server using an SSH terminal. If not checked, SSH terminal access will not be supported by the server.
Advanced SFTP/SCP/SSH Settings	
Port for SFTP/SCP/SSH	Port that is used for SFTP, SCP and SSH. The standard port number is port 22. All of these protocols use the same port number.
Listening IP addresses	Lists the IP addresses to listen on for these protocols. All network interfaces can be selected, or individual interfaces selected.
Authentication methods	<p>Methods of user-authentication that the server accepts. Note that these methods can also be specified at the user level. When a user logs in, only the authentication methods specified at both levels are available. For example, if the site permits password only, and the user permits password and publickey, then only password will be available. Available methods are:</p> <ul style="list-style-type: none"><li>• <i>Password</i> - only a password is required. A password must be set for each user using the Users view.</li><li>• <i>PublicKey</i> - public key cryptography is used to authenticate the user. For every user the server must have a public key that matches the client's private key. Note that this method of authentication does not work for Windows users, since a password is required by Windows to log the user in. Thus <i>Password</i> or <i>PublicKeyAndPassword</i> should be enabled if Windows users are to be allowed to connect.</li><li>• <i>PublicKeyAndPassword</i> - both password and public-key authentication is used.</li></ul>
RSA key	The RSA server key - the private key that the server uses to identify itself. The server normally has a DSA and an RSA key set (and more recently, an Ed25519 key and ECDSA keys). This can be imported, exported or generated via the menu that appears when selecting the ellipsis (...) in the text box displaying the key's details. The private key can only be imported or exported if the CompleteFTP manager is running on the server that the CompleteFTP service is running on.
DSA key	The DSA server key - the private key that the server uses to identify itself. DSA keys are deprecated and should no longer be used. The server normally has a DSA key and an RSA key (and more recently, an Ed25519 key and ECDSA keys). This can be imported, exported or generated via the menu that appears when selecting the ellipsis (...) in the text box displaying the key's details. The private key can only be imported or exported if the CompleteFTP manager is running on the server that the CompleteFTP service is running on.

Ed25519 key	The Ed25519 server key - the private key that the server uses to identify itself. Ed25519 keys are much shorter than RSA keys for equivalent strength. DSA keys are deprecated and should no longer be used. The server normally has a DSA key and an RSA key (and more recently, an Ed25519 key and ECDSA keys). This can be imported, exported or generated via the menu that appears when selecting the ellipsis (...) in the text box displaying the key's details. The private key can only be imported or exported if the CompleteFTP manager is running on the server that the CompleteFTP service is running on.
ECDSA keys	The ECDSA server keys - the private keys that the server uses to identify itself. The Elliptic Curve Digital Signature Algorithm (ECDSA) is popular because keys are much shorter than RSA keys for equivalent strength. There are three ECDSA server keys supported, nistp256, nistp384 and nistp524. They can be imported, exported or generated via the dialog box that appears when selecting the ellipsis (...) in the text box displaying the key's details. The private keys can only be imported or exported if the CompleteFTP manager is running on the server that the CompleteFTP service is running on.
<b>Algorithms</b>	
Available key exchange methods	Controls what key exchange methods are supported by the server. It is rare that the defaults should be changed. It is possible that an administrator may want to disable weaker key exchange methods.
Available ciphers	Controls what ciphers are available to SSH for encrypting data. The default setting of All should generally be used unless some ciphers are required to be disabled.
Available ciphers	Controls what ciphers are available to SSH for encrypting data. The default setting of All should generally be used unless some ciphers are required to be disabled.
Compression algorithms	Permits the compression algorithms used in SSH to be selected. For example, if zlib compression is to be forced, then 'None' would need to be unselected.
MAC algorithms	Permits the MAC algorithms used in SSH to be selected. The default setting of All should generally be used unless some MACs are required to be disabled.
Public key algorithms	Controls what public key algorithms are supported by the server. For example, RSA public keys can be forced by disabling DSA.

# HTTP/HTTPS settings

HTTP/HTTPS is supported in Professional and Enterprise MFT.

The various HTTP/HTTPS settings that are available to configure *CompleteFTP* sites are described below. Note that settings changes must be saved via the *Apply Changes* button, otherwise they will be lost.

HTTP / HTTPS ▲

HTTP enabled	<input checked="" type="checkbox"/>
HTTPS enabled	<input checked="" type="checkbox"/>
Advanced HTTP/HTTPS Settings ▲	
Two-factor auth enabled	<input type="checkbox"/>
Port for HTTP	80
Port for HTTPS	443
Listening IP addresses	All available
Home web-app	FileManager2
Web features	FileSharing (/Share),FileServer ...
Listing template	<!DOCTYPE HTML PUBLIC "-//...
Error template	<!DOCTYPE HTML PUBLIC "-//...
Custom headers	5 fields
Custom MIME types	None
HTTP authentication (legacy)	<input type="checkbox"/>
Security Settings (also used in FTPS) ▲	
Server certificate	C=US, S=State, L=Locality, ST...
Available cipher suites	SECURE_CIPHERS
Minimum SSL Version	TLS 1.0
Server-side Javascript (JSS) ▲	
JSS enabled	<input checked="" type="checkbox"/>
Public Access ▲	
Public HTTP access enabled	<input checked="" type="checkbox"/>

Please note that HTTP and HTTPS should not be enabled until any other HTTP/HTTPS services on the server machine are shutdown, e.g. IIS and Apache. Other products such as Skype also use the HTTP and HTTPS default ports of 80 and 443. If in doubt, use *netstat* to check what ports are in use. Note that these products can be run simultaneously, but must be configured so their port numbers do not clash. Both Apache and CompleteFTP can be set up with HTTP/S port numbers that differ from the default ports.

There are additional HTTP settings in *Limits & Timeouts*.

For HTTPS security settings, see the [FTP/FTPS Security Settings](#) (as they are shared).

Setting	Description
HTTP Enabled	If checked, the HTTP protocol is enabled. If not checked, HTTP will not be supported by the server. If enabling, ensure that other HTTP servers are not running on the server machine.
HTTPS Enabled	If checked, the HTTPS protocol is enabled. If not checked, HTTPS will not be supported by the server. If enabling, ensure that other HTTPS servers are not running on the server machine.
Public HTTP access enabled	If checked, unauthenticated users will be able to access any file in the public folder. Unauthenticated users are represented by the user called 'anonymous' (which can be found in Users panel and by selecting "Show system users/folders/sites" in the main form's Options menu). By default the home-folder of this user is /Public, so this is the folder that will be publicly accessible if this option is enabled.
JSS enabled	Enable JSS ( <a href="#">Javascript Server-Side</a> ) for this site.
Port for HTTP	Port that is used for HTTP. The standard port number for HTTP is port 80.
Port for HTTPS	Port that is used for HTTPS. The standard port number for HTTP is port 443.
Listening IP addresses	Lists the IP addresses to listen on for this protocol. All network interfaces can be selected, or individual interfaces selected.
Home web-app	Site-default web-app that users will be redirect to after logging in. This is usually the web-based file-manager. This setting may be overridden by the home web-app setting for individual users.

Web features	<p>Displays the configured web features. Clicking on the ellipsis (...) will bring up a dialogue box listing each web feature, and showing the configured URL path used by clients to access the application, and whether or not it is enabled. The URL path can be modified here, and the enabled or disabled status. Features include:</p> <ul style="list-style-type: none"> <li>• <b>AccountManagement:</b> Basic account management feature for end-users.</li> <li>• <b>AdminAPI:</b> REST API for remote configuration.</li> <li>• <b>CustomCommand:</b> Execute custom command scripts</li> <li>• <b>FileManager1:</b> Deprecated. Kept for compatibility.</li> <li>• <b>FileManager2:</b> Responsive and recommended.</li> <li>• <b>FileManager3:</b> More secure. Soon to be the default.</li> <li>• <b>FileServer:</b> Serves files over HTTP</li> <li>• <b>FileSharing:</b> Controls HTTP access to shared files</li> <li>• <b>Include:</b> Required for other web features.</li> <li>• <b>Login and Logout:</b> Handles HTTP authentication</li> <li>• <b>SamlSingleSignOn:</b> Required for SAML login</li> <li>• <b>WebAdmin:</b> Basic configuration via browser</li> </ul>
Listing template	<p>Controls the format in which directory listings are displayed to the user. Directories may be listed by users by entering a URL specifying the path of a directory. The listing is rendered using HTML containing macros, such as %FileName%. Clicking on the ellipsis (...) will bring up a dialog permitting the user to enter their own template. See <a href="#">How to format HTML listings</a> for more information.</p>
Error template	<p>Controls the format in which HTTP errors are displayed to the user. The default template uses Javascript to redirect the user to their home folder for 403 errors. Clicking on the ellipsis (...) will bring up a dialog permitting the user to enter their own template.</p>
Custom headers	<p>A custom header is a HTTP header field that will be include with every HTTP response. By default, the following headers are included:</p> <ul style="list-style-type: none"> <li>• Strict-Transport-Security:max-age=31536000; includeSubdomains</li> <li>• X-Content-Type-Options:nosniff</li> <li>• Content-Security-Policy:frame-ancestors 'self'</li> <li>• X-Frame-Options:sameorigin</li> <li>• X-XSS-Protection:1; mode=block</li> </ul> <p>CompleteFTP will automatically generate headers, such as Content-Length, Content-Type, Date, Keep-Alive as required. Content-type is calculated using MIME types (see below).</p>
Custom MIME types	<p>A MIME type, or Internet media type, is a two part identifier for file formats. In its most basic form, it defines the type and the subtype for a file extension. For example, HTML files are defined as "text/html". The webserver sends the MIME type to the browser as part of the file being downloaded, and the browser uses the MIME type to decide how best to display the file. An HTML file, with a MIME type of "text/html", will be displayed as a web page. Most common MIME types are included by default in CompleteFTP. Custom MIME types can be added, either as additional file types or to override default MIME types.</p>
HTTP authentication (legacy)	<p>If this option is enabled then HTTP logins to CompleteFTP will be handled by the browser (usually via HTTP basic authentication) rather than by the CompleteFTP login page. This is not generally recommended as it can cause problems with logouts. SAML Single Sign-On will not work if this option is enabled.</p>
Allowed host-names	<p>Specifies the host-names that are allowed in requests for password reset email messages. If left blank then any host-name is allowed. The purpose is prevent host name header injection attacks by rejecting requests containing invalid host headers. It must be a comma-separated list of allowed host-names. Each host-name may optionally begin with '*' to indicate a wildcard subdomain match. Host-names must consist of alphanumeric characters and hyphens in subdomains, and must end with a valid top-level domain (e.g. .com, .net). Spaces are not permitted. Example: <code>example.com, *.example.com</code> allows example.com and all subdomains.</p>
Server certificate	<p>Displays the dialog for controlling the server certificate. A server certificate must be set if FTPS or HTTPS is enabled. The dialog displays the current certificate's properties, and enables a new self-signed certificate to be generated. Also, a certificate signing request (CSR) can be created to send to a certificate authority (CA) to obtain a CA certificate. This property is shared with the FTP/FTPS settings.</p>
Available cipher suites	<p>This setting controls the cipher suites that can be used. Generally, the default is appropriate unless there are special requirements for a particular algorithm</p>
Minimum SSL version	<p>This setting controls the minimum version of SSL/TLS supported on this site (for both FTPS and HTTPS). The default is TLS 1.0, which means that SSL 3.0 is not supported by default. This ensures that the server is not vulnerable to the security flaws of SSL 3.0.</p>

# Two Factor Authentication

CompleteFTP Enterprise MFT supports **Two Factor Authentication** (2FA) through the use of the **Time-based One-Time Password (TOTP)** algorithm, which is supported by most authenticator apps (e.g. **Google Authenticator**, **Microsoft Authenticator** and **FreeOTP**). It can be enabled by checking a single checkbox. Once enabled, web users will be required to install the app and activate the site before they can access CompleteFTP via HTTP/HTTPS. 2FA is not currently supported in protocols other than HTTP and HTTPS, but however, compatible with all authenticators available in CompleteFTP, such as SAML, Active Directory, external database and custom authenticators.

## How does Two Factor Authentication work?


2FA invokes the security principle of 'Something you know and something have'. The 'something you know' is your password and the 'something you have' is a device that generates **One-Time Passwords** (OTPs). With CompleteFTP's 2FA, the device is a mobile phone (Android or iPhone) that has a TOTP-compatible authenticator app installed, and the OTPs are 6-digit PINs that change every 30 seconds. The trick is that only the TOTP server and each specific instance of the app know the value of the PIN at any given time, so if the correct PIN entered then we can trust that the user has the phone on which the app is installed. The PIN synchronization between the server and the phone is done by scanning a QR code that generated by Google and scanned by the app. Each QR code is unique to the specific logged-in user on the site.

## How users experience Two Factor Authentication

Once 2FA has been enabled by the admin (see **below**), users will have to go through an additional step after entering their user-name and password before they can enter the site.

## Activating Two Factor Authentication

The first time users log in after 2FA has been enabled, or after their activation status has been **reset by an administrator**, they'll be presented with a page showing (1) instructions on how to install an authenticator app on their **iPhone** or **Android phone** and (2) a QR code for the site. Once they're up and running with the app they'll need to scan the QR code to register the site. Immediately after they've done this a 6-digit PIN will be displayed inside the app. They must enter this PIN on the web page and click the 'Activate' button. Once they've done this they be granted access to the site.


 **2 Factor Authentication Setup**

**Step 1: Install Google Authenticator on your phone**

- i. Google Authenticator on iPhone
- ii. Google Authenticator on Android

**Step 2: Add this website to Google Authenticator**

- i. Open the Google Authenticator app that you installed in step 1
- ii. Click the + symbol
- iii. Select 'Scan a QR code'
- iv. Point your phone's camera at the image below:



**Step 3: Activate your account**

- i. Read the PIN next to 'CompleteFTP (UserName)'
- ii. Enter the PIN below
- iii. Click 'Activate'

**PIN**

**ACTIVATE** **LOG OUT**

Note that first part of the site identifier, 'CompleteFTP (UserName)', can be modified (see **below**).


Users can manually activate new devices via the **Account Management web-app**. This is useful when they've got a new phone.

## Backup PINs

CompleteFTP's 2FA feature supports backup PINs, which are a set of PINs that don't change with time, but can only be used once each. These are useful for situations where users don't have access to their authenticator app. Backup PINs are generated on demand by the user in the **Account Management web-app** or by the system administrator in the **2FA status** view in CompleteFTP Manager. User can copy the PINs to a safe place, such as a password manager, or obtain them directly from the system administrator.

## Validating with Two Factor Authentication

Activation is only required when a user hasn't already **activated 2FA**. After that they'll only be required to validate by entering the PIN shown in the app.

 **Two Factor Authentication**

Enter your PIN

VALIDATE PIN

LOG OUT

Instructions:

1. Open the Google Authenticator app on your phone.

2. Look for the entry labelled 'CompleteFTP (Username)'

3. Enter the PIN.

4. Press 'Validate PIN'.

Enabling Two Factor Authentication in CompleteFTP

2FA can be enabled in the settings for any site by checking the checkbox labeled 'Two-Factor Authentication enabled'.

HTTP / HTTPS	
HTTP enabled	<input type="checkbox"/>
HTTPS enabled	<input checked="" type="checkbox"/>
Advanced HTTP/HTTPS Settings	
Two-factor auth enabled	<input checked="" type="checkbox"/>
Port for HTTP	80
Port for HTTPS	443

Additionally, the site setting can be overridden for any user by checking or unchecking the checkbox labelled 'Two-Factor auth enabled'.

Authentication	
Two-factor auth enabled	<input checked="" type="checkbox"/>
Methods (SSH)	All

Users often have multiple sites activated in the authenticator app, so it's important that each site is clearly identified. By default, sites will show up as 'CompleteFTP (username)' in the app but the first part of this should be customised for the site. It can be changed by setting the Label setting for the site. It should be set to something that identifies the specific site that the user is logging into, such as the name of the organisation, business unit or website.

Site name	Default Site
Site label	CompleteFTP
Site enabled	<input checked="" type="checkbox"/>

Two Factor Authentication Status Management

The backup PINs and activation status of individual users can be managed in the Two Factor Authentication status view, which is accessible from the Users panel in CompleteFTP Manager.

USERS

Add user

Remove

General user settings

Enabled	Username	Type	Home Folder	Groups
<input checked="" type="checkbox"/>	UserName	Non-Windows	/Home/UserName	users

Select

Groups

Two-factor auth status

The authentication status view shows the backup PINs and activation status of every user who has used 2FA.

Two-Factor Authentication Status

Login user-name	Log-in-as user	Backup PINs	Has activated?
UserName	-	849069,0587...	<input checked="" type="checkbox"/>

849069

058728

665634

350550

535682

992045

032694

New x 1

New x 10

Delete

Copy all

Force reactivation

Delete row(s)

Select all

Users can be forced to reactivate by clearing the 'Has activated?' checkboxes one at a time or in bulk by selecting a set of rows and pressing the 'Force reactivation' button. Deleting row has the same effect, but also clears out backup PINs. Backup PINs can be managed by clicking on a user's 'Backup PINs' field, which pops out an editor for the PINs. PINs can generated, deleted and copied to the clipboard. Copying to the clipboard enables system administrators to easily send them to users via instant messaging, though it's important that this be done securely. Since the PINs are short they can also simply be read out to the user over the phone, if required.

# Account Management

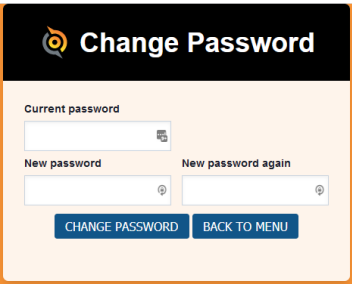
CompleteFTP users can access some basic account management features via the Account Management web-app. Users can do the following:

- See their username
- Change their password (if allowed)
- See their home folder (and navigate to the file manager)
- Set up a new device for **Two-Factor Authentication** (2FA) if it's enabled
- View and regenerate backup codes for TFA if it's enabled



## Changing Password

If the **password policy** of the server allows passwords to be changed then a link labelled 'change password'. When pressed the user is taken to a page where they can change their password. They must enter their existing password as well as their desired new password.

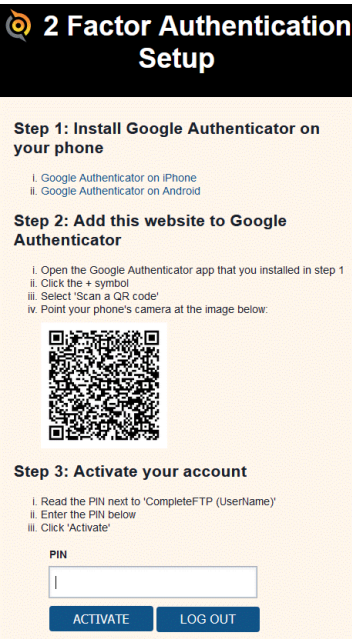


## Two-Factor Authentication (2FA)

To reduce the workload for administrators, users are able to register new devices and access backup codes via the Account Management web-app. This is only possible if 2FA is enabled for the site and the user, otherwise the 2FA section isn't shown on the account management home page.

### Setting up a new device for 2FA

This is done using the same page that users use to **activate their first device**.



### 2FA backup codes

As described [here](#), backup PINs are one-time passwords that users can use in place of the PINs generated on the device. Ten backup PINs are generated the first time the user visits the backup PIN page. The user should keep these in a secure place. PINs will be removed from the list as they're used. New PINs can be generated by clicking the 'Get new PINs' button.



## 2 Factor Authentication

### Backup PINs

#### Backup PINs

Keep the PINs below in a safe place and use them when your Authenticator app isn't available. Each can be used once only.

452270	792551
766431	765412
843731	774013
593344	364122
898233	303942

GET NEW PINs

BACK TO MENU



# File Sharing Settings

File sharing is supported in the Professional Edition and Enterprise MFT.

The file sharing settings that are available to configure *CompleteFTP* sites are described below. Note that settings must be saved via the *Apply Changes* button, otherwise they will be lost. There is also a per-user setting to enable or disable sharing.

File-Sharing ▲	
Enabled	<input checked="" type="checkbox"/>

Setting	Description
Enabled	If checked, file-sharing is enabled. If not checked, file-sharing will not be supported by the server. Note that users are enabled for sharing by default.

# Filesystem settings

The file system settings that are available to configure *CompleteFTP* sites are described below.

File System Settings	
Read-only	<input type="checkbox"/>
Show hidden files/folders	<input type="checkbox"/>
Allow backslash separators	<input type="checkbox"/>
Filename blocking filters	[Not set]
Allow overwrite on rename	<input type="checkbox"/>
ZIP-File Navigation	
Enabled	<input checked="" type="checkbox"/>
ZIP-folder extension	.folder
Encryption At Rest (EAR)	
Encrypt stored files	<input type="checkbox"/>

Setting	Description
Read-only	When checked only read access to the file-system will be permitted; all write, create, rename and delete operations will be disallowed.
Show hidden files/folders	When checked virtual folders and Windows files/folders whose hidden flag will be hidden from users. Note that users will <b>not</b> be prevented from accessing hidden files if they have the file-system permissions to do so.
Allow backslash separators	Unix-style forward slash separators are used in paths by default. Use this setting to enable support for Windows-style backslash separators (while still permitting forward slashes).
Filename blocking filters	<p>Sometimes it is a requirement to block the upload of certain file types, such as executable files. This setting allows you to specify the filters used to control what files can be uploaded. Note that the filters are also applied to renaming of files - otherwise a user could upload a file with an acceptable name and then rename it to a banned name. Administrators can choose to block files that match the filters, or to only permit files that match the filters.</p> <p>By default, wildcard characters, * and ?, are used in filter expressions, e.g. *.exe will block all files whose extension is .exe. Alternatively, regular expressions may be used by prefixing the filter with regex;, e.g. regex:A[0-9]+\..dat will block files with names like A1.dat, A024.dat and A34353.dat.</p>
Allow overwrite on rename	When enabled CompleteFTP will overwrite any existing files during rename and move operations. When disabled such operations will be blocked.
ZIP-file navigation enabled	When ZIP-file navigation is enabled ZIP files are presented to clients as though they are folders. This means that users can change into folders within the ZIP file and download specific files. This can save time and bandwidth as it allows users to download only what they need from a ZIP file instead of downloading the whole ZIP file. Note that this feature is available in Professional and Enterprise MFT only.
ZIP-folder extension	By default, both the ZIP file itself and the folder that corresponds to the ZIP file are shown in listings, the latter having the additional extension '.folder'. The additional extension can be changed by modifying the 'ZIP-folder extension' setting. If this setting is cleared completely then the ZIP file won't be displayed being replaced by the corresponding folder.
Encrypt stored files	If this setting is enabled and the user has the equivalent setting enabled, all files transferred to the server will be encrypted on the server (encryption at rest). The only way to decrypt files is by transferring them from the server, or via an administrator command. More details <a href="#">here</a> . Note that this feature is only available in Enterprise MFT.

## General user settings


The general user settings that are available to configure *CompleteFTP* sites have been moved to the *Users* tab, and can be displayed by selecting the *General user settings* link. They are also described [here](#).

# IP Filtering and Auto-Banning

All editions of CompleteFTP implement auto-banning, though it can only be customized in Professional and Enterprise MFT. Additionally, Professional and Enterprise MFT support *IP filtering*.

## Auto-banning

Auto-Banning	
Number of failures allowed in period	10
Failure counting period (seconds)	300
Duration of ban (seconds)	3,600

*Auto-banning* is an important feature related to IP filtering that helps secure the server. It works by automatically banning IP addresses from connecting if they have failed to authenticate a certain number of times within a time period. Bans last for a specified duration, or until the CompleteFTP service is restarted. If you wish for the duration of ban to last until the service is restarted then select the 'Duration of ban' property and then click the  button at the right.

For example, using the default settings, if a rogue client is attacking from a given IP address, and fails to guess a password correctly 10 times within a 5 minute (=300 second) period, their IP address is banned from connecting for the next hour. After an hour has elapsed, the ban is automatically lifted.

The auto-banning settings are configurable, although it is generally advisable to use the defaults.

The currently banned IP addresses can be viewed in the *Monitoring* panel. Bans can be removed or made permanent from this panel.

Real-Time Logging | Log Files | Connections | Performance | **Auto-bans** | TCP Port Usage

[Refresh](#) ☒ Refresh automatically

Site	IP Address	Begin Time	End Time	No. of Blocks
Default Site	10.0.2.41	01/11/2017 4:51:14 PM	01/11/2017 5:51:14 PM	2
Default Site	10.0.2.32	01/11/2017 4:55:57 PM	01/11/2017 5:55:57 PM	2

[Remove selected ban\(s\)](#) [Remove all bans](#) [Permanently ban selected IP address\(es\)](#)

Note that auto-bans are not persistent - if the service is restarted, all bans are lost (unless they are made permanent via the *Monitoring* panel).

Auto-banning can be disabled for specified IP addresses using an allow-always IP filter-rule (see below). Clients connecting from IP addresses that fall under an IP filter-rule whose action is *allow-always* will be immune from auto-banning. This is especially helpful in situations where many clients are connecting from a single Internet IP address since it stops a single 'misbehaving' client from causing all other clients to lose their ability to connect.

## IP filtering

IP Filtering and Auto-Banning	
IP filtering	Allow all ...

CompleteFTP's IP filtering is very flexible and may be used to implement many different policies. The following step-by-step guides are available:

- Step-by-step guide: [Block connections from specific IP addresses \(blacklisting\)](#)
- Step-by-step guide: [Accept connections only from specific IP addresses \(whitelisting\)](#)
- Step-by-step guide: [Accept connections only from specific users at specific IP addresses](#)
- Step-by-step guide: [Restrict some users to connect only from specific IP addresses](#)

An IP filter consists of a collection of rules. Each rule consists of a *mask* and an *action*. The mask specifies which IP addresses the rule pertains to and the action specifies what should be done. Each rule also has a scope that is either the entire site or a specific user.

IP Filtering

Precedence ☒ Deny over allow ☐ Allow over deny

Rules:

Action	IP Address or Hostname
Click here to add a new rule	
Allow (unless autobanned)	All
Deny	
Allow (unless autobanned)	
Allow always	

[Delete](#) [Export](#) ☐ show user-specific rules [Test filter](#)

IP filtering

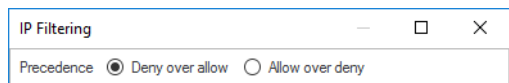
IP filtering allows rules to be established to prevent certain IP addresses from successfully connecting, or to allow only certain IP addresses to connect. Each rule consists of a mask, an action and (optionally) a user. The mask specifies which IP addresses the rule pertains to and the action

The mask may be (a) a specific IP address, such as 123.123.123.123, (b) the first one, two or three parts of an IP address, such as 192.168, which specifies all the IP addresses that begin with those parts, (c) a specific numeric range of IP addresses, such as 192.168.2.100-150, (d) an Internet host-name, or (e) a LAN network name.

There are three possible actions - *deny*, *allow* and *allow-always*. *Deny* will block access from clients whose IP addresses are included in the mask. *Allow* will permit access, unless the auto-banning system has detected too many failed login-attempts (see above). *Allow-always* is like allow except that auto-banning is disabled.

Rules are combined to filter incoming IP addresses. When a connection attempt is made, the rules are consulted to see if the remote IP address matches the rules, and the connection is dropped if it is determined that access should be denied.

The *precedence* of operations is important. By default, the precedence is *Deny over allow*. This means that if all IP addresses are allowed, and there is a deny rule for a specific address or range, then the deny rule takes precedence and any IP address matching the deny rule is denied. *Allow over deny* is the other option, which means if an IP address matches an allow rule, then it will be permitted access even if it is listed in a deny rule.

A screenshot of a dialog box titled "IP Filtering". The dialog box has a title bar with a minus sign, a maximize button, and a close button. Below the title bar, there is a section labeled "Precedence" with two radio buttons. The first radio button is selected and is labeled "Deny over allow". The second radio button is unselected and is labeled "Allow over deny".

Each rule may optionally be associated with a user. If it is associated with a user then it applies only to that user. If it's not associated with a user then it applies to the entire site. Typically *user-specific rules* are used in the following scenarios:

- Restrict specific users to connect only from specific IP addresses while the other users can still connect from any IP addresses.  
Please refer to Step-by-step guide: [Restrict some users to connect only from specific IP addresses](#) for detailed instructions.
- Reject all connections except those that are from specific users on specific IP addresses.  
Please refer to Step-by-step guide: [Accept connections only from specific users at specific IP addresses](#) for detailed instructions.

# Email Settings

The SMTP settings instruct CompleteFTP how to send emails. They are the same settings that are used when setting up email applications, such as Microsoft Outlook.

CompleteFTP sends emails from (1) email notification events, (2) forgot/reset password feature, (3) custom JSS scripts.

SMTP Settings

Host-name / IP address

smtp.fastmail.com

Port

587

☐ SSL/TLS

Username

hcaandersen@fastmail.com

Password

nugp28nqyd3mza5

☒ Show characters

Default 'from' email address

hcaandersen@fastmail.com

Test

SMTP Settings

The SMTP settings instruct CompleteFTP how to send emails. They are the same settings that are used when setting up email applications, such as Microsoft Outlook. CompleteFTP sends emails from (1) email notification events, (2) forgot/reset password on the login page, (3) custom JSS scripts. At a minimum a host-name must be defined. The most common ports are 25 if no encryption is being used and 465 or 587 if encryption is being used. If encryption is used the SSL/TLS checkbox must be checked.

Close

At a minimum a host-name must be defined. The most common ports are 25 if no encryption is being used and 465 or 587 if encryption is being used. If encryption is used the SSL/TLS checkbox must be checked.

Most SMTP servers require authentication, so the username and password should usually be set. Many SMTP servers related to online email services, such as Gmail, require special passwords to be generated for use with applications such as CompleteFTP.

The default 'from' email address is used whenever a 'from' address isn't specified.

Clicking the 'Test' button will cause a test email to be sent to a specified 'to' address, which may be entered when prompted. Note that CompleteFTP Manager will freeze while the test message is being sent, which may take up to 30 seconds if the connection times out.

# Limits and Timeouts

The various limits and timeouts that are available to configure *CompleteFTP* sites are described below. Note that settings must be saved via the *Apply Changes* button, otherwise they will be lost.

Limits and Timeouts ▲	
Timeout for HTTP Sessions (seconds)	1,800
Timeout for logging in (seconds)	60
Timeout for stalled transfers (seconds)	60
Timeout for idle sessions (seconds)	300
Timeout for PASV wait (seconds)	60
Max. allow login attempts	3
Max. non-HTTP connections to site	Unlimited
Max. HTTP connections to site	200
Max. connections per user	Unlimited

Setting	Description
Timeout for HTTP sessions	The maximum time in seconds before HTTP sessions expire. If the timeout is set to zero, sessions do not expire. The default is 1,800 seconds (30 minutes).
Timeout for logging in	The maximum time in seconds permitted for a login attempt. If a connection is made and login is not completed within this period, the connection is closed. If the timeout is set to zero, the timeout is infinite, i.e. the connection does not time out.
Timeout for stalled transfers	The maximum time in seconds that the underlying sockets wait for while performing read operations. If the timeout is set to zero, the timeout is infinite, i.e. read operations do not time out.
Timeout for idle sessions	The maximum time in seconds that a connection is permitted to be idle, i.e. if no commands are sent from the client. If the timeout is set to zero, the timeout is infinite, i.e. the connection does not time out.
Timeout for PASV wait	In passive mode, the server listens on a socket, waiting for the client to make a connection. The client can connect before actually sending the next command (LIST,NLST,RETR or STOR), and the server must wait for this next command before it can do anything with the client connection. If the timeout is set to zero, the timeout is infinite, i.e. the connection does not time out.
Max. allow login attempts	Sets the maximum number of login attempts that can be made for a user account.
Max. simultaneous connections	Sets the maximum number of simultaneous connections permitted. If this limit has been reached, any further connection attempts are denied until current connections are closed. This setting does not apply to HTTP connections.
Max. simultaneous HTTP connections	(Professional and Enterprise MFT only) Sets the maximum number of simultaneous HTTP connections permitted. If this limit has been reached, any further HTTP connection attempts are denied until current connections are closed.
Max. connections per user	Sets the maximum number of simultaneous connections permitted per user. If this limit has been reached for a user, any further connection attempts by that user are denied until current connections are closed.

# Messages

The message settings that are available to configure *CompleteFTP* sites are described below. Note that settings must be saved via the *Apply Changes* button, otherwise they will be lost.

Messages ▲

Welcome message	Complete FTP server
Hide server product details	<input type="checkbox"/>

Setting	Description
Welcome message	The text that is displayed (i.e. sent to the client) during a login attempt. In SFTP this is the banner message, and is often not displayed by SFTP clients.
Hide server product details	If this option is enabled, the product name and version of the server is not displayed in welcome messages. This may be required for PCI compliance. This option is not enabled by default.



# How to secure your server

FTP servers are always vulnerable to attack from unauthorised people, and a number of steps should be taken to minimize the risks of this occurring:

1. Keep up to date with the latest patches for your operating system. Microsoft regularly publishes security updates, so ensure that they are applied to your host.
2. Keep up to date with the [latest release](#) of CompleteFTP. Each release usually includes at least one security fix, and it is important to keep your installation current.
3. A good firewall is the first line of defence for security, and should be your first port of call in denying/permitting access to certain IP addresses.
4. If your server is not required to be accessible from the Internet, ensure that it is only reachable internally. If it is not accessible externally, the only attacks can be from within your organization, greatly reducing the risk.
5. Use the [IP filtering](#) capabilities of CompleteFTP to only permit the IP addresses you want (if this is possible).
6. Ensure that [auto-banning](#) is configured correctly (e.g. the defaults) to prevent dictionary attacks on passwords.
7. Regularly review log files for unwanted intrusions and take remedial action (such as banning IP addresses).
8. Disable protocols that aren't being used, e.g. if you are running an SFTP server only, disable FTP, FTPS, HTTP, HTTPS and SCP.
9. Don't use HTTP - use HTTPS. Usernames, passwords and other sensitive data should not be sent via HTTP.
10. Don't use FTP - use FTPS. Usernames, passwords and other sensitive data should not be sent via FTP.
11. Disable SSL 3.0, TLS 1.0, and TLS 1.1. SSL 3.0 is disabled by default. TLS 1.0 and 1.1 are currently enabled by default as some clients still use them. Disable them if possible.
12. Disable the automatic Windows users feature, so that only explicitly permitted users are permitted.
13. Use strong password policies - see [here](#). Also disable any anonymous FTP access unless it is required.
14. For SFTP disable password authentication, and only permit public key authentication. This means users must have valid private keys and have their public keys registered on the server. This is not always possible of course.
15. For SFTP disable SSH terminal access. This is disabled by default. SSH terminal access permits Windows users who have this feature enabled to execute almost any program or DOS command, and is potentially a significant security hole.
16. Disable weak algorithms. For SSH, disable weak ciphers such as TripleDES and Blowfish. Disable weak MACs such as HMACMD5 and HMACSHA1. For FTPS/TLS, disable any ciphers using 3DES.
17. Hide the server product details. This is found in the Settings->Messages section, and conceals the version and name of the product from clients. This is disabled by default. The default welcome message should also be changed, as it identifies the server.
18. Enforce hashed passwords for database users. If database users are enabled (not the default), make sure any stored passwords are hashed. Even better, use a 'salt' prepended to passwords before they are hashed. See [how to configure database users](#)
19. Enable encryption at rest. This means all files transferred to the server using accounts that have encryption at rest enabled will be encrypted on the server filesystem. This means that if an attacker succeeds in obtaining unauthorized access to the server machine, files are still secure. See [here](#) for more details.

## How to set a default Windows domain

If a default Windows domain is set, it means that Windows users on the default domain can login without specifying the domain (instead of using `MyDomain\MyUserName` they can use `MyUserName`). This can be very convenient for users.

You can set the default Windows domain in the *General User Settings* dialog box of the *Users* panel as shown below.

Add user Remove General user settings More columns

Enabled	Username	Email	Type	Home Folder	Groups
<input checked="" type="checkbox"/>	Omaha\t...		Windows	/Home/Omaha-test1	users

### General User Settings

For site: Default Site

☐ Allow anonymous FTP users

☒ Allow public HTTP access

☐ Home folder is root (for site)

**Default Windows domain**

[Password policies](#)

Close

**Default Windows domain**  
 If a default Windows domain is set, it means that Windows users on the default domain can login without specifying the domain (instead of using MyDomain\MyUserName they can use MyUserName). This can be very convenient for users. The default Windows domain tells CompleteFTP which domain to authenticate against if (1) a domain-name was not given and

Select Groups Two-factor auth status

### AUTHENTICATORS

For site: Default Site

Authenticator	Enabled	Log in as	
Database	<input type="checkbox"/>	defaultDatabase	<a href="#">Configure...</a>
Gateway	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
LDAP	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Single sign-on / SAML	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Windows/AD Group	<input checked="" type="checkbox"/>	defaultWindows	<a href="#">Configure...</a>

### User Properties (Omaha\test1):

Username	Omaha\test1
User type	Windows
Email	
Home folder	/Home/Omaha-test1
Home web-app	[Site default]
Enabled	<input checked="" type="checkbox"/>
Expires	[Never]

#### Authentication

Two-factor auth enabled	<input checked="" type="checkbox"/>
Methods (SSH)	All
Password	[Not set]
Public keys	No keys
Allowed IP addresses	All

#### File-System

Home folder is root	<input type="checkbox"/>
Encrypt stored files	<input checked="" type="checkbox"/>
Report decrypted size	<input type="checkbox"/>
Access control	Windows
Read-only	<input type="checkbox"/>

#### Allowed Protocols

FTP	<input checked="" type="checkbox"/>
-----	-------------------------------------

### General User Settings

General user settings have been moved to the Users panel.

The default Windows domain tells CompleteFTP which domain to authenticate against if (1) a domain-name was not given and (2) no exact match was found in CompleteFTP's list of users.

For Windows users that have been explicitly added to CompleteFTP, if the client tries to log in using a username without a domain-name then the server will first look for this exact username in the config (just like it does now). If a match is not found and a default domain has been set then it tries to prefix the default domain to the username and look up the user-list.

If the user is not found in the explicitly added list of users and automatic Windows users are enabled, the default domain is again used if the domain name was not specified while logging in. This will result in logging in as 'DefaultDomain\MyUserName'. One consequence of this is that if a default Windows domain has been set then an attempt to log in without specifying a domain name will never log into the local machine. CompleteFTP will always assume the default domain should be used. If the user does want to log into the local machine when a default domain has been set they have to log in using ".\MyUserName".

# How to use HTTP/HTTPS

CompleteFTP supports HTTP and HTTPS in the *Professional Edition* and in *Enterprise MFT*.

HTTP and HTTPS may be enabled by selecting the "HTTP enabled" and "HTTPS enabled" settings (see the HTTP/HTTPS group of settings).

Please note that HTTP and HTTPS should not be enabled until any other HTTP/HTTPS services on the server machine are shut down, e.g. IIS, Apache. Other products such as Skype also use the HTTP and HTTPS default ports of 80 and 443. If in doubt, use *netstat* to check what ports are in use. Note that these products can be run simultaneously, but must be configured so their port numbers do not clash. Both Apache and CompleteFTP can be set up with HTTP/S port numbers that differ from the default ports.

[How to use HTTP](#)

[How to use HTTPS](#)

[How to perform HTTP uploads](#)

## How to use the File Manager

CompleteFTP supports HTTP and HTTPS (Professional and Enterprise MFT) and is a capable web-server in its own right, it includes three file-manager web-apps. The latest, *FileManager3*, is the most secure and modern. It's recommended, but isn't yet the default as it hasn't yet proven itself in end-user environments. We look forward to making it the default soon. *FileManager2* is the default. It's been around for a few years and still gives an excellent user experience, but is showing its age and will soon be deprecated. *FileManager1* is available for backward compatibility, but will be discontinued in an upcoming release.

### Responsive file-manager (FileManager3):

The screenshot shows a file explorer interface. The left sidebar displays a directory tree with the following structure:







- ▼ /Home/javaftp
  - > New folder

The main pane shows the contents of the selected directory, /Home/javaftp. It contains a table with the following data:

<input type="checkbox"/>	Name	Date Modified	File Size
	New folder	2/26/2025, 3:40 PM	
	ftp-codes.java	3/10/2025, 1:50 PM	16 KB
	image02.png	3/10/2025, 1:50 PM	16 KB
	winscp.log	3/10/2025, 1:51 PM	1.5 MB

At the bottom left, it says "4 files/folders".

### Responsive file-manager (FileManager2):

/Home/javaftp			
<div>▼  /</div> <div>&gt;  New folder</div>	Name ▼	Size	Date
	 New folder		2/4/2019 4:10:45 PM
	 ftp-codes.java	16.4 kB	2/4/2019 4:10:27 PM
	 image02.png	16.3 kB	2/4/2019 4:10:39 PM
	 winscp.log	1.5 MB	2/4/2019 4:10:39 PM

**Original file-manager (FileManager1):**

Name	Modified	Size	Kind
New folder	Today 04:10 PM	0 b	Folder
ftp-codes.java	Today 04:10 PM	16 KB	Application
image02.png	Today 04:10 PM	16 KB	PNG Image
winscp.log	Today 04:10 PM	1.54 MB	Application

/Home/javaftp/ items: 4, size: 1.57 MB

## Selecting a default File-Manager

New installations of CompleteFTP set FileManager2 as the default file-manager. Upgrading CompleteFTP won't change the existing default file-manager, so this must be done manually. It can be done for sites or for individual users:

To select a default file-manager for a site:

1. Open to the Settings panel (called 'Sites' in Enterprise MFT)
2. [Enterprise MFT only] Select the Default Site
3. Open the 'Advanced HTTP/HTTPS Settings' section
4. Select the desired file-manager as the 'Home web-app'.
5. Click Apply changes

To select a default file-manager for a specific user

1. Open the Users panel
2. Select the desired user
3. Select the desired file-manager as the 'Home web-app'.
4. Click Apply changes

## Accessing the File-Manager

By default, the file-managers are available at the URLs, /FileManager/1, /FileManager/2, and /FileManager/3, but users are also redirected to their default file-manager when they access their home directories. If users have not authenticated, they will be prompted to do so by the standard browser authentication dialog. Once authenticated, the file-manager will be displayed.

## Disabling the File-Managers

All file-manager web-apps are enabled by default, but can be disabled. To disable a file-manager, go to the Settings/Sites tab. Go down to the HTTP/HTTPS section. Select the ellipsis (...) in the Web features setting and uncheck the file-manager you wish to disable. More details can be found [here](#).

## Customizing the File-Managers

FileManager1 is based on the *eFinder* file manager, and is written entirely in Javascript. Accordingly, it can be modified (at your own risk!). The code can be found at *C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\FileManager\1* or the equivalent (depending on your OS).

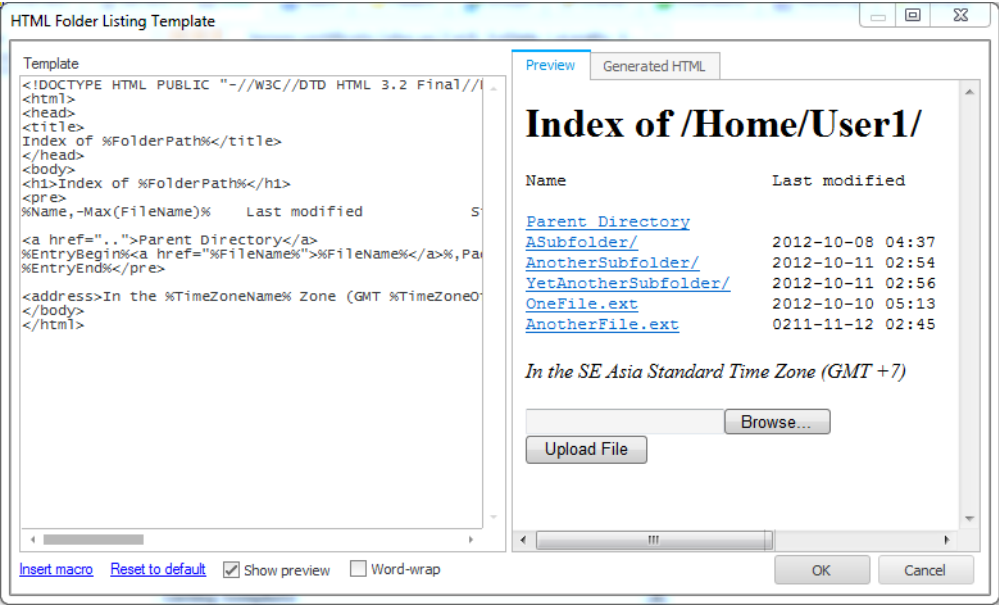
FileManager2 is a fork of the GitHub project, [angular-filemanager](#). Our fork is available at [github.com/EnterpriseDT/completeftp-filemanager](#). We greatly appreciate any pull requests that developers feel might be useful for other users.

Customizing FileManager3 requires a license for [DevExtreme](#). [Contact support](#) for more information.

# How to format HTML listings

When a HTTP client requests a folder URL that it has permission to list and the file-explorer is disabled, CompleteFTP will return a listing in HTML. The format of this listing is controlled by a template that may be modified. The template is defined by a single HTML page containing macros (see the 'Template Format' section below).

The template is modified using an editor that features real-time previews. Clicking on the ellipsis (...) in the Listing template property displays this editor, shown below.



The HTML to be used for the folder listing is entered on the left. On the right is shown the rendered HTML, and the generated HTML (i.e. with macros expanded). Macros may be entered manually or inserted by clicking the 'Insert macro' link at the bottom. The 'Reset to default' link may be clicked to set the template back to CompleteFTP's default format.

## Template Format

The template is split into three sections: the header, the entry-block and the footer. The header is everything before the %EntryBegin% macro; the entry-block is everything between %EntryBegin% and %EntryEnd%; and the footer is everything after %EntryEnd%. When a folder-listing is rendered the entry-block is repeated once for each entry in the folder. An entry can be a file or a sub-folder.

The table below defines the available macros as well as stating which section each macro may be used in:

Macro	Section	Description
%EntryBegin%	Start of entry-block	Marks the start of a block that defines the format of a single entry in the listing - file or folder
%EntryEnd%	End of entry-block	Marks the end of a block that defines the format of a single entry in the listing - file or folder
%FileCreated%	Entry-block	Date and time at which the file or folder was created
%FileGroup%	Entry-block	Group of the folder or file
%FileLength%	Entry-block	Length of the file in bytes, kB, MB or GB
%FileModified%	Entry-block	Date and time at which the file or folder was last modified
%FileName%	Entry-block	Name of the file or folder (not including the path)
%FileOwner%	Entry-block	Owner (user) of the folder or file
%FilePath%	Entry-block	Full virtual file-system path of file or folder
%FilePermissions%	Entry-block	UNIX-like permission string of the file or folder
%FileType%	Entry-block	Type of the directory-entry - either 'File' or 'Folder' (useful for CSS classes)
%FolderName%	Header or footer	Name of the folder being listed
%FolderPath%	Header or footer	Full virtual file-system path of the folder being listed
%TimeZoneName%	Header or footer	Name of the current time-zone
%TimeZoneOffset%	Header or footer	Offset (in hours) of the current time-zone compared to UTC

## Formatting:

Formatting instructions may be provided with each macro to refine how it is rendered. These instructions use a syntax similar to the **composite formatting** syntax defined by Microsoft. The general pattern is:

%macro-name[,alignment][:formatString]%

where **macro-name** is the name of the macro as shown in the table above or plain text (see below); **alignment** is an integer defining the preferred width of the field (positive=left-aligned, negative=right-aligned); and **formatString** provides instructions on how the text is formatted. These instructions depend on whether the field's type is a **date** or a **number**.

The functions, **Max(m)** and **Pad(m)**, may be placed in a format-string. **m** is the name of an entry-block macro, such as **FileName**. **Max(m)** returns the maximum width of the field, **m**. **Pad(m)** returns the difference between the maximum and minimum widths of the field, **m**.

If the text that is not a legal macro-name is provided then that text is rendered literally. If no macro-name is provided then spaces will be inserted according to the **alignment** format. For example, **,Max(FileName)** will insert a number of spaces equal to the longest file-name.

### Example:

The following example produces a listing similar to the default listing format of Apache:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
<head>
<title>Index of %FolderPath%</title>
</head>
<body>
<h1>Index of %FolderPath%</h1>
<pre>
%Name,-Max(fileName)% Last modified Size Description

<a href="..">Parent Directory</a>
%EntryBegin%<a href="%FileName%">%FileName%</a>%,Pad(fileName)% %FileModified,-25:yyyy-MM-dd hh:mm%FileLength,-10%
%EntryEnd%</pre>

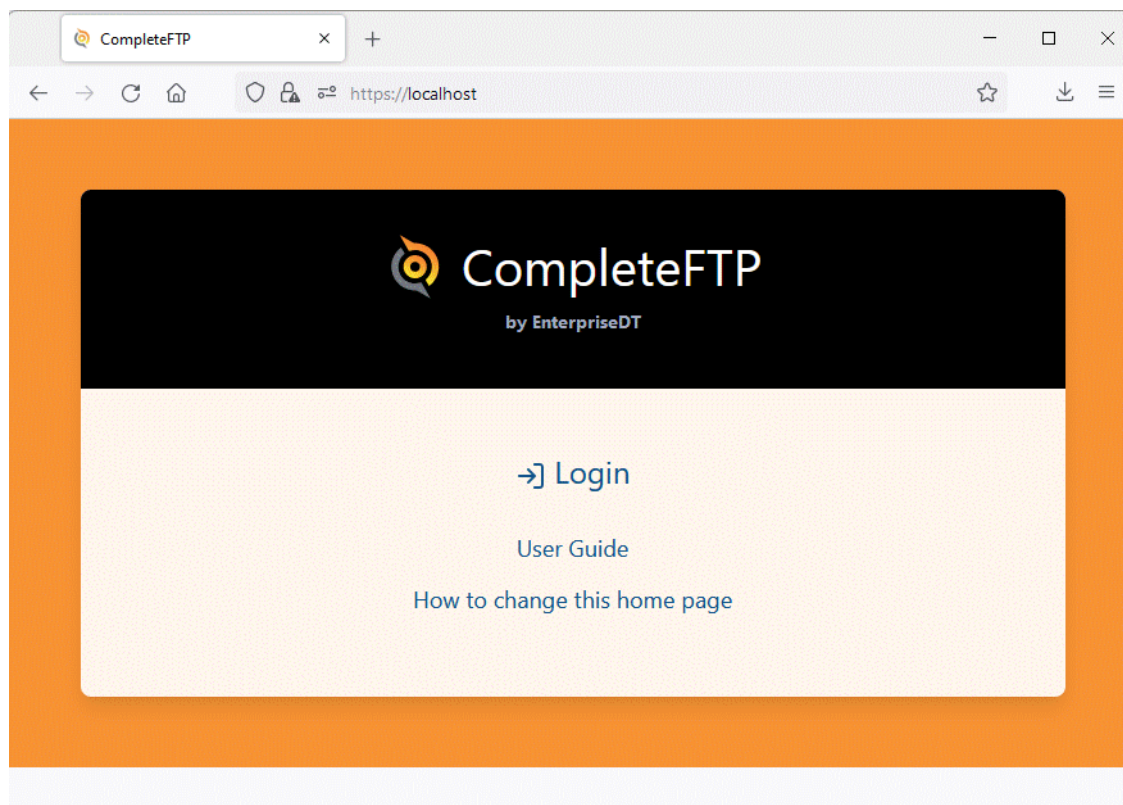
<address>In the %TimeZoneName% Zone (GMT %TimeZoneOffset:+#;-#;0%)</address><FORM METHOD=POST ENCTYPE="multipart/form-data"
ACTION="."><INPUT TYPE=FILE NAME="upfile"><INPUT TYPE=SUBMIT VALUE="Upload File"></FORM>
</body>
</html>
```

### Notes:

1. Since Name is not a legal macro-name, the text %Name,-Max(fileName)% will render the literal characters "Name" followed by the number of spaces required to yield the same total width as the longest file-name.
2. %,Pad(fileName)% yields a number of spaces equal to the difference between the longest and shortest file-names.
3. %FileModified,-25:yyyy-MM-dd hh:mm% renders a file's last-modified date left-aligned in the format "2012-15-11 14:02" and padded with spaces to give a total width of 25 characters.
4. %TimeZoneOffset:+#;-#;0% yields the timezone offset hours with a plus- or minus-sign in from of it.

# How to customize the welcome page

CompleteFTP's default home page (below) is displayed when a user first navigates to the server via a web browser.



The text on the home page can be modified or the whole page can be replaced entirely. Both are done by creating a file with a set of customization instructions at the following path:

```
C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Include\custom\customizations.jss
```

This file is a **JSS (Javascript Server-Side)** file. You may modify the text and logo of the existing home page by placing a set of variable assignments in this file, as illustrated below:

```
publicPageContent.homeURL = "https://mycompany.com";
publicPageContent.logo = "https://mycompany.com/logo.jpg";
publicPageContent.title = "My Company";
publicPageContent.subTitle = "";
publicPageContent.login = "Sign In";
publicPageContent.links = [
  { url: "https://mycompany.com/instructions.html", text: "Usage Instructions" },
  { url: "https://mycompany.com/privacy.html", text: "Privacy Policy" }
];
```

Alternatively, you can replace the page entirely by setting the variable, `publicPageTemplate`, to the path of an HTML file, as shown in the example below:

```
publicPageTemplate = "../Include/custom/homePage.html";
```

This example instructs CompleteFTP to replace the default home page with a file at

```
C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Include\custom\homePage.html
```

The HTML file must be created by you and should include a link to `/Login`. Alternatively it could contain Javascript code to redirect the user to the login page automatically, as shown below:



```
<html>
<body>
  <script type = "text/javascript">
    window.location = "/Login";
  </script>
</body>
</html>
```

Similarly, the login and logout pages may be replaced by setting the `loginPageTemplate` and `logoutPageTemplate` variables, respectively. Note that these two pages contain a number of critical functional elements (e.g. `<form>` and `<input>`), as well as macros (in double curly brackets) and Javascript code, so care must be taken when developing alternative templates. A good approach would be to copy the existing template and modify it incrementally, testing carefully after each modification. It's also important to note that future versions of CompleteFTP may rely on modifications to the default templates to support new features, so it will be important to check release notes for advice on such changes, and to confirm that any customized templates work reliably after CompleteFTP updates are installed.

For more information on customizing the home page, please refer to Step-by-step guide: [Customize the web home page](#)

# How to use CompleteFTP with firewalls

Issue: using FTPS, clients can connect but transfers and directory listings hang and eventually timeout.

Firewalls block connections to networks (and thus servers) on certain port numbers, and can thus prevent client applications connecting.

If problems are being experienced connecting to CompleteFTP, the firewall is a good place to start looking.

HTTP and HTTPS rarely have firewall problems, while SFTP has fewer problems with firewalls than FTPS. All commands and transfers go through a single port (the default is port 22). As long as the firewall is permitting connections to port 22, SFTP should work correctly.

The situation is more complicated with FTP/FTPS, mainly because directory listings and file transfers are performed using separate socket connections on a different (and potentially random) port number to the standard port 21. Of course firewalls are designed to prevent connections on random port numbers, so listings and transfers can get blocked. This is not usually an issue for FTP, as firewalls can read FTP commands and dynamically open data ports. But for FTPS, commands are encrypted and so firewalls cannot open ports dynamically. Instead, a firewall rule must be set up permitting inbound connections to the server (for passive mode transfers, also known as PASV).

In passive mode, the FTP/FTPS client initiates all transfers, and the server returns an IP address and port number to connect to. This port number is either assigned by the server's operating system (if the defaults are used), or is used from a passive port range configured in the server. To set up a passive port range with CompleteFTP, see [how to set a PASV port range](#). This port range should also be set in the firewall, permitting clients to connect in this range.

Often, the FTP server will be on a machine with only an internal IP address. As the IP address is returned to the client via the PASV command, this is problematic. The client requires an external IP address, and the firewall must be configured to reroute packets to this external IP address to the server machine. To set an external IP address with CompleteFTP, see [how to set an external IP address](#).

# How to set an external IP address (IPv4 only)

FTP uses a separate connection for each file-transfers and directory-listing. In passive mode file-transfers, the server listens on a random port between the minimum and the maximum port numbers specified, waiting for a connection from the client to this port. The server tells the client which port it's chosen. For IPv4, it also tells the client which IP address to connect to. For cases where the server is behind a NAT device, the IP address that the clients need to connect to is different from the IP address of the interface on the computer, which is why this setting is provided. It should be set to the external (usually Internet) IP address of the server. If it's not set then the LAN IP address of the interface that the server is listening on is provided. In fact, most modern clients ignore the IP address provided here, so for those clients the setting is not required.

In IPv6 the server doesn't tell the client which IP address to connect to as the client is required to assume that the IP address to connect the data connections to is the same as the main (control) connection.

<b>FTP / FTPS</b>	
FTP enabled	<input checked="" type="checkbox"/>
FTPS explicit enabled (standard FTPS)	<input checked="" type="checkbox"/>
FTPS implicit enabled (legacy FTPS)	<input checked="" type="checkbox"/>
<b>Anonymous Logins</b>	
<b>Advanced FTP/FTPS Settings</b>	
Port for FTP / FTPS explicit	21
Port to use for FTPS implicit	990
Listening IP addresses	All available
Folder listing format	Unix
Character set	Default - Unicode (UTF-8)
<b>Passive (PASV) Transfer Settings</b>	
External IP address (IPv4 only)	
Use external IP address for	<input checked="" type="radio"/> FTP/FTPS <input type="radio"/> FTPS only
IP white-list (in addition to client IP)	Only client IP allowed
<b>Port Range</b>	
Minimum port number	
Maximum port number	
<b>Security Settings</b>	

## Using the external IP address for FTPS only

There is one caveat to using a hard-coded external IP address - some routers don't like it. These routers expect to see the local LAN IP address in the PASV reply, and they replace this local address with the external IP address themselves. If the local LAN IP address is not found, some routers will abruptly terminate the connection.

The solution in these situations is to only supply the hard-coded external IP address for encrypted (i.e. FTPS) connections, which the router can't inspect (and hence can't know what IP address is in the PASV reply). To do this, use the *Use external IP address for* setting and select "FTPS only" rather than "FTP/FTPS" (which is the default).

# How to set a PASV port range

In FTP's **passive (PASV) mode**, transfers and directory listings are performed on a separate network connection to the control connection, which is typically on port 21. Instead, the server listens on a different port number which is in the server's passive port range. The PASV command sends this port number to the client, asking it to connect on this port to make the transfer or listing.

The passive port range is set in the server as described below. Typically, firewalls must also be configured on the server end to allow clients to connect to the server for all ports in the passive port range. An unconfigured firewall is usually the issue if clients can connect successfully but fail to transfer files or list directories. This is not applicable to SFTP, which uses a single port - only FTP and FTPS.

The minimum and maximum port numbers are set using the *Minimum port number* setting and the *Maximum port number* setting. These settings are part of the Passive (PASV) Transfer Settings group of settings. It is recommended to set a passive port range of 100, and the range should be between 1024 and 65535. For example, set the lower bound to 15000, and the upper bound to 15100.

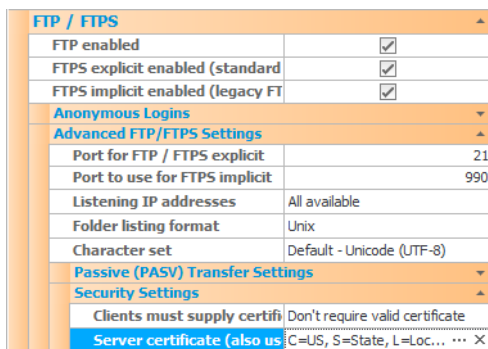
<b>FTP / FTPS</b>	
FTP enabled	<input checked="" type="checkbox"/>
FTPS explicit enabled (standard FTPS)	<input checked="" type="checkbox"/>
FTPS implicit enabled (legacy FTPS)	<input checked="" type="checkbox"/>
<b>Anonymous Logins</b>	
<b>Advanced FTP/FTPS Settings</b>	
Port for FTP / FTPS explicit	21
Port to use for FTPS implicit	990
Listening IP addresses	All available
Folder listing format	Unix
Character set	Default - Unicode (UTF-8)
<b>Passive (PASV) Transfer Settings</b>	
External IP address (IPv4 only)	
Use external IP address for	<input checked="" type="radio"/> FTP/FTPS <input type="radio"/> FTPS only
IP white-list (in addition to client IP)	Only client IP allowed
<b>Port Range</b>	
Minimum port number	
Maximum port number	
<b>Security Settings</b>	

If these values are not set explicitly, the defaults are used.

# How to set up the server certificate

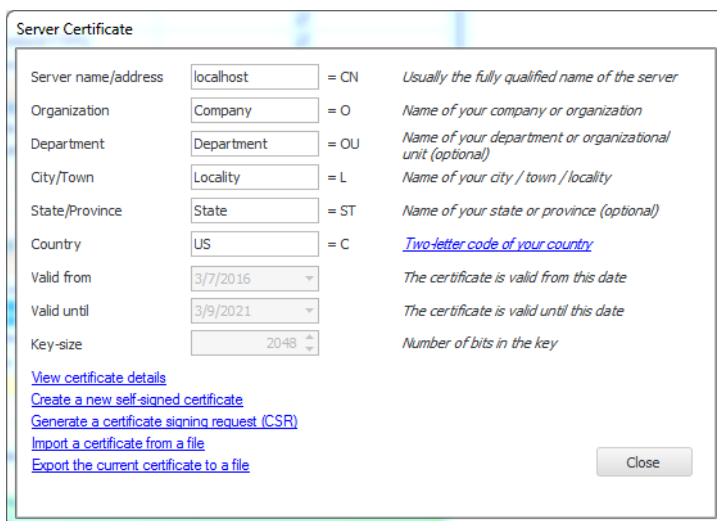
When using FTPS and HTTPS, the server **must** have an SSL certificate installed. The certificate is an integral part of how the secure connection is established.

By default, a server certificate is already installed. It can be viewed via the *Server certificate* setting in the FTP/FTPS group of settings. Selecting the "..." button (see below) will bring up the Server Certificate dialog box.



FTP / FTPS	
FTP enabled	<input checked="" type="checkbox"/>
FTPS explicit enabled (standard)	<input checked="" type="checkbox"/>
FTPS implicit enabled (legacy FT	<input checked="" type="checkbox"/>
Anonymous Logins	
Advanced FTP/FTPS Settings	
Port for FTP / FTPS explicit	21
Port to use for FTPS implicit	990
Listening IP addresses	All available
Folder listing format	Unix
Character set	Default - Unicode (UTF-8)
Passive (PASV) Transfer Settings	
Security Settings	
Clients must supply certificate	Don't require valid certificate
Server certificate (also used for HTTPS)	C=US, S=State, L=Locality, O=Company, OU=Department, CN=localhost

The default certificate has a Common Name (CN) of "localhost", and is a self-signed certificate. By convention, the CN is often set to the hostname of the server for use in server validation, and another self-signed certificate with the desired hostname as CN can be generated by selecting "Create a new self-signed certificate" (below). This will replace the existing certificate.



**Server Certificate**

Server name/address	localhost	= CN	Usually the fully qualified name of the server
Organization	Company	= O	Name of your company or organization
Department	Department	= OU	Name of your department or organizational unit (optional)
City/Town	Locality	= L	Name of your city / town / locality
State/Province	State	= ST	Name of your state or province (optional)
Country	US	= C	Two-letter code of your country
Valid from	3/7/2016		The certificate is valid from this date
Valid until	3/9/2021		The certificate is valid until this date
Key-size	2048		Number of bits in the key

[View certificate details](#)  
[Create a new self-signed certificate](#)  
[Generate a certificate signing request \(CSR\)](#)  
[Import a certificate from a file](#)  
[Export the current certificate to a file](#)

Close

On production machines, certificates issued by a **certificate authority** (CA) may be required. If the server will be accessed from HTTPS or FTPS clients external to the organization (e.g. across the Internet) then a CA certificate is essential. Many organizations already have CA certificates issued and centrally controlled. Otherwise, you will need to **purchase and install** a CA certificate.

# How to set up a CA SSL certificate

On production machines, the default self-signed SSL certificate will probably not be sufficient. Instead, a certificate issued by a **certificate authority** (CA) may be required. If the server will be accessed from HTTPS or FTPS clients external to the organization (e.g. across the Internet) then a CA certificate is essential. Many organizations already have CA certificates issued and centrally controlled. Otherwise, a CA certificate must be purchased and installed, as described below.

There are three steps in obtaining and installing a CA certificate:

1. **Generate a Certificate Signing Request (CSR)**
2. **Purchase the CA certificate using your CSR**
3. **Download and install your CA certificate**

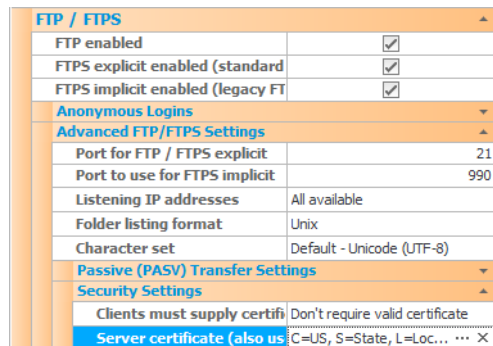
# How to generate a CSR

To purchase a CA SSL certificate from a CA, a Certificate Signing Request (CSR) must be generated. A CSR is a message sent to a CA that contains the information identifying the applicant, as well as the applicant's public key. This information (as well as the public key) is embedded in the issued certificate. It includes fields such as the organization name, city, region and country.

It is important to note that SHA-1 signed certificates are being phased out by CAs, and will not be trusted after January 1, 2017. The stronger SHA-2 family of hash algorithms is replacing SHA-1. This means if you have an older SHA-1 certificate it may need to be replaced.

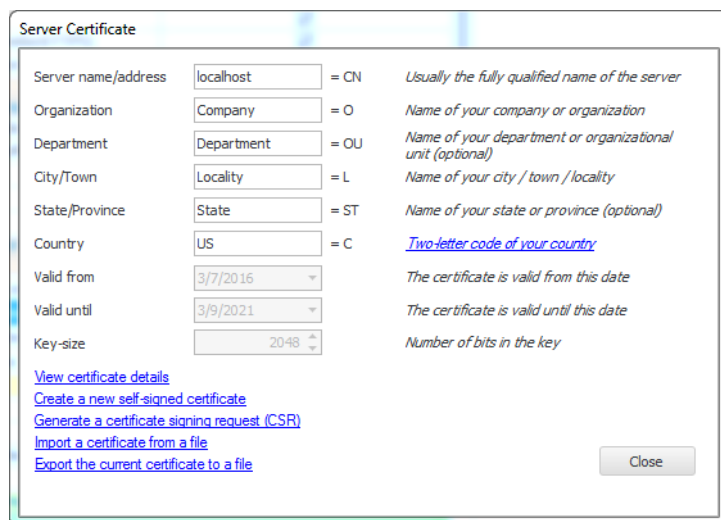
The most important field is the **Common Name (CN)** which is normally the fully qualified domain name that you wish to secure using the certificate, e.g. 'www.mydomain.com' or 'myserver.mydomain.com'. Before the CSR is sent, it must be signed by the applicant's private key. The CA sends back the certificate containing these fields, signed by the public key of the CA.

The CompleteFTP manager can generate the keypair (private and public keys) required together with the CSR to be sent to the CA. In the Site/Sites panel, the FTP/FTPS group settings must be chosen. Drill down to Advanced FTP/FTPS Settings and then to Security Settings. Click on the *Server certificate* field to display the ellipsis (...) button (see below) and select that to display the Server Certificate dialog box.



FTP / FTPS	
FTP enabled	<input checked="" type="checkbox"/>
FTPS explicit enabled (standard)	<input checked="" type="checkbox"/>
FTPS implicit enabled (legacy FT	<input checked="" type="checkbox"/>
Anonymous Logins	
Advanced FTP/FTPS Settings	
Port for FTP / FTPS explicit	21
Port to use for FTPS implicit	990
Listening IP addresses	All available
Folder listing format	Unix
Character set	Default - Unicode (UTF-8)
Passive (PASV) Transfer Settings	
Security Settings	
Clients must supply certifi	Don't require valid certificate
Server certificate (also us	C=US, S=State, L=Loc... ... X

Select the *Generate a certificate signing request (CSR)* link (see image below) to begin the process of generating a CSR. Note that a private key (in PVK format) is first generated; this key must be saved to a secure location for use with the issued certificate. Do not lose it! Your certificate is of no use without the corresponding private key. The end result of the process should be a CSR file and a private key. The CSR file is to be sent to the CA you have selected. The private key is not sent to the CA.



Server Certificate		
Server name/address	localhost	= CN <i>Usually the fully qualified name of the server</i>
Organization	Company	= O <i>Name of your company or organization</i>
Department	Department	= OU <i>Name of your department or organizational unit (optional)</i>
City/Town	Locality	= L <i>Name of your city / town / locality</i>
State/Province	State	= ST <i>Name of your state or province (optional)</i>
Country	US	= C <i>Two-letter code of your country</i>
Valid from	3/7/2016	<i>The certificate is valid from this date</i>
Valid until	3/9/2021	<i>The certificate is valid until this date</i>
Key-size	2048	<i>Number of bits in the key</i>
<a href="#">View certificate details</a> <a href="#">Create a new self-signed certificate</a> <a href="#">Generate a certificate signing request (CSR)</a> <a href="#">Import a certificate from a file</a> <a href="#">Export the current certificate to a file</a>		
<button>Close</button>		

For step by step instructions on how to generate a CSR, please refer to Step-by-step guide: [Request a CA certificate](#).

# How to purchase a CA SSL certificate

Once you have your CSR (and have your CSR's private key safely stored), you can purchase an SSL certificate from a CA.

Firstly, you must select a CA to purchase from. There are a number of CAs that sell SSL certificates, and for compatibility reasons, it is best to choose one of the more well-known providers. Some of the largest are listed [here](#). Please note that prices vary considerably, so compare prices before purchasing.

Purchasing will involve uploading your CSR to the CA's website. The CA will then want to verify your organization's identity before issuing the certificate. In particular, they will need to verify that your organization is the legal owner of the domain being set as the Common Name. They may require certain documents to be sent to them as part of the verification process, or send an email to the registered owner of the domain (to confirm you have access to this email).

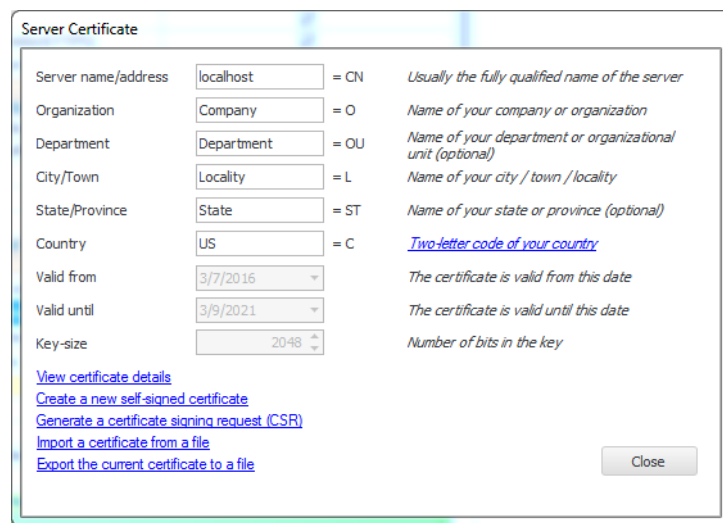


# How to install a CA SSL certificate

Your chosen CA will send you an email with a link to download your new certificate once their verification procedure is completed.

Most CAs will allow you to choose a format in which you can download your certificate. Generally, "Apache/SSL" or "Other" will be suitable for CompleteFTP - this will usually be CER or CRT file that can be directly imported.

To import into CompleteFTP, use the *Import a certificate from a file* link in the *Server Certificate* dialog box (see below). The private key that was generated with the CSR will also be required - you will be prompted to select it.



**Server Certificate**

Server name/address	<input type="text" value="localhost"/>	= CN	<i>Usually the fully qualified name of the server</i>
Organization	<input type="text" value="Company"/>	= O	<i>Name of your company or organization</i>
Department	<input type="text" value="Department"/>	= OU	<i>Name of your department or organizational unit (optional)</i>
City/Town	<input type="text" value="Locality"/>	= L	<i>Name of your city / town / locality</i>
State/Province	<input type="text" value="State"/>	= ST	<i>Name of your state or province (optional)</i>
Country	<input type="text" value="US"/>	= C	<i><a href="#">Two-letter code of your country</a></i>
Valid from	<input type="text" value="3/7/2016"/>		<i>The certificate is valid from this date</i>
Valid until	<input type="text" value="3/9/2021"/>		<i>The certificate is valid until this date</i>
Key-size	<input type="text" value="2048"/>		<i>Number of bits in the key</i>

[View certificate details](#)  
[Create a new self-signed certificate](#)  
[Generate a certificate signing request \(CSR\)](#)  
[Import a certificate from a file](#)  
[Export the current certificate to a file](#)

Close

## Importing a certificate chain

From 9.0.1, CompleteFTP supports importing the certificate chain, not just the leaf certificate (i.e. your organization's certificate).

To do this, your certificate and the intermediate certificates (usually supplied by your CA) must be combined into a single file. They can be concatenated, with your certificate placed at the top. So the combined certificate file would look something like this:

```
-----BEGIN CERTIFICATE-----
MIIFhjCCBG6gAwIBAgIQfZnwRktIbUWoel4F0LoKATANBgkqhkiG9w0BAQsFADBj

.... [your certificate data] ....

zWgmVzIYRVRpzJGr89zVnygjUmFEB7KSLhQ=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIE0jCCA7ggAwIBAgIQLGnhL2pnC9md0g+RnvCeUTANBgkqhkiG9w0BAQsFADCB

.... [intermediate CA certificate data] ....

zWgmVzIYRVRpzJGr89zVnygjUmFEB7KSLhQ=
-----END CERTIFICATE-----
```

This file is then imported into CompleteFTP in the same way as the single certificate.

For step by step instructions on how to install a CA SSL certificate, please refer to Step-by-step guide: [Install a CA certificate](#) .

If CompleteFTP was not used to generate a CSR, you may be provided with an SSL certificate and private key in different formats. See [how to import SSL certificates in other formats](#).

# How to import SSL certificates in other formats

If CompleteFTP is used to [generate a CSR](#) so that a certificate can be requested from a certificate authority (CA), then the private key that is generated along with the CSR is stored in the Microsoft PVK format. The SSL certificate is normally supplied by the CA in CRT or CER format, which can usually be [imported straight into CompleteFTP](#) along with the PVK file (which is prompted for after loading the certificate file).

If CompleteFTP is not used to generate a CSR, your SSL certificate and its private key might be in other formats. The commonest certificate and private key format is PEM-encoded. Certificates begin with "-----BEGIN CERTIFICATE-----", and private keys begin with "-----BEGIN RSA PRIVATE KEY-----". PEM supports certificates and private keys in the same file.

From version 12.1, CompleteFTP can directly import certificates in PEM format. If the certificate's private key is not included in the PEM file, you will be prompted for it. PEM private key files are also supported from 12.1.

Previous versions of CompleteFTP cannot directly import SSL certificates with PEM-encoded private keys, but it is easy to convert them to a supported format using OpenSSL. The supported format is PFX, also known as PKCS#12 format. Windows binaries of OpenSSL can be obtained [here](#). Normally one of the Win64 "Light" distributions is sufficient if you are running Windows 7 or Windows 8, or another 64 bit Windows OS.

**Please note** that OpenSSL should be run in a command prompt that is being run with administrator privileges.

Sometimes, your PEM-encoded certificate file will have the certificate authority's certificates embedded in it as well as your SSL certificate. From 9.0.1, all certificates can be imported, but for versions prior to 9.0.1 your certificate will need to be copied to a separate file and installed. Your certificate can usually be identified by the attributes listed above it (especially its common name). To import into PFX format (below), create a file that contains only your certificate (cut and paste inclusive from the "-----BEGIN CERTIFICATE-----" to "-----END CERTIFICATE-----").

Use the following OpenSSL command to convert your certificate and private key into PFX format:

```
openssl pkcs12 -export -out certificate.pfx -inkey privateKey.key -in certificate.crt
```

You can then import the PFX file directly into CompleteFTP via the *Import a certificate from a file* link in the *Server Certificate* dialogue box.

**Server Certificate**

Server name/address	localhost	= CN	Usually the fully qualified name of the server
Organization	Company	= O	Name of your company or organization
Department	Department	= OU	Name of your department or organizational unit (optional)
City/Town	Locality	= L	Name of your city / town / locality
State/Province	State	= ST	Name of your state or province (optional)
Country	US	= C	<a href="#">Two-letter code of your country</a>
Valid from	3/7/2016		The certificate is valid from this date
Valid until	3/9/2021		The certificate is valid until this date
Key-size	2048		Number of bits in the key

[View certificate details](#)  
[Create a new self-signed certificate](#)  
[Generate a certificate signing request \(CSR\)](#)  
[Import a certificate from a file](#)  
[Export the current certificate to a file](#)

Close

## Converting a private key to PVK format

Sometimes the certificate is already in PFX format, but the private key is PEM-encoded. In this situation the private key can be converted to the Microsoft PVK format. When the PFX file is imported without a private key, the private key in PVK format is then prompted for. Again, OpenSSL can be used:

```
openssl rsa -inform PEM -in mykey.pem -outform pvk -out mykey.pvk
```

Some users have found it easier to also convert their PFX certificate to CER format (as well as their private key into PVK format).

## Exporting the server certificate

The currently installed server certificate can be exported to a file from the *Server Certificate* dialogue box. It is important to note that the certificate's private key is **not** included in the exported file unless the format of the file to export to is in the PFX format. This requires the filename that is provided to have the P12 or PFX file extension. If the filename has the extension CER or PEM, the private key is not included.

# How to set up the server keys

When using SFTP, the server **must** have private keys installed. The keys are an integral part of how the secure connection is established. They identify the server to its users.

There are four types of keys - RSA keys, DSA keys, ECDSA keys, and Ed25519 keys. When clients connect, they indicate what type they prefer to be used, and a key type negotiated. In Settings->SFTP->SSH Settings->Public Key Algorithms, one or the other type of key may be disabled if, for example, only RSA is to be supported.

By default, private keys are already installed - they are generated when CompleteFTP is installed.

The private keys can be regenerated or private keys can be imported via the popup menu shown by selecting the ellipsis (...) on RSA Key, DSA key or ECDSA Key, found via Settings->SFTP/SCP->Advanced SFTP/SCP Settings, as shown below. The server's public keys can also be exported via this menu.

SFTP / SCP

SFTP enabled☒

SCP enabled☒

SSH terminal access enabled☐

SSH forwarding enabled☐

Advanced SFTP/SCP Settings

Port for SFTP/SCP/SSH22

Listening IP addressesAll available

Authentication methodsAll

RSA Key

DSA Key

Ed25519 Key

ECDSA Keys

Algorithms

Generate new private key

Import private key

Export public key in OpenSSH format

Export public key in SECSH format

Show bit-length and public key fingerprint

Export private key in new OpenSSH format

RSA key

The RSA server key - the private key that the server uses to identify itself. The server supports RSA, DSA, ECDSA and Ed25519 server keys. The keys can be imported, exported or generated via the menu that appears when selecting the ellipsis (...) in the text box displaying the key's details. RSA keys are 2048 bits in length by default.

# How to use SSH terminal access

CompleteFTP (Professional and Enterprise MFT only) supports SSH terminal access. This means that command-line SSH client can SSH onto the CompleteFTP server machine, obtain a shell and use various command-line commands.

Note that CompleteFTP must have SSH terminal access enabled (disabled by default).

SFTP / SCP		
SFTP enabled	<input checked="" type="checkbox"/>	
SCP enabled	<input checked="" type="checkbox"/>	
SSH terminal access enabled	<input checked="" type="checkbox"/>	
SSH forwarding enabled	<input type="checkbox"/>	

And the user must also have SSH terminal access enabled.

User Properties (John):	
Allowed Protocols	
FTP	<input checked="" type="checkbox"/>
FTPS	<input checked="" type="checkbox"/>
SFTP	<input checked="" type="checkbox"/>
SCP	<input checked="" type="checkbox"/>
SSH terminal	<input type="checkbox"/>
SSH forwarding	<input type="checkbox"/>
HTTP	<input checked="" type="checkbox"/>
HTTPS	<input checked="" type="checkbox"/>
File-Sharing	
Quotas and Limits	
Scripting and Shells	
JSS (Server-Side Javascript)	<input checked="" type="checkbox"/>
SSH terminal shell	UNIX
Personal Details	UNIX
Email Address	JSS

Both Windows and non-Windows (internal) users can be granted terminal access. However as non-Windows users run by default as the local SYSTEM user, for security reasons they are not permitted to exec processes.

## SSH Terminal Shell

The setting, 'SSH terminal shell', determines the type of command-line that the user is given when they log in via SSH terminal.

The 'UNIX' shell offers UNIX-like commands, such as `ls`, `pwd` and `pwdA` list may be obtained by executing the command, `help`. If an administrator logs in in this way then [administration commands](#), such as `useradd` and `folderadd`, are available.

The JSS shell is a Javascript interpreter environment, similar to that found in the developer tools in browsers like Google Chrome and Microsoft Edge. The full [JSS API](#) is available. If an administrator is connected then configuration changes may be made via the [Config](#) object returned by the `system.getConfig()` function.

## Connecting

Once the server settings (in SFTP/SCP/SSH settings) are enabled for SSH terminal access, users can login to CompleteFTP via `ssh username@hostname`, from any command-line SSH client on any operating system.

Once logged in, users are presented with a simple Unix-like command-line shell that permits certain commands, such as changing directory, creating directories, and copying and renaming files.

The `help` command lists all available commands.

## Exec command

The most useful command (and only permitted for Windows users) is `exec`, which can be used to launch programs and scripts. For example, to run a DOS command such as 'echo', you would use the standard 'cmd' program, e.g.

```
exec cmd /c echo test
```

The same technique can be used to run the DOS 'type' command to display the contents of a text file:

```
exec cmd /c type test1.txt
```

For DOS commands, the '/c' flag is necessary as it instructs the DOS shell to terminate after running the command.

# How to use SSH forwarding

CompleteFTP (Professional and Enterprise MFT only) supports local SSH port forwarding (often known as SSH tunneling). This means SSH tunnels can be established between a client machine and CompleteFTP that other protocols can use.

Why would you use SSH port forwarding (also called SSH tunneling)? There are two reasons - firewalls and security.

Consider the scenario where an employee using a laptop outside the corporate network wants to give a demonstration to a customer using a machine within the corporate network. Perhaps they want to use RDP (the Remote Desktop Protocol) and it has not been set up in secure mode. In this scenario a CompleteFTP server is also within the corporate firewall.

## Firewalls

By default, RDP uses port 3389, but say for security reasons the corporate firewall does not permit external connections to this port number. If SSH port forwarding is used, all RDP traffic is transmitted over port 22 (the standard SSH port). There is no need to open the RDP port in the firewall.

## Security

What's more, the RDP protocol is transmitted securely over the SSH tunnel. So unsecure protocols can be safely used via SSH tunnels. The only unsecure portion of the route is between CompleteFTP and the destination machine (the RDP machine), and both of these machines are inside the corporate firewall.

## Setting up forwarding

How is port forwarding set up? For security reasons, it is disabled by default in CompleteFTP. It must be enabled both for the site and individual users. To enable port forwarding for the site, the Site tab in the CompleteFTP manager must be opened, and SSH Port Forwarding flag enabled, which is under the SFTP/SSH settings section.

SFTP / SCP		
SFTP enabled	<input checked="" type="checkbox"/>	
SCP enabled	<input checked="" type="checkbox"/>	
SSH terminal access enabled	<input type="checkbox"/>	
SSH forwarding enabled	<input checked="" type="checkbox"/>	
Advanced SFTP/SCP Settings		

The users who require port forwarding must also have this option enabled, which is done in the User panel by selecting the user.

User Properties (John):	
Username	John
User type	Non-Windows
Home folder	/Home/John
Enabled	<input checked="" type="checkbox"/>
Expires	[Never]
Authentication	
File-System	
Allowed Protocols	
FTP	<input checked="" type="checkbox"/>
FTPS	<input checked="" type="checkbox"/>
SFTP	<input checked="" type="checkbox"/>
SCP	<input checked="" type="checkbox"/>
SSH terminal	<input type="checkbox"/>
SSH forwarding	<input checked="" type="checkbox"/>
HTTP	<input checked="" type="checkbox"/>
HTTPS	<input checked="" type="checkbox"/>

On the client side, the tunnel must be set up by an SSH utility such as PuTTY. When PuTTY sets up a local port forward, it listens on the client for connections on the local port specified. When a connection on the client is made to this *local* port on the local machine, the SSH tunnel is established with CompleteFTP. All the data on this local connection is sent through the tunnel to CompleteFTP, which forwards it to the ultimate destination (set by PuTTY).

# How to configure anonymous logins

FTP servers often permit anonymous logins. This means users log in as the 'anonymous' user, and don't require a password.

In CompleteFTP anonymous logins are disabled by default, but can be enabled via the Settings window, as shown below. The *Anonymous Logins* section has a checkbox for *Anonymous logins enabled*. This should be checked if anonymous logins are to be permitted.

To prevent malicious users logging in multiple times anonymously, the maximum number of anonymous logins can also be set. The default is 10, but if anonymous logins are enabled this should be set to a number in keeping with the site requirements.

FTP / FTPS ▲		
FTP enabled	<input checked="" type="checkbox"/>	
FTPS explicit enabled (standard	<input checked="" type="checkbox"/>	
FTPS implicit enabled (legacy FT	<input checked="" type="checkbox"/>	
Anonymous Logins ▲		
Anonymous logins enabled	<input type="checkbox"/>	
Max no. of anonymous conne		10
Advanced FTP/FTPS Settings ▼		

# How to use multiple sites

Every installation of CompleteFTP supports at least two *sites* - the 'Admin' site and the default site. A site consists of a related group of protocols and port numbers. Sites have their own settings, including public keys and certificate.


CompleteFTP *Enterprise MFT* supports the creation and administration of additional sites, as described [here](#).

# How to view the Admin site

CompleteFTP sites are described [here](#). Every installation of CompleteFTP supports at least two sites - the 'Admin site' and the 'Default site'.

The "Admin site" is what the CompleteFTP manager connects to, to administer the server instance. By default the server listens on port 14983 for SFTP connections from the manager. The Admin site cannot be deleted (the server could not be administered). Settings should be changed with care - if the port numbers are altered administrators must be informed of the changes.

In the *Standard Edition* and *Professional Edition*, the Admin site is configured via the [Admin](#) panel.

**ADMIN**  Step-by-step guides

Administrator username  
 [Change administrator password](#)


Administrator email address

Administrator full name

[Hide advanced administrator connection settings](#)

Advanced Administrator Connection Settings:

Site label	CompleteFTP
<b>FTP / FTPS</b>	
FTP enabled	<input type="checkbox"/>
FTPS explicit enabled (standard FTPS)	<input type="checkbox"/>
FTPS implicit enabled (legacy FTPS)	<input type="checkbox"/>
<b>Anonymous Logins</b>	
<b>Advanced FTP/FTPS Settings</b>	
Port for FTP / FTPS explicit	14982
Port to use for FTPS implicit	990
Listening IP addresses	All available
Folder listing format	Unix
Character set	Default - Unicode (UTF-8)
<b>Passive (PASV) Transfer Settings</b>	
<b>Security Settings</b>	
<b>SFTP / SCP</b>	
SFTP enabled	<input checked="" type="checkbox"/>
SCP enabled	<input checked="" type="checkbox"/>
SSH terminal access enabled	<input type="checkbox"/>
SSH forwarding enabled	<input type="checkbox"/>
<b>Advanced SFTP/SCP Settings</b>	
<b>HTTP / HTTPS</b>	
HTTP enabled	<input type="checkbox"/>
HTTPS enabled	<input checked="" type="checkbox"/>

**Advanced administrator settings** 

These settings apply only to the administrative server, which runs on a different port to that of the standard server accessed by all other user accounts.

In *Enterprise MFT*, the Admin site is displayed in *Sites* by checking 'Show system users/folders/sites' in the Options menu at the bottom left of the window.



Site	Site label	CompleteFTP
<b>Admin</b>	<b>FTP / FTPS</b>	
Default Site	FTP enabled	<input type="checkbox"/>
	FTPS explicit enabled (standard FTPS)	<input type="checkbox"/>
	FTPS implicit enabled (legacy FTPS)	<input type="checkbox"/>
	<b>Anonymous Logins</b>	
	<b>Advanced FTP/FTPS Settings</b>	
	Port for FTP / FTPS explicit	14982
	Port to use for FTPS implicit	990
	Listening IP addresses	All available
	Folder listing format	Unix
	Character set	Default - Unicode (UTF-8)
	<b>Passive (PASV) Transfer Settings</b>	
	<b>Security Settings</b>	
	<b>SFTP / SCP</b>	
	SFTP enabled	<input checked="" type="checkbox"/>
	SCP enabled	<input checked="" type="checkbox"/>
	SSH terminal access enabled	<input type="checkbox"/>
	SSH forwarding enabled	<input type="checkbox"/>
	<b>Advanced SFTP/SCP Settings</b>	
	<b>HTTP / HTTPS</b>	
	HTTP enabled	<input type="checkbox"/>
	HTTPS enabled	<input checked="" type="checkbox"/>
	<b>Advanced HTTP/HTTPS Settings</b>	
	<b>Server-side Javascript (JSS)</b>	
	<b>Public Access</b>	
	<b>File System Settings</b>	
<b>IP Filtering and Auto-Banning</b>		
<b>Limits and Timeouts</b>		
<b>Messages</b>		
<b>Settings</b>		
e HTTP and HTTPS protocols.		
Server online		
Ve		

Change skin

Show logging window

☒ Show context help

Show guide window

☒ Show system users/folders/sites

Import Filezilla Configuration

# How to set up a secure server

CompleteFTP supports secure FTP via the SFTP protocol (FTP over SSH), and the FTPS protocol (or FTP over SSL). CompleteFTP also supports SCP and HTTPS (but not in the *Standard Edition*).

## FTPS

FTPS may be enabled by selecting the "FTPS enabled" setting (see the FTP/FTPS group of settings), and its use can be made mandatory (i.e. no 'plain' insecure connections) by unselecting the "FTP enabled" setting. In this case users can only use FTPS, and ordinary FTP connections will be rejected (i.e. initial connection to the server must be followed by the AUTH command).

By default, the FTPS and FTP port is port 21.

## SFTP

SFTP may be enabled by selecting the "SFTP enabled" setting (see the SFTP/SCP group of settings). Similarly, SCP may be enabled by selecting the "SCP enabled" setting.

By default, the SFTP and SCP port is port 22.

SFTP, SCP and FTPS can be enabled simultaneously (and by default are).

## HTTPS

HTTPS (which is HTTP secured over SSL/TLS) may be enabled by selecting the "HTTPS enabled" setting (see the HTTP/HTTPS group of settings).

# How to use HTTP

CompleteFTP supports both public HTTP access to files, and authenticated access to files. HTTP directory listings are also supported (if listing permissions are in place).

## Public access

For public (unauthenticated) access, the "anonymous" user (which may be found in Users panel and by selecting "Show system users/folders/sites" in the main form's Options menu) must be enabled and the "Public HTTP access enabled" setting must be checked. Only the files accessible by the anonymous user can be viewed publicly via HTTP (i.e. by a web browser without logging in). When the server is being accessed publicly the root of file-system is always the anonymous user's home directory, thus `http://hostname/index.html` maps to `/Public/index.html` in the virtual file-system, assuming that `/Public` is the anonymous user's home directory (the default). Any access to a path that is not accessible to the anonymous user will result in an authentication request (i.e. login dialog in the browser). The user will thus be given a chance to authenticate. Alternatively, the default `index.html` file has a *Login* link that can be used to login (or this link, */login*, can be placed on any page desired).

## Private access

A particular user's files can be accessed via HTTP if the user is authenticated. For example, the user 'javaftp' normally has their files accessible under '/home/javaftp' (as long as the General User Setting *User's home folder appears as root* is not selected). So all of the 'javaftp' user's files will be accessible via '`http://hostname/home/javaftp/filepath`'. When an attempt is made to navigate to this URL in a web browser, an authentication dialog will be popped up by the browser, and the 'javaftp' user's credentials can be entered and a session established. Note that in HTTP all data sent to the server is not secure, especially usernames and passwords. If *User's home folder appears as root* is enabled, the authenticated user's files are accessed via '`http://hostname/filepath`', i.e. the '/home/javaftp' part of the URL is not required. The login url `http://hostname/login` can be used to authenticate if necessary.

## Sessions

HTTP sessions are maintained by a cookie stored in the browser. The timeout for HTTP sessions can be set in Limits & Timeouts (in Settings). A session can also be terminated by navigating to `http://hostname/logout`, and can be initiated by navigating to `http://hostname/login`.

# How to use HTTPS

The HTTPS protocol is the HTTP protocol secured by an SSL/TLS connection. It should be used when usernames and passwords (as well as any other sensitive data) is sent via HTTP. A server certificate is required for HTTPS, and one is generated on installation and can be modified via Settings->FTP/FTPS->Advanced FTP/FTPS Settings->Security Settings. The default certificate does not have your hostname, and so browsers will raise a warning when the site is navigated to using HTTPS. A [certificate authority \(CA\)](#) must be contacted to purchase a certificate for your domain to remove the browser warning. To do this, a Certificate Signing Request (CSR) must be sent to the CA. See [how to generate a CSR](#) for more details.

# How to perform HTTP uploads

CompleteFTP supports HTTP file uploads to directories where uploads are permitted by the authenticated user. This uses the standard *input* element with *type="file"* in an HTML form. For example:

```
<FORM METHOD=POST ENCTYPE="multipart/form-data" ACTION=".">  
  
  <INPUT TYPE=FILE NAME="upfile"><INPUT TYPE=SUBMIT VALUE="Upload File">  
  
</FORM>
```

If a directory is writable, the CompleteFTP directory listing page will include an HTML form permitting HTTP file uploads into this directory.

# File System

CompleteFTP has an extremely flexible virtual file system, which means that folders can be mounted from almost anywhere on the physical file system (or network) and combined in various ways.

There are six different types of folders supported:

- *Virtual folders* don't exist on the actual file system - they permit a virtual folder structure to be created.
- *Windows folders* are standard Windows folders
- *Windows special folders* are folders such as *My Documents*.
- *Network/Macro folders* are UNC (network) folders, or folders with an expanded macro. These are not supported in the Free Edition.
- *Gateway folders* are folders that a mounted from remote FTP/FTPS/SFTP servers. These are only available in Enterprise MFT.
- *Custom folder-types* may be developed in Enterprise MFT (see [here](#)).

CompleteFTP may have many root level folders mounted, and if virtual folders are used the CompleteFTP file system may not bear any resemblance to the underlying file system.

Add root folder

Add sub-folder

Remove

Name	Type	Owner	Group
DatabaseUser	Network/Macro Folder	defaultDatabase	users
FileManager	Windows Folder	admin	users
Home	Windows Folder	admin	admins
Carol	Windows Folder	Carol	users
Joe	Windows Folder	Joe	users
Mary	Windows Folder	Mary	users
Peter	Windows Folder	Peter	users
Login	Windows Folder	anonymous	users
MyDocuments	Windows Special Folder	defaultWindows	users
Public	Windows Folder	anonymous	users
Share	Share Folder	admin	users

Folder Properties:

Name	Mary
Type	Windows Folder
Path	C:\ProgramData\Enterpr...
Hidden	<input type="checkbox"/>
Access (non-Windows users)	
Owner	Mary
Group	users
Permissions	Click to view
Access (Windows users)	
Permissions	Click to view
Timestamps	

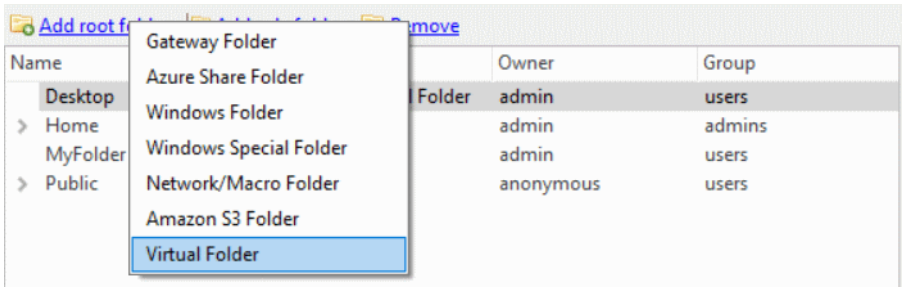
If the Hidden checkbox is checked then the folder will not show up in listings. This works independently of access permissions. For example, a folder may be shown but not accessible, or it may be hidden but still accessible.

Note that **macros** may be inserted into folder-names using the same %xyz% notation used elsewhere.

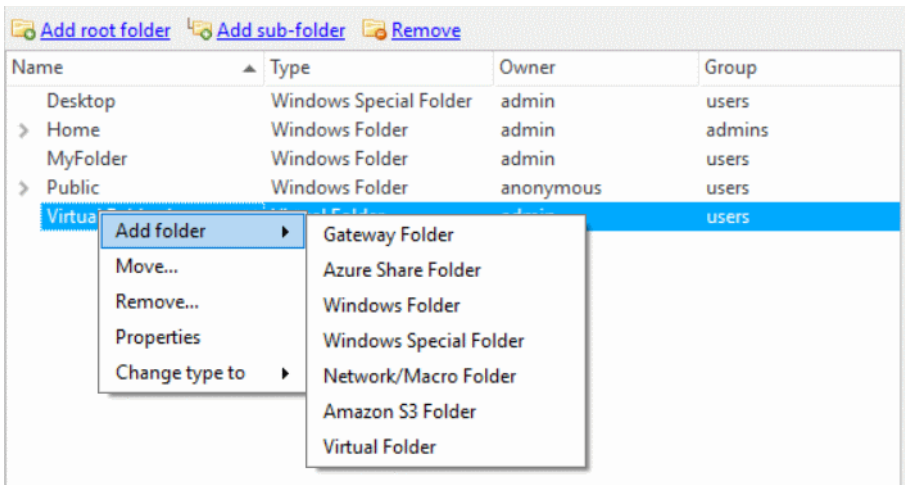
# How to create a virtual folder

Virtual folders don't physically exist. They are used to create virtual hierarchies of folders under which real folders can be placed.

CompleteFTP administrators should carefully plan out their folder structure, and create virtual folders as appropriate. For example, a virtual folder called 'home' might be created at the root level, by selecting "Add root folder" and choosing the 'Virtual folder' option, as shown below. Once created, its name may be modified from 'New VirtualFolder' by selecting this name and typing over it with the desired name.



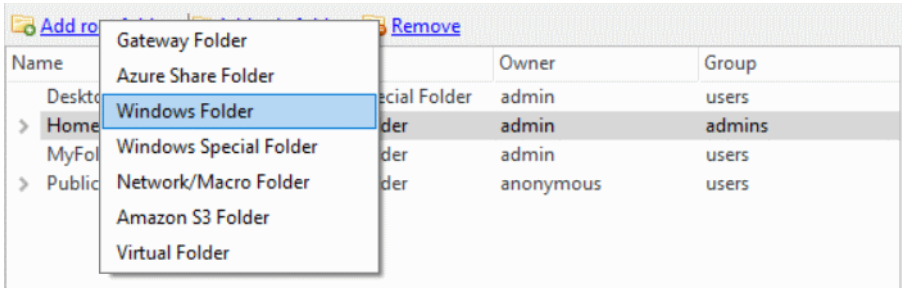
A virtual folder is of little use unless a real folder is one of its children. Right-clicking on the folder will display a *New* menu that allows creating of subfolders beneath the current folder, as shown below. For example, a Windows folder named after a user might be created here, so that it is under `/home/username` in the hierarchy - even though it may exist at a location such as `D:\Users\username\ftphome` on the disk.



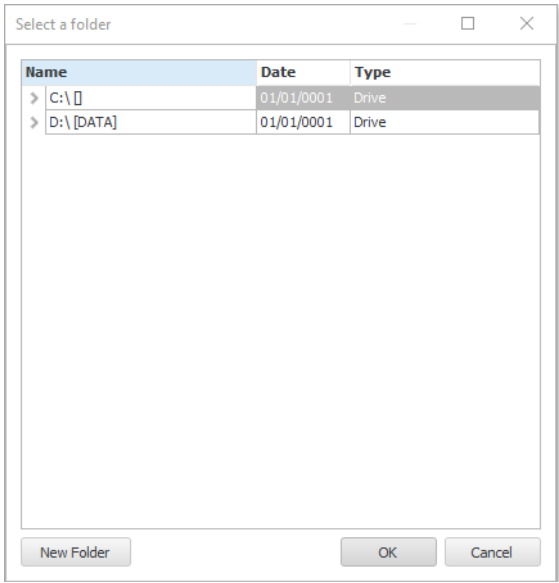
# How to add a Windows folder

Windows folders are standard folders on the local machine. They can be inserted into the virtual folder hierarchy at any point, including the root.

They are added by selecting "Add root folder" (or "Add sub-folder") and choosing "Windows folder".



A folder browser dialog displays the available drives, and a folder can be chosen.

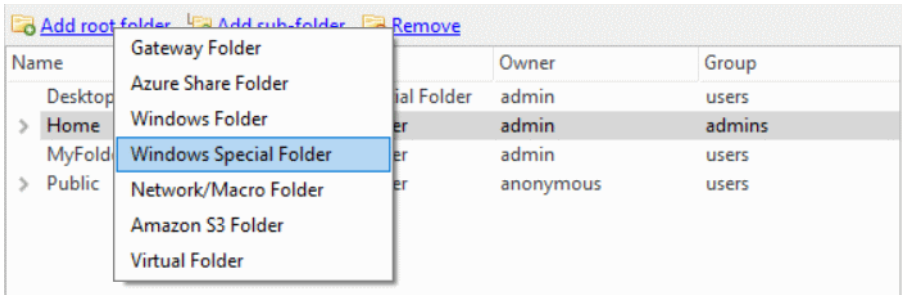




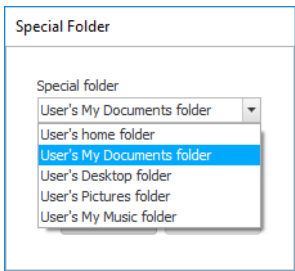
# How to add a Windows special folder

Windows special folders are folders that are predefined by Windows, and include folders such as "My Documents" and "My Music". They can be inserted into the virtual folder hierarchy at any point, including the root.

They are added by selecting "Add root folder" (or "Add sub-folder") and choosing "Windows special folder".



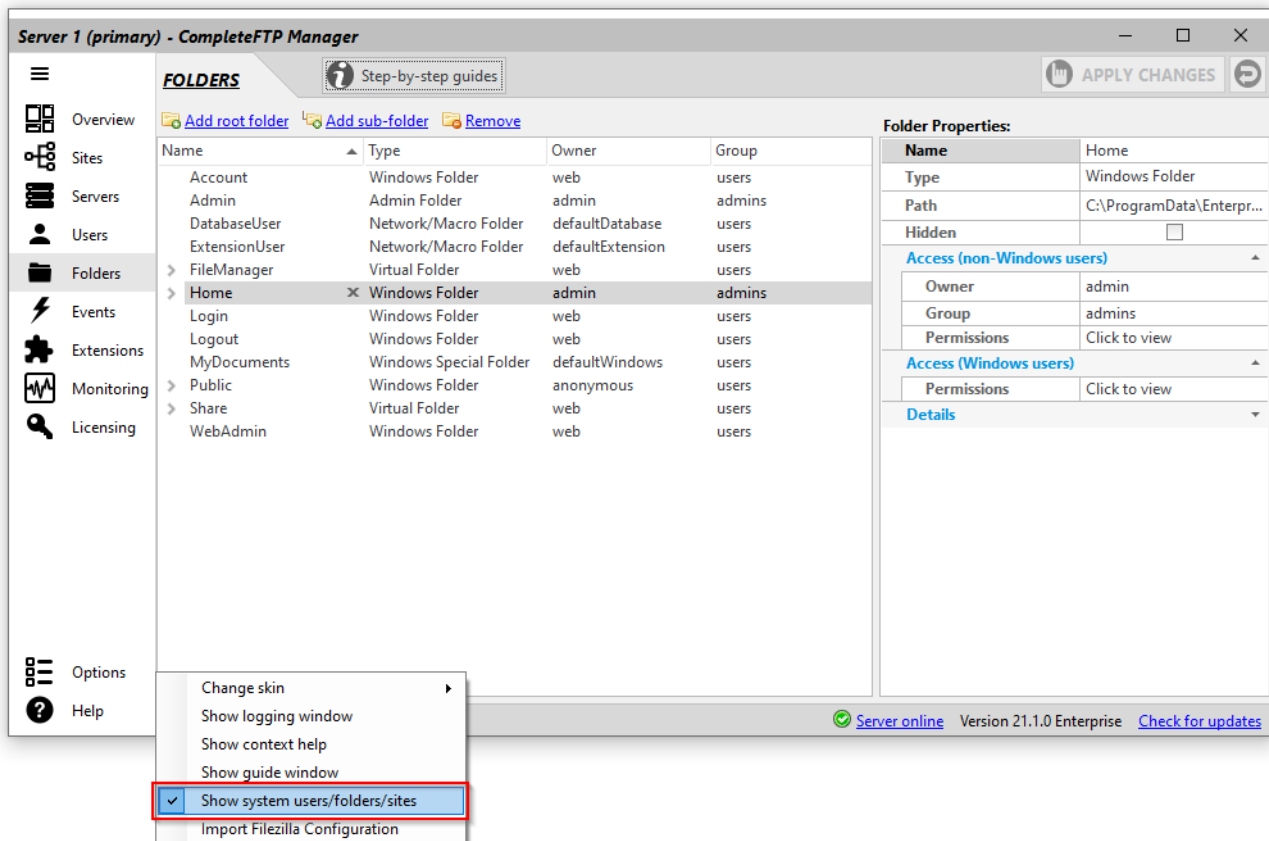
A dialog box displays the types of special folders that are available.



It should be noted that only Windows users can access special folders. They have no meaning for non-Windows users, and an error is returned if a non-Windows user tries to change into a folder that is a Windows special folder.

# How to view system folders

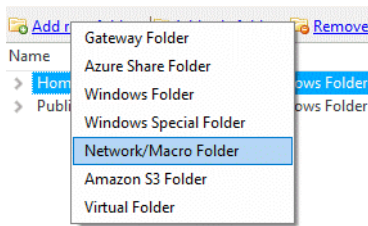
There are a number of pre-configured system folders that are not normally displayed. To view these folders, and to alter their properties, select 'Show system users/folders/sites' in the main applications's Options menu (bottom left) as shown below. Exercise caution in editing system folders, as this may have unexpected side affects.



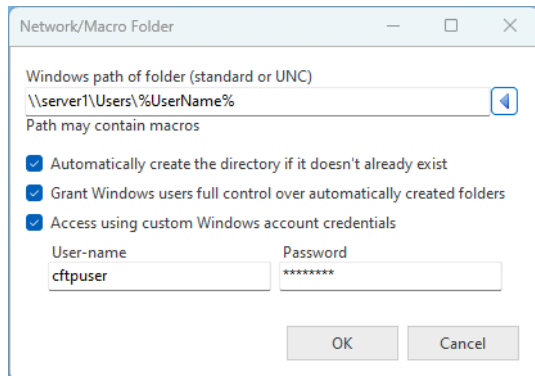
# How to create a network/macro folder

Network/macro folders are folders whose paths are entered manually. Paths may include macros, such as the name of the logged in user or an environment variable.

To add a new network/macro folder, select "Add root folder" or "Add sub-folder" and choose the 'Network/Macro folder' option.



A dialog box will appear into which the Windows path of the folder can be entered (in standard format or in UNC format). For example, `\\server1\Users` could be the UNC path used and macros can be inserted into the path, e.g. `\\server1\Users\%UserName%`.



If `Automatically create the directory if it doesn't already exist` is checked then the directory will be created when the user tries to access it. Furthermore, if a Windows user accesses a network/macro folder that has `Grant Windows users full control over automatically created folders` enabled then the permissions of the Windows directory will be modified to give that Windows user full control of the directory.

Custom Windows account credentials are the username and password of an account that permission to access the directory. They only need to be entered if neither of the following are true:

1. The connecting user is a non-Windows user and the account under which the CompleteFTP service is running does not have permission to access the directory, and
2. The connecting user is a Windows user, but that user's Windows account does not have permission to access the directory.

For detailed steps on how to create a network folder, please refer to Step-by-step guide: [Add a network folder](#).

To insert a CompleteFTP macro into the path, use the arrow button on the right hand side of the dialog box to insert the macro of your choice. For example, `"D:\users\%UserName%"`. Whenever this folder is accessed by a user the username is substituted in for the macro, so a "user1" user would be accessing `"D:\users\user1"`.

Windows [environment variables](#) may also be included by surrounding them with percent (%) characters, e.g. `%MyVariable%`. Windows has two classes of environment variables: system and user. User environment variables should not be used with non-Windows users, but for Windows users both system and user environment variables are accessible.

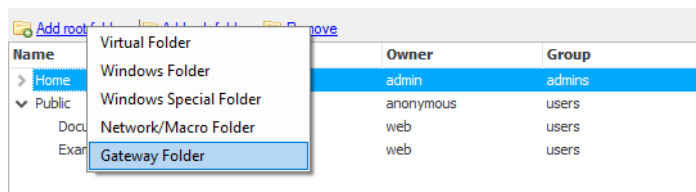
# Gateway folders

A *gateway folder* is a directory on a different server to the current CompleteFTP server instance. It's basically a remotely mounted directory (analogous to a network share). This feature is only available in CompleteFTP Enterprise MFT.

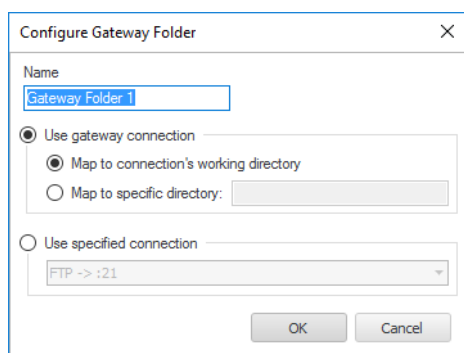
Unlike normal network shares, gateway folders are specified by protocol - either FTP, FTPS or SFTP. This means that a remote folder from an SFTP or FTPS server can be accessed by logged in users as if it were a local folder on the CompleteFTP server.

So a user who logs into CompleteFTP via FTP is able to access the files on a different SFTP server if those folders are made accessible as a gateway folder. In effect, CompleteFTP does *protocol translation*. Some scenarios that are possible are listed [here](#).

Gateway folders are created in the same way as other folders in the **filesystem** - via the *Folders* tab, and by choosing *New root folder* or *new sub-folder* - in CompleteFTP Enterprise MFT, there is a *Gateway folder* menu item.



This brings up the *Configure Gateway Folder* dialog box.



Two options are available when configuring the new gateway folder:

1. Use the gateway connection. When a user logs into CompleteFTP via a **gateway authenticator**, they are authenticated by the remote server, and a connection to that remote server is stored in their session. This is the current gateway connection. By default, it uses the connection's working directory, although a specific directory on the remote server can be provided (as long as it exists and is accessible). This means the gateway folder is only accessible by users who are authenticated by the **gateway authenticator**.
2. Use a specified connection, i.e. specify the credentials to be used to access the gateway folder as part of the folder configuration. Of course, the protocol and remote server name must also be specified. This means any user can access this gateway folder.

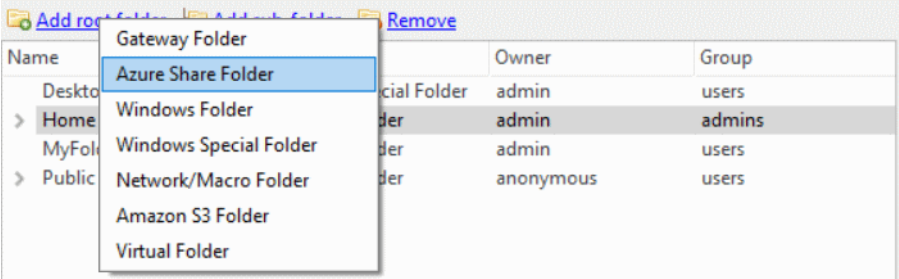
To use the gateway folder, users must change directory into it, or have the gateway folder set as their home directory. **Gateway users** require the *defaultExtension* user's home directory to be set to a gateway folder that uses the current connection (you can find *defaultExtension* user in Users panel by selecting "Show system users/folders/sites" in the Options menu at the bottom left of the window).

# Azure Share Folders

Azure Files is a Microsoft service that offers network drives hosted in the cloud or, more specifically, the Azure Cloud Computing Platform Azure Shares may be mounted in the virtual file-system of CompleteFTP Enterprise MFT. Once an Azure Share has been mounted as a folder in CompleteFTP, it may be accessed seamlessly by users in the same way that any other folder is accessed.

Before an Azure Share folder can be mounted the following are required: (1) an Azure Storage account and (2) a share in that account's File service.

Azure Share folders are added to CompleteFTP's virtual file-system in the same way as other folders in the filesystem - via the *Folders* tab, and by choosing *New root folder* or *new sub-folder* - in CompleteFTP Enterprise MFT, there is a *Azure Share folder* menu item.



This brings up the *Configure Azure File Share Folder* dialog box.

Configure Azure File Share Folder X

Account name

Account key

Share name

Directory (optional)

[Edit raw connection string](#)

OK Cancel

Each Azure storage account has a name and one or more access keys. Each share has a name and may contain a directory tree. In the dialog box, enter the account name, access key and the name of the share. You may also specify a directory within the share.

For special cases, it's also possible to enter the raw Azure File connection string. This connection string must contain an AccountName and an AccountKey attribute.

Once the changes have been applied, the folder should be accessible to connecting clients.

## Important Note

Files stored in Azure folders will not be automatically encrypted.

# Amazon S3 Folders

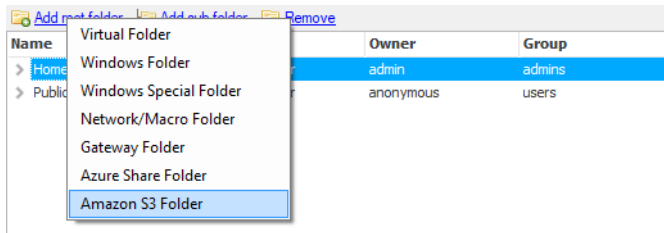
**Amazon S3** is a generic cloud data storage system offered by Amazon Web Services (AWS). S3 storage containers are called 'buckets'. S3 buckets may be mounted in the virtual file-system of CompleteFTP Enterprise MFT. Once an Amazon S3 bucket has been mounted as a folder in CompleteFTP, it may be accessed seamlessly by users in the same way that any other folder is accessed.

Access to an S3 bucket may be granted via either role-based access control or a secret access key.

Role-based access control may be used if CompleteFTP is installed on an AWS EC2 virtual machine instance. An IAM role may be assigned to an instance and this role may be granted access to a bucket. If this has been done then a secret access key is not required.

A secret access key is required in cases where role-based access control is not being used. Each AWS account can have many secret access keys. They may be created and destroyed in the Security Credentials panel of the **AWS console**. Each secret access key has an 'access key ID'.

Amazon S3 folders are added to CompleteFTP's virtual file-system in the same way as other folders in the **filesystem** - via the *Folders* tab, and by choosing *New root folder* or *new sub-folder* - in CompleteFTP Enterprise MFT, there is a *Amazon S3 Folder* menu item.



This brings up the *Configure Amazon S3 Folder* dialog box.

A screenshot of the 'Configure Amazon S3 Folder' dialog box. The dialog has a title bar with a close button (X). It contains several input fields: 'Region name' (a dropdown menu), 'Access key ID' (a text field), 'Secret access key' (a text field), 'Bucket' (a text field), and 'Directory (optional)' (a text field). At the bottom of the dialog are two buttons: 'OK' and 'Cancel'.

In the dialog box, the following may be entered: (1) the name of the region in which the bucket was created, (2) the access key ID, (3) the secret access key, (4) the bucket name, and (5) optionally the path of the directory that's to be mounted. The access key ID and the secret access key should be left blank if role-based access control is being used. If the directory is left blank then the root-level of the bucket will be mounted.

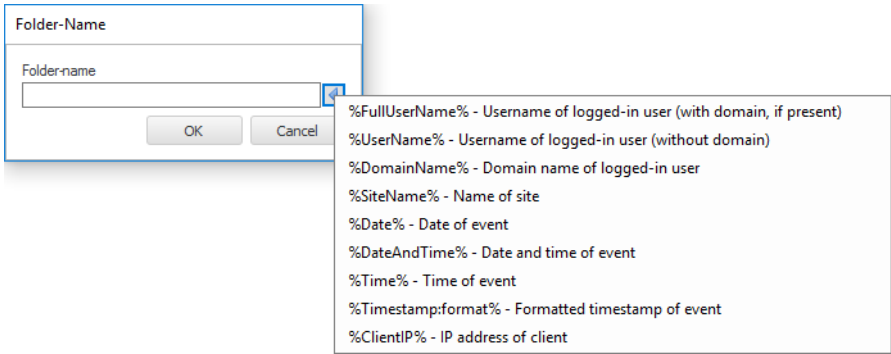
Once the changes have been applied, the folder should be accessible to connecting clients.

## Important Notes

- Clients may experience timeouts if the connection between the client and CompleteFTP is faster than the connection between CompleteFTP and Amazon.
- Directory timestamps are not available from S3 and will therefore always be given by CompleteFTP as 1/1/1901. Why? Since Amazon S3 is actually an object store rather than a file-store, it doesn't implement directories, but instead has a naming-convention called *prefixes*. Prefixes are not objects by themselves, but rather simply parts of object keys (i.e. file-names). Since they're not objects, they don't have their own timestamps.
- Files stored in Amazon S3 folders will not be automatically encrypted.
- AWS virtual machines with no Internet access require an **endpoint configured as an S3 gateway**. If a gateway to S3 isn't available then CompleteFTP has no way to access S3.

# How to insert macros into folder-names

Macros may be inserted into the name of any folder in the virtual file-system. A common case where this may be useful is in setting the name of the home-folder of **Automatic Windows Users** such that the user-name matches the name of the folder that they see when they log in. To do this, change the name of the 'MyDocuments' folder to '%UserName%'. To see the available macros, click the ellipsis (i.e. "...") button next to the name field in the folder properties panel. This shows a dialog containing a text-box with a small button to its right. Clicking this small button shows a list of the available macros.



# How to give users shared access to a folder

It's often necessary to give a group of users shared access to a folder. CompleteFTP's very flexible virtual file-system offers many ways to do this. Three such methods are described below.

1. Step-by-step guide: [Allow users to share a folder \(Method 1\)](#) (All editions)

Illustrates how to allow multiple users to access a single Windows folder by creating a subfolder under each user's home folder that maps to that Windows folder.

2. Step-by-step guide: [Allow users to share a folder \(Method 2\)](#) (Standard edition and greater)

Illustrates how to allow multiple users, with different Windows home folders, to share a common sub-folder. This is done by using a `%UserName%` macro in the Windows path of the home folder so that, even though they all have the same home folder in the virtual file-system, this folder maps to different Windows folders.

3. Step-by-step guide: [Allow users to share a folder \(Method 3\)](#) (Professional edition and greater)

Illustrates how to allow a group of users, who have their own home folder, to share access to a single folder which is not underneath their home folder. This is done by giving full access permissions to the folder to the group that these users belong to.



# How to set a folder's permissions

Access to files is controlled either by CompleteFTP itself or by the Windows file-system. For non-Windows users CompleteFTP will always control access. Windows users will be subject to Windows' own file-system permissions by default, though this can be changed (see below).

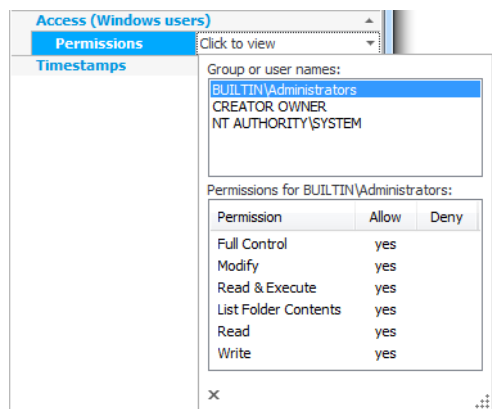
## User settings affecting file access

Access control for Windows users can be managed by CompleteFTP if the 'access control' property of those users has been manually set to 'non-Windows'. It's also possible to restrict users to read-only access by setting the 'read-only' property of those users.

File-System	
Encrypt stored files	<input checked="" type="checkbox"/>
Access control	Non-Windows
Read-only	<input type="checkbox"/>

## Managing Windows permissions

While Windows file-system permission may be viewed from within CompleteFTP Manager (see 'Access (Windows users)') they can't be modified from there. Windows Explorer must be used to edit Windows permissions.



## Managing CompleteFTP (non-Windows) permissions

Standard, Professional and EnterpriseMFT editions of CompleteFTP have enhanced permission controls from 22.0 onwards. Previously, the *Standard Edition* shared basic permission controls with the Free Edition.

### Free Edition permissions

In Free Edition, three levels of access are available for each folder:

#### Owner (full)

Owner has full permissions

#### All users (read)

Owner has full permissions, other users have read permissions

#### All users (full)

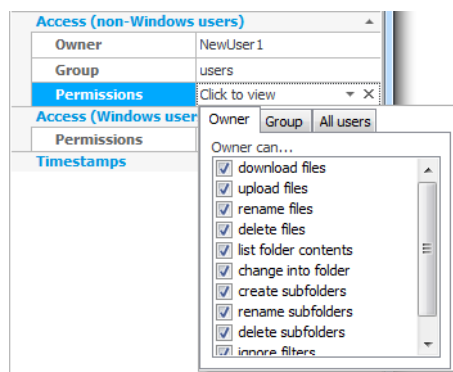
All users have full permissions

The owner of the folder can be selected in the Folder Properties section of the File System panel. The user permissions are shown as radio buttons under *Access (non-Windows users)*.

Access (non-Windows users)	
Owner	admin
Permissions	<input checked="" type="radio"/> Owner (full) <input type="radio"/> All users (read) <input type="radio"/> All users (full)

## Standard/Professional/Enterprise MFT permissions

The Standard, Professional and Enterprise MFT editions of CompleteFTP offer a permission system that's similar to that used in UNIX-like systems, such as Linux and OSX. This system was chosen because it's simple, flexible and widely known.



Permissions are managed in three different scopes: *owner*, *group* and *all users*. Each folder in the CompleteFTP's file-system will either inherit permissions from its parent folder or specify an owner, a group and a set of permissions for each of these scopes. Permissions are defined separately for the owner, the group and all users. The following permissions are available:

#### download files

files may be downloaded/read from this folder

#### upload files

files may be uploaded/written to this folder

#### rename files

files may be renamed in this folder

#### delete files

files may be deleted from this folder

#### list folder contents

the contents of this folder may be listed

#### change into folder

user may change into this folder

#### create subfolders

folders may be created under this folder

#### rename subfolders

subfolders of this folder may be renamed

#### delete subfolders

subfolders of this folder may be deleted

#### ignore filters

user is not subject to any file-name filters that are defined for this folder

A user will have permission to perform a particular action if:

1. they are the owner of the group **and** that permission is enabled for the owner, **or**
2. they are a member of the group of that folder **and** that permission is enabled for the group, **or**
3. that permission is enabled for all users

The owner of the folder can be selected in the Folder Properties section of the Folderspanel. The folder's group can be selected from the list of user groups. Any non-Windows user who is a member of that group will be granted the permissions that group has on the folder. Finally the permissions are shown. Click on "[Click to view]" and a popup will display the permissions for the folder's owner, group, and for all users. The permissions themselves are self-explanatory.

## Hidden folders

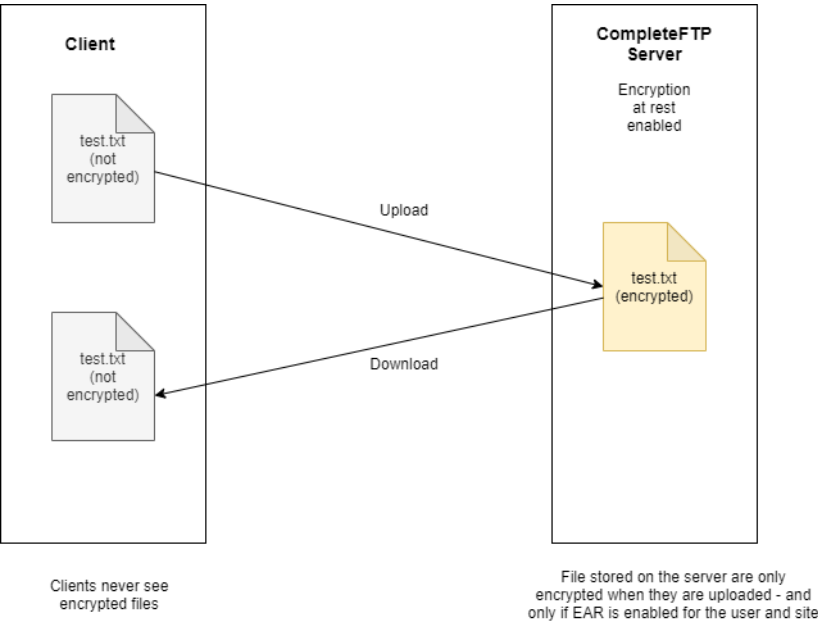
If the Hidden checkbox is checked then the folder will not show up in listings. This works independently of access permissions. For example, a folder may be shown but not accessible, or it may be hidden but still accessible.

Folder Properties:	
Name	NewUser1
Type	Windows Folder
Path	C:\ProgramData\Enterpris...
Hidden	<input type="checkbox"/>

# How to enable encryption at rest (Enterprise MFT only)

Enterprise MFT supports *Encryption At Rest* (EAR) from version 11.0. This means that if this feature is enabled for a particular user, all files transferred to the server as that user will be automatically encrypted as they are written onto the server file system. Users directly logged onto the server machine will not be able to decrypt the files. The encryption used is 128 bit AES.

Importantly, EAR is completely transparent to end users who are transferring files. They only see decrypted files as shown in the diagram below.



Once a file is encrypted, the only way to decrypt it is by transferring it (i.e. downloading it) from the server, or via the *decrypt* administrator command run from the [SSH command-line](#). Note that EAR has implications for [events](#) that might copy or post-process uploaded files. If EAR is enabled, uploaded files will not be able to be decrypted by other programs or by process triggers. For example, as FTP scripts are executed by an external program, they cannot decrypt encrypted files. Batch files and standalone executables also cannot decrypt server files. The exception is [JSS process triggers](#), which are fully integrated into the server filesystem and can decrypt files. These should be used if encrypted files need to be decrypted by process triggers.

To enable encryption at rest for a user, the user setting "Encrypt stored files" must be enabled, as well as the site-wide "Encrypt stored files" setting. Turning off the site-wide setting disables encryption at rest for all users. Files that are already encrypted will still be automatically decrypted when downloaded, but no new files will be encrypted.

Any files uploaded while EAR is enabled will be encrypted. The sizes of these files will be a few bytes larger than the decrypted file. By default, CompleteFTP will show the encrypted file size in directory listings, although querying the attributes of individual files will give the decrypted size. If the user setting "Report decrypted size" option is enabled then directory listings will show the decrypted file size. Since it takes time to determine whether or not a file is encrypted, the time taken to get a directory listing will be SIGNIFICANTLY SLOWER if this option is enabled. It's therefore advised that the option is disabled unless the decrypted file size is specifically required.

User Properties (test):

Username	test
User type	Non-Windows
Sites & home folders	Default Site1
Home web-app	[Site default]
Enabled	<input checked="" type="checkbox"/>
Expires	[Never]
Authentication	
Methods (SSH)	All
Password	*****
Public keys	RSA: 1, DSA: 0, ECDSA: 0
Allowed IP addresses	All
File-System	
Home folder is root	<input type="checkbox"/>
Encrypt stored files	<input checked="" type="checkbox"/>
Report decrypted size	<input checked="" type="checkbox"/>
Access control	Non-Windows
Read-only	<input type="checkbox"/>
Allowed Protocols	
FTP	<input checked="" type="checkbox"/>
FTPS	<input checked="" type="checkbox"/>
SFTP	<input checked="" type="checkbox"/>

File System Settings

Read-only	<input type="checkbox"/>
Show hidden files/folders	<input type="checkbox"/>
Allow backslash separators	<input type="checkbox"/>
Filename blocking filters	Filters set
Allow overwrite on rename	<input type="checkbox"/>
ZIP-File Navigation	
Enabled	<input checked="" type="checkbox"/>
ZIP-folder extension	.folder
Encryption At Rest (EAR)	
Encrypt stored files	<input checked="" type="checkbox"/>

# How to list a folder-tree (Professional and Enterprise MFT)

CompleteFTP has the ability to obtain the listing of an entire folder tree in a single operation. This saves having to recurse through the tree, which can be time-consuming. The listing is available through a file called `_treeList.json` which is listed when the *Folder-tree file* option is enabled. This option is in the *Details* category of the folder properties panel. It's off by default.

This feature was developed in cooperation with the developers of [SyncBack Pro/SE](#), which, since version 9, uses it to speed up synchronization.

Details <span>▲</span>	
Folder-tree file	<input type="checkbox"/>
Created	2/15/2019 2:27:45 PM
Modified	3/6/2019 3:46:33 AM

## Format of the folder-tree file

Although the extension of the file is `.json`, the format has been optimised slightly in order to reduce its size since many folder-trees are extremely large. The optimisation is that field names are not in quotes, which, in the opinion of the developer, should've been optional anyway.

`_treeList.json` contains a single line per file or directory. It looks something like this:

```
[
{"n:"file001.txt",l:1024,t:"2018-08-05T07:08:57Z"},
{"n:"file002.txt",l:1024,t:"2018-08-05T07:08:57Z"},
{"n":"/directory11",t:"2018-08-05T07:08:57Z"},
{"n:"file111.txt",l:1024,t:"2018-08-05T07:08:57Z"},
{"n:"file112.txt",l:1024,t:"2018-08-05T07:08:57Z"},
{"n":"/directory11/directory12/",t:"2018-08-05T07:08:57Z"},
{"n:"file121.txt",l:1024,t:"2018-08-05T07:08:57Z"},
{"n:"file122.txt",l:1024,t:"2018-08-05T07:08:57Z"}
]
```

where `n` = name, `l` = length (only for files) and `t` = timestamp (UTC)

If the name starts with a forward slash then it's a directory, otherwise it's a file. For directories, the name is the full path with the directory in which the folder-tree file is placed being the root. For files the name is just the file-name without the path. Files at the beginning of the listing (i.e. not preceded by a directory) are in the same directory as the folder-tree file. Files listed immediately below a directory are in that directory.

## How to extend the filesystem (Enterprise MFT only)

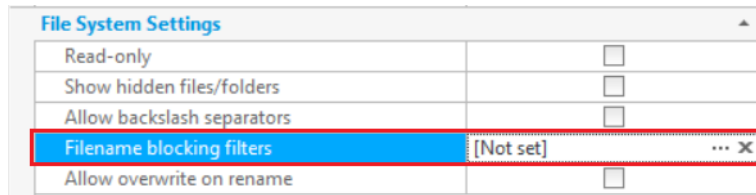
The regular file-system can be completely replaced or extended by one of your own design. When a virtual folder is associated with your extension, your code will be called by the server whenever a client performs file operations (e.g. reading, writing, directory-listing) in that folder.

For example, your extension may read from and write to a database instead of the Windows file system.

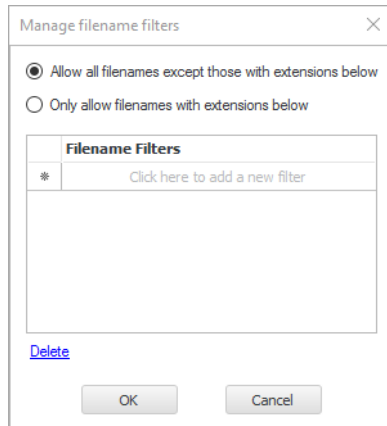
See [file system extensions](#) for more details.

# How to stop executables being uploaded

Sometimes it is a requirement to block the upload of certain file types, such as executable files. This can be done by using filename filters, in the [Filesystem settings](#) of the site.



Select the ellipsis to bring up the Manage Filename Filters dialog, as shown below and add the filename masks required. For example, to match executable files, use `*.exe`. Multiple filters can be added to match various file types.



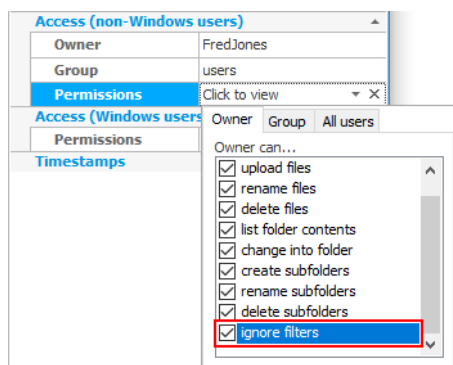
The filters apply to all protocols - SFTP, SCP, FTP/FTPS and HTTP/HTTPS.

These filters can be applied in one of two ways. The default is to **block** the upload of filenames that match the filters. But this can be inverted by selecting the radio button to only **permit** files that match the filters.

It is important to realize that once set up, these filename filters apply to all files that are uploaded to the site - and to files that are being renamed (filename filters also prevent renaming files to banned names).

## Ignore Filters permission

The only way to avoid filename filters is by specifically adding the "ignore filters" permission to a virtual directory. Then all users who have this permission will be able to upload any files they wish.



# How to mount ZIP files as directories

ZIP files contain many files combined into one compressed archive. When residing on a file server, normally the entire ZIP file would need to be downloaded to access a single file within the ZIP file.

The Professional and Enterprise MFT editions support ZIP file navigation. When enabled, ZIP files are presented to clients as though they are folders. This means that users can change into folders within the ZIP file and download specific files. This can save time and bandwidth as it allows users to download only what they need from a ZIP file instead of downloading the whole ZIP file.

By default, both the ZIP file itself and the folder that corresponds to the ZIP file are shown in listings, the latter having the additional extension `.folder`. The additional extension can be changed by modifying the 'ZIP-folder extension' setting. If this setting is cleared completely, then the ZIP file won't be displayed, being replaced by the corresponding folder.

ZIP-File Navigation	
Enabled	<input checked="" type="checkbox"/>
ZIP-folder extension	.folder

Currently ZIP files cannot be updated on the server - they are read-only. This feature may be added to a future release.

**Related Feature:** CompleteFTP also allows users to download a folder as a ZIP file by appending `.folder.zip` to the folder's path. This is an independent feature that generates a ZIP file of the folder in real-time without writing it to disk.

- **Navigating into a ZIP File:** Use `.zip.folder` to browse the contents of a ZIP file.
- **Downloading a Folder as ZIP:** Use `.folder.zip` to download an entire folder as a ZIP file.

For more information on downloading folders as ZIP files, see [How to download a folder as a ZIP file](#).

See the [filesystem settings](#) for more details.

# How to download a folder as a ZIP file

CompleteFTP allows users to download an entire folder or directory as a single ZIP file via FTP, FTPS and HTTPS (SFTP/SCP support to be added). This can be done by downloading a file whose path is the same as the folder but with `.folder.zip` appended to it. When this path is accessed, the ZIP archive is generated on the fly and is not written to disk.

This feature is particularly useful for users downloading from the command line with scripts or through APIs. Since these are virtual files and not actual files on the server, they won't appear in client applications like FileZilla, CyberDuck, or WinSCP. To use this feature with such clients, you would need to enter the path explicitly.

For example, to download the folder `/documents/reports` as a ZIP file, you would download `/documents/reports.folder.zip`.

If you're using the online file manager, you can simply right-click on a folder and select the option to download it as a ZIP file. This action utilizes the same feature internally.

**Difference from ZIP File Navigation:** This feature is independent of the [ZIP file navigation](#) feature, which allows users to navigate into a ZIP file as though it's a folder by appending `.zip.folder` to the ZIP file's name. While `.folder.zip` is used for downloading folders as ZIP files, `.zip.folder` is used for navigating into existing ZIP files on the server. Note the reversal of the extension

- **Downloading a Folder as ZIP:** Appending `.folder.zip` to a folder's path to download it as a ZIP file.
- **Navigating into a ZIP File:** Appending `.zip.folder` to a ZIP file's name to browse its contents as if it were a directory.

It's important to note the difference in extensions to use these features correctly.

**Note on legacy behavior:** This ability to download a folder as a ZIP file has been an undocumented feature of CompleteFTP since version 13.0.0 (released August, 2020), but used a different extension. It was changed to `.folder.zip` for version 24.1.0 due to unintended behavior when the different extension was used.



# How to enable backslash separators

Most FTP servers including CompleteFTP use Unix-style forward slash separators in paths. Occasionally it is desirable to be able to use Windows-style backslash separators - for example if there are clients accessing the server that only support backslash separators. CompleteFTP does not support backslash separators by default, as shown below, but enabling this option enables support for backslash separators (forward slash separators are still supported if this option is enabled).

File System Settings	
Read-only	<input type="checkbox"/>
Show hidden files/folders	<input type="checkbox"/>
Allow backslash separators	<input type="checkbox"/>
Filename blocking filters	[Not set]
Allow overwrite on rename	<input type="checkbox"/>
ZIP-File Navigation	
Enabled	<input checked="" type="checkbox"/>
ZIP-folder extension	.folder
Encryption At Rest (EAR)	
Encrypt stored files	<input type="checkbox"/>

See the [filesystem settings](#) for more details.

# Web File Manager

CompleteFTP Professional and Enterprise MFT include file-manager web-apps that allows uploading, downloading and management of files that are hosted on the server. They work both on desktop computers and mobile phones.

There are three file-manager web-apps. The default is currently *FileManager2*. *FileManager3* is the newest file-manager and will be made the default for new installations in the near future. *FileManager1* is a legacy application and is no longer maintained, so it's not recommended for new users. See [here](#) for instructions on how to change the file-manager that users see.

/Home/javaftp			
<div><div><div></div><div></div><div></div><div></div><div></div><div></div></div><div><div></div><div></div><div></div><div></div><div></div><div></div></div></div>			
<div><div><div></div><div></div><div></div><div></div><div></div><div></div></div><div><div></div><div></div><div></div><div></div><div></div><div></div></div></div> <div>&gt; <div><div></div><div></div></div> New folder</div>	Name ▼	Size	Date
	<div><div></div><div></div></div> New folder		2/4/2019 4:10:45 PM
	<div><div></div><div></div></div> ftp-codes.java	16.4 kB	2/4/2019 4:10:27 PM
	<div><div></div><div></div></div> image02.png	16.3 kB	2/4/2019 4:10:39 PM
	<div><div></div><div></div></div> winscp.log	1.5 MB	2/4/2019 4:10:39 PM

All users connecting to a CompleteFTP public site are automatically brought to the File Manager app after they've logged in, unless the server is configured otherwise.

It supports the following operations:

- [Uploading files](#)
- [Downloading files](#)
- [Viewing files](#)
- [Deleting files](#)
- [Renaming files](#)
- [Adding folders](#)
- [Deleting folders](#)
- [Renaming folders](#)
- [Sharing files privately](#)
- [Sharing files publicly](#)
- [Changing language](#)
- [Changing list view type](#)

# Web File Manager: Uploading Files

To upload a file first you select the folder you wish to upload the item to then select the upload icon (arrow points to a cloud).



Once selected you will then be prompted to upload the file to the selected folder.

Upload files

Files will be uploaded to /Example Folder

SELECT FILES

☐ Overwrite destination?

CANCEL

UPLOAD

Once you have chosen the file you wish to upload, you will have a chance to view it then delete it if necessary before you upload it to the folder of your choosing.

Upload files

Files will be uploaded to /Example Folder

SELECT FILES

EnterpriseDT.png

11.3 kB


x

☐ Overwrite destination?

CANCEL

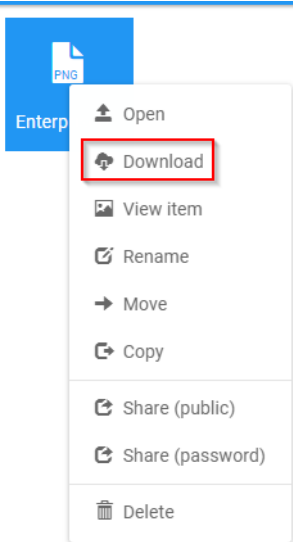
UPLOAD

Once you have uploaded the file you will then be able to view the file in the folder it has been uploaded to.

Name ▼
 EnterpriseDT.png

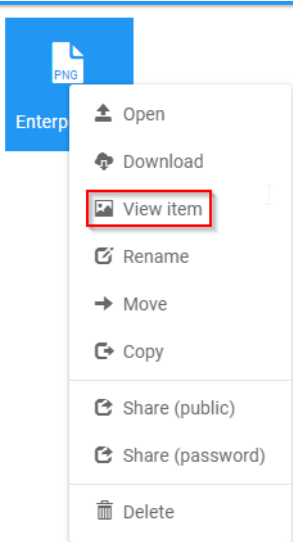
# Web File Manager: Downloading Files

To download a file you select the file that you would like to download, then select download.

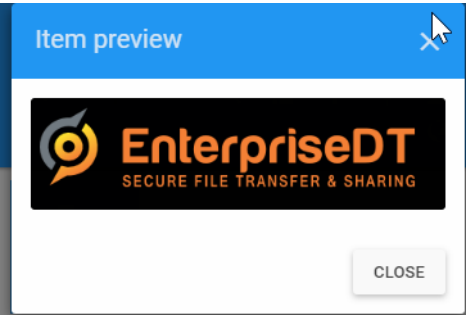


# Web File Manager: Viewing Files

To View a file first you select the file you wish to view then select then you click on view item.

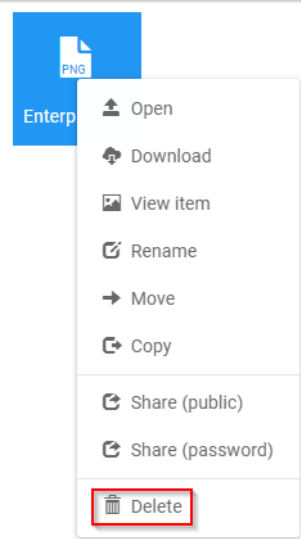


Once selected you will then be able to view the file you have chosen to view.



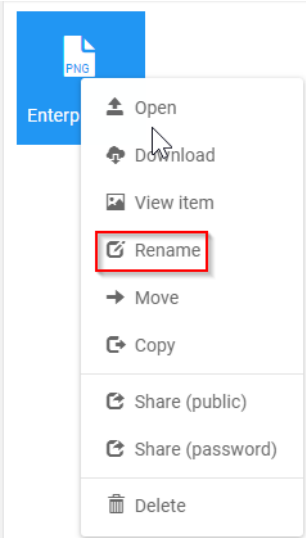
# Web File Manager: Deleting Files

To delete a file you select the file that you would like to delete, then select delete.



# Web File Manager: Renaming File

To rename a file you select the file that you would like to rename, then select rename.



Once selected you will then be prompted to rename the file, then select the rename icon.

Rename

×

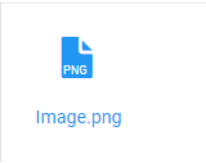
Enter new name for EnterpriseDT.png

Image.png

CANCEL

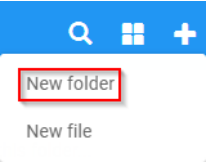
RENAME

Once that has been done the icons name has changed.

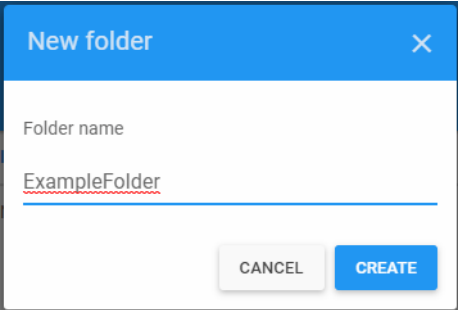


# Web File Manager: Adding Folders

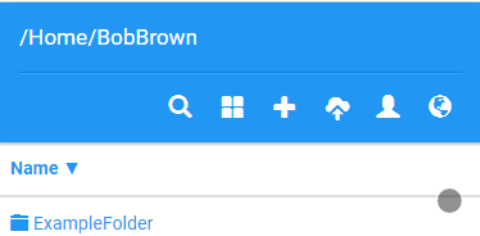
To add a folder, Select the plus icon and select New Folder



Once the New Folder has been selected you will be prompted to create a name for the folder, then select the create button



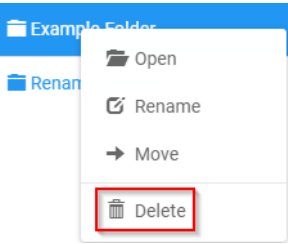
Once it has been created the folder will be shown in the home folder





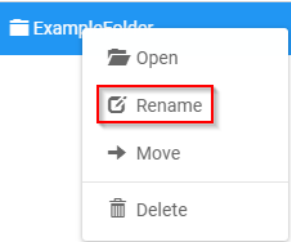
# Web File Manager: Deleting Folders

To delete a folder you select the folder that you would like to delete, then select delete.



# Web File Manager: Renaming Folders

To Rename a folder you select the folder that you would like to Rename, then select select Rename.



Once rename has been selected you will then be prompted to Renaming the folder, then select the Rename button

Rename

×

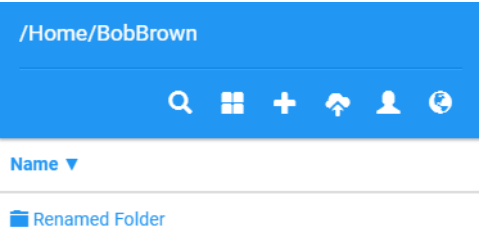
Enter new name for **ExampleFolder**

Renamed Folder|

CANCEL

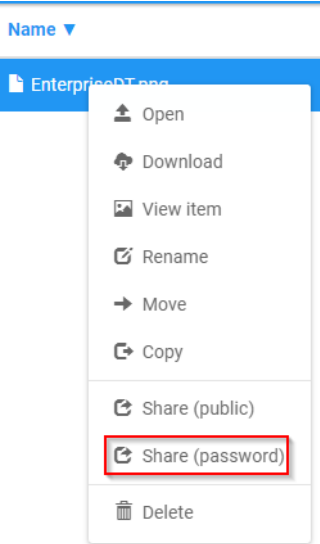
RENAME

Once the folder has been renamed you will then see the renamed folder in your directory.



# Web File Manager: Share Files Privately

To Share a file privately first select the file you wish to share then select the share(password) icon.



Once selected you will then be prompted to input a password of your choosing for the file you wish to have password protected.

Share (password)

Password

.....

Confirm password

.....

SHARE

CANCEL

Once that has been done you will be given a link that you will be able to share.

Share

Anyone with the following link can access the file:

ENTERPRISEDT.PNG

COPY

OK

Once you have retrieved the link you are now able to share it how ever this link requires the password that you placed on it.

//localhost/Share/Yal1Irfy47z+cPYUpdRHCA/EnterpriseDt.png

Password required

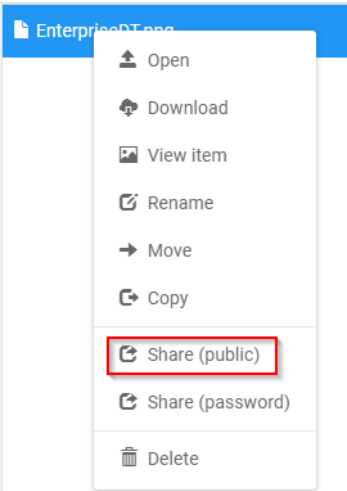
P

password

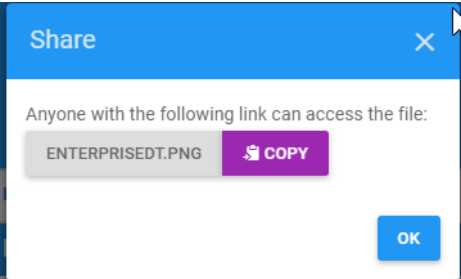
Download

# Web File Manager: Sharing Files Publicly

To Share a file Public first select the file you wish to share then select the share(public) icon.

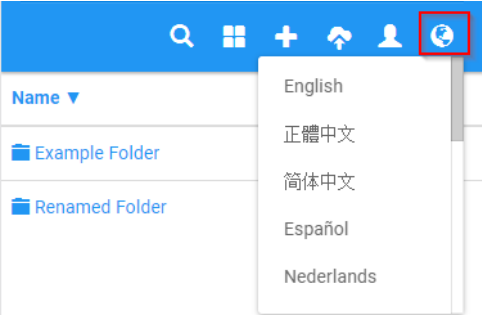


Once selected you will be given a link that you will be able to share publicly.

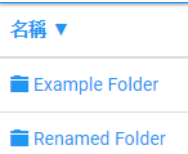


# Web File Manager: Changing Languages

To change the language for the file manager click on the Earth icon then select the language you wish to view the file manager in.

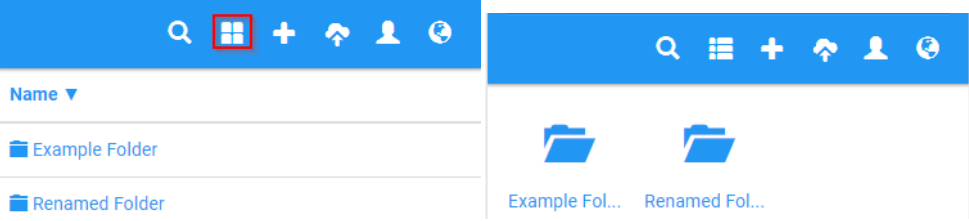


Once the language you wish to have file manager has been displayed, you will then be able to view your file manager in the language of your choosing keep note that if your folders and files have been written in English they won't be translated.



# Web File Manager: Changing List View Type

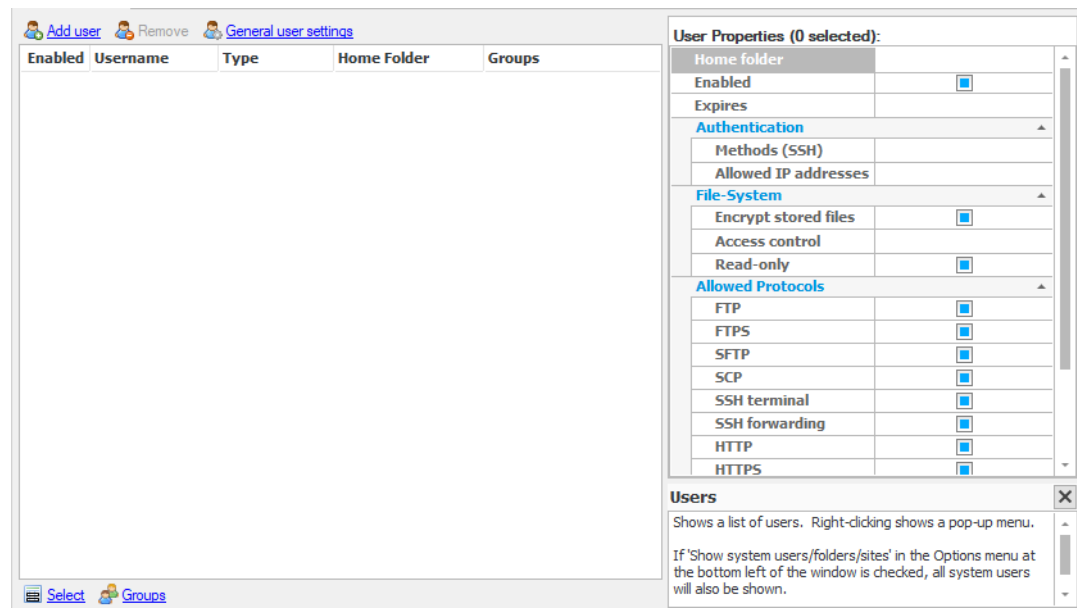
In the File Manager you are able to view your folders and your files in a list format or a tile format to do this click on the window icon once it has been selected you are able to toggle which way you wish to view your files and folders.



# Users

CompleteFTP supports unlimited numbers of users. All editions support non-Windows users. Windows users are supported in Standard, Professional and Enterprise MFT.

The *Users* screen displays the current users, and ways to add, modify and delete users.



If an existing user's details are selected in the list of users shown, a panel at the right enables the details to be changed, such as the username, password, and home directory (For more information on changing the user's home directory, please refer to Step-by-step guide: [Change the home folder of a user](#)). The allowed protocols for the user are also displayed - as well as on a site basis, protocols can be enabled or disabled for each user. The user can also be disabled by unchecking the *Enabled* checkbox.

Note that CompleteFTP supports the Automatic Windows Users (AWU) feature. This allows the administrator to enable login of any Windows user without having to manually add the user to CompleteFTP's internal user list. See [how to configure automatic Windows users](#) for more details.

User data can be [exported](#) by right-clicking on the user-list and selecting 'Export users'.

# General user settings

The general user settings that are available to configure *CompleteFTP* sites are described below. They are found on the *Users* tab, and can be displayed by selecting the *General user settings* link. Note that settings must be saved via the *Apply Changes* button, otherwise they will be lost.

General User Settings

For site: 

Default Site

☐ Allow anonymous FTP users

☒ Allow public HTTP access

☐ Home folder is root (for site)

Default Windows domain

[Password policies](#)

Close

General User Settings

General user settings have been moved to the Users panel.

## General settings

Setting	Description
Allow anonymous FTP users	If checked, anonymous access is enabled. This means connections can be made by logging in as an anonymous user rather than as a named user with a password. Anonymous users are represented by the user called 'anonymous'. By default the home-folder of this user is /Public, so this is the folder that will be publicly accessible if this option is enabled.
Allow public HTTP access	If checked, unauthenticated users will be able to access any file in the public folder. Unauthenticated users are represented by the user called 'anonymous'. By default the home-folder of this user is /Public, so this is the folder that will be publicly accessible if this option is enabled.
User's home folder appears as root	If this option is enabled then the user will see their home directory as the root (i.e. /). If it is disabled then the user will see the absolute virtual file-system path, e.g. "/home/javaftp". By default this option is not checked.
Default Windows domain	If a default Windows domain is set, it means that Windows users on the default domain can login without specifying the domain (instead of using MyDomain\MyUserName they can use MyUserName). This can be very convenient for users. The default Windows domain tells CompleteFTP which domain to authenticate against if (1) a domain-name was not given and (2) no exact match was found in CompleteFTP's list of users. See <a href="#">How to set a default Windows domain</a> .

## Other authentication methods

Users may be authenticated by other methods such as:

- **Automatic Windows** (all editions) - enables login of any Windows user without having to manually add each user to CompleteFTP's internal user-list (see [here](#)).
- **Database** (Professional and Enterprise MFT) - authenticates against an external data-source such as a database table or spreadsheet (see [here](#)).
- **Gateway** (Enterprise MFT) - authenticates via another FTP or SFTP server (see [here](#)).
- **SAML Single Sign-On** (Enterprise MFT) - delegates authentication to a SAML IDentity Provider (IDP) (see [here](#)).
- **Custom** (Enterprise MFT) - write your own authentication method using any .NET language (see [here](#)).

## Log-in-as Users

Users that are authenticated via one of these methods are subject to the settings of the specified **'log-in-as'** user. For example, if Automatic Windows users are only to be allowed to connect via SFTP then only SFTP should be enabled for the log-in-as user that's defined for the Automatic Windows authenticator, which is the defaultWindows user by default (note that this user can be found in Users panel by checking "Show system users/folders/sites" in the Options menu at the bottom left of the window).

## Password policies

Password policies can be found by selecting the "Password policies" link on the General User Settings dialog box, and are described [here](#).



# How to add a user

Both Windows and non-Windows users can be added to CompleteFTP. By default, they will be assigned home directories within the special folder for Common Application Data.

On most Windows machines from Windows 7 onwards, and Server 2008 onwards, the parent directory is:

```
C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Users
```

On Windows XP and Server 2003 the parent directory is:

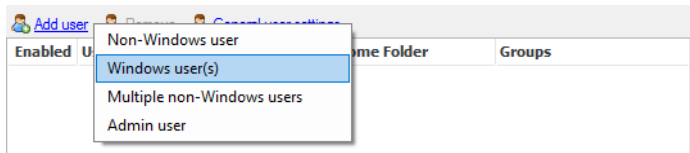
```
C:\Documents and Settings\All Users\Application Data\Enterprise Distributed Technologies\Complete FTP\Users
```

See [how to add a Windows user](#) for Windows users. Windows users can also be enabled automatically - see [how to configure automatic Windows users](#).

See [how to add a non-Windows user](#) and [how to add multiple users](#) for non-Windows users, and [How to configure external users](#) for users stored in an external database.

# How to add a Windows user

To add an existing Windows user, click the *Add user* link in the Users panel and then select *Windows user(s)* from the menu.



This brings up the *Add Windows User* dialog box.

Add Windows Users

Windows users on the server are shown below. Accounts that have already been added are not shown. Hold Ctrl to select multiple users.

Username	Full Name	Description	Home Folder	Primary Gro...	Last Login	User Flags
----------	-----------	-------------	-------------	----------------	------------	------------

List server's local users

OR

Domain name

☐ Login when querying domain

User-name filter

\*

List size-limit

10000

List domain users

Add selected user(s)

☒ Automatically create home folders ...

Close

When adding Windows users from the local server then click *List server's local users*. When adding Windows domain users (not available in Standard Edition) enter the domain name and click *List domain users*. For domains with a very large number of users the size of the list can be restricted and/or a filter can be provided to reduce the number of users listed.

Once the list of users has been fetched and displayed as in the example below, select the users to be added from the list. Hold down the shift or control keys to select multiple users.

Add Windows Users

Windows users on the server are shown below. Accounts that have already been added are not shown. Hold Ctrl to select multiple users.

Username	Full Name	Description	Home Folder	Primary Gro...	Last Login	User Flags
▸ auburn.local\Administrator		Built-in account for a...			01/01/0001	None
auburn.local\Guest		Built-in account for g...			01/01/0001	None
auburn.local\krbtgt		Key Distribution Cent...			01/01/0001	None
auburn.local\sshd		copSSH privilege sep...			01/01/0001	None
auburn.local\joesmith	Joe Smith				01/01/0001	None
auburn.local\fredjones	Fred Jones				01/01/0001	None

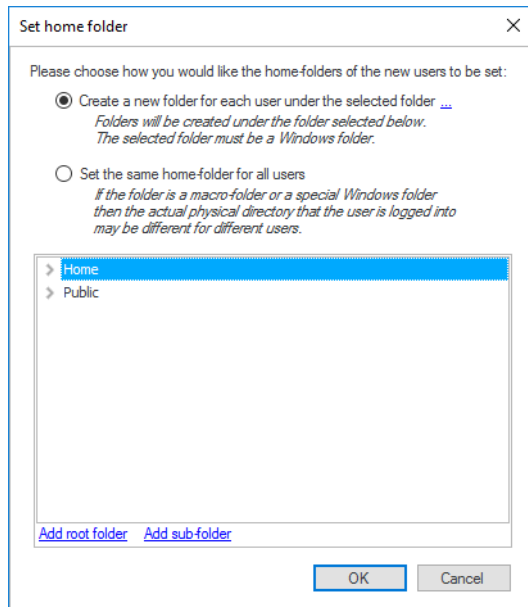
Add selected user(s)

☒ Automatically create home folders ...

Close

If a single user is being added and the checkbox labeled *Automatically create home folders* is checked then a home-folder with the same name as the user will be created automatically. The folder will be created under the `/Home` folder in the virtual file-system, and by default will be mapped to a Windows directory underneath the directory to which `/Home` is mapped. The mapping can be changed by clicking the ellipsis (...) to the right of the label of the checkbox. By default, the mapping is created using the template `%HomeBaseFolder%/%UserName%`, but this can be changed as desired.

If multiple users are selected when the *Add selected user(s)* button is clicked then a dialog-box will pop up.



In this dialog-box you can choose to create a home-folder in the same manner as described above for single users, or you can choose to use the same directory for all users.

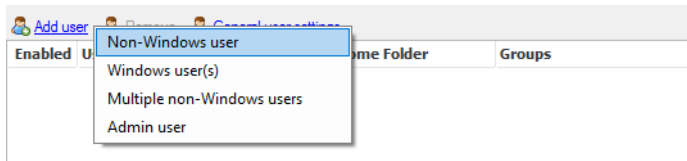
Once all users have been added click *Close* to close the dialog.

Back in the Users window, all the users that have just been added will be selected, so that any properties that must be set commonly for all users may be set.

For detailed steps on adding a Windows user, please refer Step-by-step guide: [Add a Windows user \(Standard Edition\)](#) if you're using Standard Edition, or Step-by-step guide: [Add a Windows user \(Professional/Enterprise MFT\)](#) if you're using Professional or Enterprise MFT.

# How to add a non-Windows user

*Add a non-Windows user* should be selected to add a user that does not exist as a current Windows user on the server machine (or domain). This means an FTP user does not have to exist as a Windows user.



The username and details should be entered in Step 1.

A screenshot of the 'Add User' form, Step 1: Set Username. The form has a title 'Add a non-Windows user' and a subtitle 'Step 1: Set Username'. It contains four input fields: 'Username' (filled with 'JoeSmith'), 'Full name (optional)' (filled with 'Joe Smith'), 'Description (optional)' (empty), and 'E-mail (optional)' (filled with 'joesmith@domain.com'). At the bottom are three buttons: 'Previous', 'Next', and 'Cancel'.

A password and/or public keys must be set in Step 2. Public keys are only used in SFTP, SCP and SSH (Standard Edition or better).

A screenshot of the 'Add User' form, Step 2: Set Authentication. The form has a title 'Add a non-Windows user' and a subtitle 'Step 2: Set Authentication'. It contains two main sections: 'Password' and 'Public keys (SFTP/SCP/SSH)'. The 'Password' section has a checkbox for 'Enabled' (checked), a checkbox for 'Show characters' (unchecked), a 'Random' button, a 'Password' input field (masked with asterisks), and a 'Password again' input field (masked with asterisks). The 'Public keys' section has a checkbox for 'Enabled' (unchecked), 'RSA keys' and 'DSA keys' dropdown menus (both set to 'None'), and an 'Add public key(s)' button. At the bottom are three buttons: 'Previous', 'Next', and 'Cancel'.

In step 3, the home directory for this user must be selected. By default, the new user's home directory will be created in the Home directory (see [how to add a user](#) for more details).

Add User

**Add a non-Windows user**

**Step 3: Select home folder**

Home folder:

☒ Create default home folder

/Home/JoeSmith (C:\ProgramData\Enterprise Distributed Technologies\Comp)

☐ Select non-default home folder

> Home

> Public

[Add root folder](#) [Add sub-folder](#)

Previous Next Cancel

If a different home directory is desired, use 'Select non-default home directory' to choose one of the folders listed. You may need to [create a virtual folder](#) for them.

After selecting the home directory, select the 'Next' button, and finally 'Apply' to add this new user to the server database.

Add User

**Add a non-Windows user**

**Step 4: Verify Details**

Summary:

Username:  
JoeSmith

Full name:  
(none)

Description:  
(none)

Email:  
(none)

Home folder:  
/Home/JoeSmith (C:\ProgramData\Enterprise Distributed Technologies\Comp)

< >

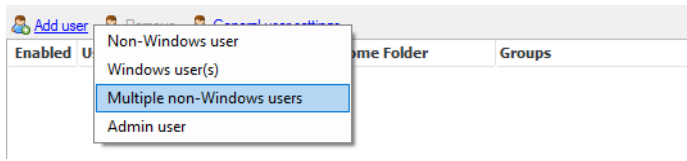
Previous Apply Cancel

For detailed steps on adding a non-Windows user, please refer to Step-by-step guide: [Add a non-Windows user](#).

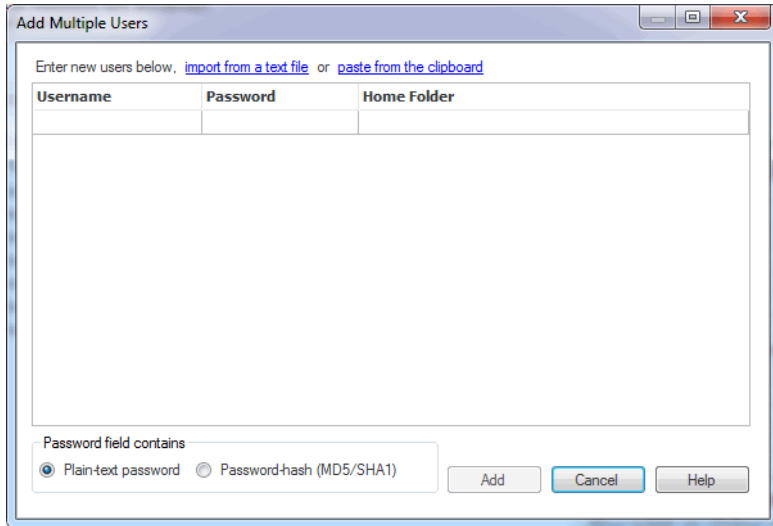
Back in the Users window, which displays the current users, note that selecting on any user displays their details in the right hand panel. Any of these details can be edited at any time (although any changes will not apply to currently logged in users).

# How to add multiple users

Often, it is desirable to add multiple users at the same time rather than one by one. This can be done for non-Windows users by selecting *Add user* and choosing the *Multiple non-Windows users* menu item.



This displays a dialog that allows multiple non-Windows users to be inputted by various means (displayed below).



Users can be explicitly typed in, imported from a text file, or pasted from the clipboard. For the latter two cases, the text file or clipboard data needs to have a user per line, and fields must be separated by commas or tabs. The first field contains the username, the second field the password, and the third field the Windows path of the home folder. Both username and password fields are mandatory. If the password is supplied as an MD5 hash rather than plaintext, then the MD5 radio button should be selected.

Finally, if the administrator has created any groups, they are displayed and can be checked so that all added users are members of the checked groups.

## How to change user properties

User properties are modified by selecting the user's row in the list of users displayed in the *Users* tab. The user properties are displayed on the right.

Add user
 Remove
 General user settings
 More columns

Enabled	Username	Email	Type	Home Folder	Groups
<input checked="" type="checkbox"/>	FredJones		Non-Windows	/Home/FredJones	users
<input checked="" type="checkbox"/>	JoeSmith		Non-Windows	/Home/JoeSmith	users

Select
 Groups
 Two-factor auth status

### AUTHENTICATORS

For site: Default Site

Authenticator	Enabled	Log in as	
Database	<input type="checkbox"/>	defaultDatabase	<a href="#">Configure...</a>
Gateway	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
LDAP	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Single sign-on / SAML	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Windows/AD Group	<input type="checkbox"/>	defaultWindows	<a href="#">Configure...</a>

### User Properties (FredJones):

Username	FredJones
User type	Non-Windows
Email	
Home folder	/Home/FredJones
Home web-app	[Site default]
Enabled	<input checked="" type="checkbox"/>
Expires	[Never]

#### Authentication

Two-factor auth enabled	<input checked="" type="checkbox"/>
Methods (SSH)	All
Password	*****
Public keys	No keys
Allowed IP addresses	All

#### File-System

Home folder is root	<input type="checkbox"/>
Encrypt stored files	<input checked="" type="checkbox"/>
Report decrypted size	<input type="checkbox"/>
Access control	Non-Windows
Read-only	<input type="checkbox"/>

#### Allowed Protocols

FTP	<input checked="" type="checkbox"/>
FTPS	<input checked="" type="checkbox"/>
SFTP	<input checked="" type="checkbox"/>
SCP	<input checked="" type="checkbox"/>
SSH terminal	<input type="checkbox"/>
SSH forwarding	<input type="checkbox"/>
HTTP	<input checked="" type="checkbox"/>
HTTPS	<input checked="" type="checkbox"/>

#### File-Sharing

Enabled	<input checked="" type="checkbox"/>
---------	-------------------------------------

#### Quotas and Limits

#### Scripting and Shells

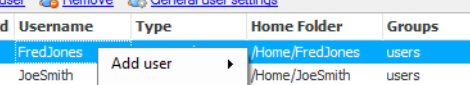
#### Personal Details

#### Timestamps

After properties have been changed *Apply Changes* must be selected to commit the changes to the server.

**Multiple users** can be selected from the list of users by using the combination of the control key and left button click. The properties panel displayed can be used to set properties for all of the selected users.

Multiple users can also be selected via the right mouse button menu item, *Select*. All users can be selected, or users from a particular group or matching a wildcard.



[Add user](#) [Remove](#) [General user settings](#)

Enabled	Username	Type	Home Folder	Groups
<input checked="" type="checkbox"/>	FredJones		/Home/FredJones	users
<input checked="" type="checkbox"/>	JoeSmith		/Home/JoeSmith	users

- Add user
- Remove user
- Select
- Groups

- Select all
- Select group
- Select by filter...
- Clear selection
- Invert selection

Groups can be quickly added/removed from this menu as well, using the *Groups* menu item. Besides, the *Manage Groups* dialog box can be accessed from here. See [How to manage groups](#) for more details.

Add user

Remove

General user settings

Enabled	Username	Type	Home Folder	Groups
<input checked="" type="checkbox"/>	FredJones	users	/Home/FredJones	users
<input checked="" type="checkbox"/>	JoeSmith	users	/Home/JoeSmith	users

Add user

Remove user

Select

Groups

Add selected users to group

Remove selected users from group

Add group...

Remove group...

Manage groups...



# How to retrieve/reset admin password

It's sometimes necessary to retrieve or reset the password for the admin user. The password can only be retrieved from inside CompleteFTP Manager (i.e. after the admin user has already authenticated), but can be reset either from inside CompleteFTP Manager or from the command-line.

## Retrieving the admin password

The method for retrieving the admin password is different in Enterprise MFT from the other editions.

### Free, Standard and Professional Editions

1. Open CompleteFTP Manager and connect to the server
2. Select the Admin panel in the left sidebar.
3. Click 'Change administrator password'
4. Check the 'Show characters' checkbox
5. The admin password is displayed

### Enterprise MFT

1. Open CompleteFTP Manager and connect to the server
2. Select the Users panel in the left sidebar
3. Click on the Options item in the left sidebar to show a context menu
4. Click on the menu item, 'Show system users/folders/sites' (unless this item already has a tick next to it)
5. Select the admin user from the list of users
6. Select the Password property
7. Click on the ellipsis (...) at the right to show the Enter Password form
8. Check the 'Show characters' checkbox
9. The admin password is displayed

## Resetting the admin password

The admin password can be reset from inside the CompleteFTP Manager or from the command-line.

### From CompleteFTP Manager

In cases where the user can log into CompleteFTP Manager, the admin password can be changed as follows:

#### Free, Standard and Professional Editions

1. Open CompleteFTP Manager and connect to the server
2. Select the Admin panel in the left sidebar.
3. Click 'Change administrator password'
4. Enter a new admin password
5. Click OK
6. Click Apply changes

#### Enterprise MFT

1. Open CompleteFTP Manager and connect to the server
2. Select the Users panel in the left sidebar
3. Click on the Options item in the left sidebar to show a context menu
4. Click on the menu item, 'Show system users/folders/sites' (unless this item already has a tick next to it)
5. Select the admin user from the list of users
6. Select the Password property
7. Click on the ellipsis (...) at the right to show the Enter Password form
8. Check the 'Show characters' checkbox
9. Enter a new admin password
10. Click OK
11. Click Apply changes

### From the command-line





If the user has lost access to CompleteFTP Manager the admin password may be reset as follows:

1. Close CompleteFTP Manager if it's running
2. Stop the CompleteFTP Service in the Windows services console
3. Open a console window with administrator privileges (Run as Administrator)
4. Enter the command: `cd "C:\Program Files\Complete FTP\Server"` (or for versions prior to 22.0.0, `cd "C:\Program Files (x86)\Complete FTP\Server"`)
5. Enter the command: `cftpconfig /p NEWPASSWORD` where NEWPASSWORD is the new admin password




6. Restart the CompleteFTP Service
7. Start CompleteFTP Manager and connect using the new password

## How to view system users

There are a number of pre-configured system users that are not normally displayed. To view these users, or to edit their properties, select 'Show system users/folders/sites' in the main applications's Options menu as shown below. Exercise caution in editing system users, as this may have unexpected side affects.

 [Add user](#)
 [Remove](#)
 [General user settings](#)
 [More columns](#)



Enabled	Username	Email	Type	Groups	Sites
<input checked="" type="checkbox"/>	admin		Non-Windows	admins	Admin
<input checked="" type="checkbox"/>	anonymous		None		Admin, Default Site
<input checked="" type="checkbox"/>	defaultDat...		Non-Windows	users	Default Site
<input checked="" type="checkbox"/>	defaultExt...		Non-Windows	users	Default Site
<input checked="" type="checkbox"/>	defaultWi...		Windows	users	Default Site
<input checked="" type="checkbox"/>	FredJones		Non-Windows	users	Default Site
<input checked="" type="checkbox"/>	JoeSmith		Non-Windows	users	Default Site
<input type="checkbox"/>	web		None		


 [Select](#)
 [Groups](#)
 [Two-factor auth status](#)


### AUTHENTICATORS


For site: Default Site


Authenticator	Enabled	Log in as	
Database	<input type="checkbox"/>	defaultDatabase	<a href="#">Configure...</a>
Gateway	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
LDAP	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Single sign-on / SAML	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
		defaultWindows	<a href="#">Configure...</a>


 Change skin
 

 Show logging window

 Show context help

 Show guide window

 **Show system users/folders/sites**

 Import Filezilla Configuration

# How to configure external database users

The database authentication feature allows the administrator to authenticate against an external data-source, rather than explicitly adding users to CompleteFTP's internal user-list. Typically, this data-source is a database table or spreadsheet containing a list of usernames, passwords or password-hashes, optionally home directories, groups, password-salts, RSA, DSA or ECDSA public keys. This is a feature of the Professional and Enterprise MFT only.

Database authentication may be set up in the General User Settings dialog, accessed from the User panel. The Database authenticator must be enabled and configured via the 'Configure' link shown below

[Add user](#) [Remove](#) [General user settings](#) [More columns](#)

Enabled	Username	Email	Type	Home Folder	Groups
---------	----------	-------	------	-------------	--------

[Select](#) [Groups](#) [Two-factor auth status](#)

### AUTHENTICATORS

For site: Default Site

Authenticator	Enabled	Log in as	
Database	<input checked="" type="checkbox"/>	defaultDatabase	<a href="#">Configure...</a>
Gateway	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
LDAP	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Single sign-on / SAML	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Windows/AD Group	<input type="checkbox"/>	defaultWindows	<a href="#">Configure...</a>

Clicking the 'Configure' link will bring up the *Database Configuration* dialog.

Database Configuration

Database type  
SQL Server Compact 4.0

Connection-string [examples](#)  
Data Source=D:\work\cftp\test\data\users.sdf

Test connection

Query for authenticating a user  
Given a username (specified by %UserName%), it must return the password or password-hash. It may also return the Windows path of the user's home-folder; a comma-separated list of groups; the password salt; an RSA public key; a DSA public key, a log-in-as user-name, and an ECDSA public key.  
SELECT Password, null, null, null, null, null, ECDSA FROM [Users] WHERE UserName='%UserName%'

Test login  
User-name  
Password  
Test login

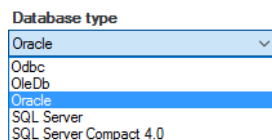
Password encoding  
☒ Plain text ☐ MD5/SHA1 hash

Query for authenticating a user  
This is the SQL query that retrieves the user's authentication data from the data-source. It must return at least the password or password-hash. It may also optionally define a Windows home-folder; a comma-separated list of groups to which the user belongs; the password 'salt'; an RSA public key, a DSA public key, a log-in-as user-name, and an ECDSA public key..

OK Cancel

In the database authentication configuration dialog, you can select the database type (ODBC, OleDb, Oracle...) you want from the 'Database type' drop-down list.

Support for new database types may be added by installing the appropriate ADO.NET provider and restarting the CompleteFTP service. Once this has been done the new database type should appear in this list. For example, support for MySQL may be added by downloading and installing the MySQL Connector/Net provider from the MySQL website and restarting the CompleteFTP service.



After a desired database type is selected, enter the connection-string in the 'Connection-string' text-box and press the 'Test connection' button to verify that it works. The website, [connectionstrings.com](http://connectionstrings.com), is an excellent resource for connection-strings. It lists various types, so remember to choose a connection-string that is appropriate to the selected database type. Also remember that the connection is made from the server, so if you are using CompleteFTP Manager on a machine other than the server then the connection-string may differ from the one you would use on your local machine.

## Fields

The SQL query retrieves the user's password/password-hash. It may also optionally define additional fields that are described below: a Windows home-folder, a list of groups to which the user belongs, a password salt, and an RSA and/or DSA public key for the user.

Field	Description
password or password-hash	This <b>must</b> be the first field returned. <u>It is highly recommended that any passwords stored in a database are hashed.</u> Even better, use a 'salt' prepended to passwords before they are hashed. See the fourth field, below.
Home folder path	If a second field is returned then it will be assumed to be the Windows path of the user's home-folder. If the home-folder doesn't already exist when the user logs in then it will be created at that time. Note that this field will not be used if the type of the home-folder of the log-in-as user is "Virtual folder".
Groups	If a third field is returned then it will be assumed to be a comma-separated list of group-names (e.g. MyGroup1,MyGroup2,MyGroup3). The user will be treated as a member of all of the groups listed as well those of which the "defaultDatabase" user is a member.
Password salt	If a fourth field is returned then it will be assumed to be the 'salt' that was prepended to the password before it was hashed (if hashing is enabled). The salt should be a random string of at least 8 characters. Salting is intended to defeat rainbow table attacks. Rainbow tables are sets of pre-calculated hashes for common passwords. If salts are not used, attackers can simply match these hashes with the hashes in your database. If salts are used, a new rainbow table must be calculated for each salt value (and these tables are computationally expensive).
RSA public key	If a fifth field is returned then it will be assumed to be the RSA public key for the user. Public key fields should be text fields in either OpenSSH or SECSH format.
DSA public key	If a sixth field is returned then it will be assumed to be the DSA public key for the user. Public key fields should be text fields in either OpenSSH or SECSH format.
Log-in-as user-name	If a seventh field is returned then it will be assumed to be the log-in-as user-name. This user-name will be used instead of that log-in-as user-name that's been configured for the authenticator.
ECDSA public key	If an eighth field is returned then it will be assumed to be the ECDSA public key for the user. Because the order of fields must be maintained for legacy reasons, this field is after the Log-in-as user field. Public key fields should be text fields in either OpenSSH or SECSH format.

If a field is required without the field preceding it, simply return the preceding field (or fields) as a null.

An SECSH format public key starts like this:

```
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "imported-openssh-key"
AAAAB3NzaC1yc2EAAAABIwAAAIEA44J6LBloMWVvhOjMHZPnmGjWw+UWB19nFEWa
```

An OpenSSH format public key starts like this:

```
ssh-dss AAAAB3NzaC1kc3MAAACBAJ3hB5SAF6mBXP1Z1RoJEZi0KSIN+NU2iGiaXZXi9CDrgVxTp6/
sc56UcYCp4qjfrZ2G3+6PWbxYso4P4YyUC+61RU5KPy4EcTJske30+aNvec/20cW7PT3TvH1+sxwGry
```

## Examples

```
SELECT PasswordHash, HomeFolder,Groups,Salt FROM [Users] WHERE UserName='%UserName%'

SELECT Password, HomeFolder,null,null,SSHPublicKeyRSA FROM [Users] WHERE UserName='%UserName%'

SELECT Password, HomeFolder,null,null,SSHPublicKeyRSA,SSHPublicKeyDSA FROM [Users] WHERE UserName='%UserName%'

SELECT Password, null, null, null, null, null, null, ECDSA FROM [Users] WHERE UserName='%UserName%'
```

## Default user settings

Users connecting via this method are subject to the settings of the "defaultDatabase" user, which may be found in the Users panel (and by selecting 'Show system users/folders/sites' in the main form's Options menu).

[Add user](#)
[Remove](#)
[General user settings](#)
[More columns](#)

Enabled	Username	Email	Type	Groups	Sites
<input checked="" type="checkbox"/>	admin		Non-Windows	admins	Admin
<input checked="" type="checkbox"/>	anonymous		None		Admin, Default Site
<input checked="" type="checkbox"/>	defaultDat...		Non-Windows	users	Default Site
<input checked="" type="checkbox"/>	defaultExt...		Non-Windows	users	Default Site
<input checked="" type="checkbox"/>	defaultWi...		Windows	users	Default Site
<input checked="" type="checkbox"/>	FredJones		Non-Windows	users	Default Site
<input checked="" type="checkbox"/>	JoeSmith		Non-Windows	users	Default Site
<input type="checkbox"/>	web		None		

[Select](#)
[Groups](#)
[Two-factor auth status](#)

### AUTHENTICATORS

For site: Default Site

Authenticator	Enabled	Log in as	
Database	<input type="checkbox"/>	defaultDatabase	<a href="#">Configure...</a>
Gateway	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
LDAP	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Single sign-on / SAML	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
		defaultWindows	<a href="#">Configure...</a>

☐ Change skin
 ☐ Show logging window
 ☒ Show context help
 ☐ Show guide window
 ☒ Show system users/folders/sites
 ☐ Import Filezilla Configuration

The administrator may, for example, enable only certain protocols for database user connections.

By default, the home folder is shown as /DatabaseUser. If the home folder is not provided by the query, the value of this folder will be used (which is a macro folder by default). If a home folder is provided by the query, the query's value will be used, but /DatabaseUser will still be displayed as the name of the user's home directory. The displayed name can be changed by modifying the name of the DatabaseUser system folder (which can be found in Folders panel if the "Show system users/folders/sites" menu item is currently checked) to use a macro, as described [here](#).

For detailed steps on configuring external database users, please refer to Step-by-step guide: [Authenticate users from a database](#).

# How to configure Windows/AD group users

The Automatic Windows authentication method allows login of any Windows or Active Directory (AD) user without having to manually add each user to CompleteFTP's internal user-list. AD users logging in by this method can use either of two user name formats:

1. **User Principal Name (UPN)** specified an Internet-style name, i.e. MyUserName@MyDomain. Note that the part of the name after the @ symbol are sometimes just the domain name and other times are the fully qualified domain name, e.g. MyUserName@MyDomain.com. Both formats can be supported by setting the Default Domain such that both are included (see [below](#))
2. **Down-Level Logon Name** specifies a more traditional name, i.e. MyDomain\MyUserName.

The feature may be enabled by checking the checkbox in the Enabled column.

In the Professional and Enterprise MFT, logins may be restricted to one or more Active Directory (AD) or local groups in the Configuration form (opened by the Configure link).

The screenshot shows the 'General user settings' tab in the CompleteFTP configuration interface. The 'AUTHENTICATORS' section is expanded, showing a table of authentication methods. The 'Windows/AD Group' authenticator is highlighted with a red box, indicating it is the selected method.

Authenticator	Enabled	Log in as	
Database	<input type="checkbox"/>	defaultDatabase	<a href="#">Configure...</a>
Gateway	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
LDAP	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Single sign-on / SAML	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
<b>Windows/AD Group</b>	<input checked="" type="checkbox"/>	defaultWindows	<a href="#">Configure...</a>

## Setting a default domain

Once Automatic Windows authentication is set up, local Windows users and domain users will be able to log in. However, to log in AD users, their domain name must be supplied so CompleteFTP knows what domain they are in. For example, MyDomain\MyUserName. However, a default Windows domain can be set so that the domain name prefix is not required.

Multiple default domains may be specified as default domains in a comma-separated list, e.g. MyDomain, MyDomain.com. If no domain name is specified then CompleteFTP tries each one until it's successful. Additionally, if any one of the listed domain names is specified then all are tried until one is successful. As mentioned in the introduction, this may be used to allow, for example, both MyUserName@MyDomain and MyUserName@MyDomain.com to work.

See [here](#) for more details of how to do this from the General User Settings.

## Configuring Automatic Windows authentication

The Automatic Windows authentication method allows login of any Windows or Active Directory user without having to manually add each user to CompleteFTP's internal user-list. Logins may be restricted to one or more Active Directory (AD) or local groups. These groups should be listed in the AD/Windows Group column. Each AD/Windows group may be associated with multiple CompleteFTP groups, a specific log-in-as user, which will override the one set for the authenticator in the Authenticators panel, and a Windows home folder.

The screenshot shows the 'Windows/AD Group Configuration' dialog box. It contains a table with the following data:

AD or Windows...	CompleteFTP Groups	Log-In-As User	Home Folder Path Override
SalesDept	Sales	defaultWindows	N:\Users\Sales
AccountsDept	Accounts	defaultWindows	N:\Users\Accounts\%UserName%

### AD/Windows Group

Name of the AD or Windows group that's to be permitted. AD groups are usually of the pattern MyDomain\MyGroup and Windows groups are specified just by MyGroup. If you're unsure about the exact name of an AD group then enter what you think it should be have the user attempt to log in. If the user entered the correct user-name and password, but was disallowed because they're not in an allowed group then a list of all the groups that they're a member of will be shown in the realtime log (and log file).

## CompleteFTP Group

Users who authenticate successfully because they're a member of the AD/Windows group in this row will be made members of the CompleteFTP groups in this field. This can be useful for controlling access to folders in the virtual file-system.

### Log-in-as User

Although a log-in-as user is defined for the Windows/AD group authenticator, as a whole, it may be overridden by the log-in-as user defined for the AD/Windows group of this row. The authenticated user will inherit the user settings of the log-in-as user, including their CompleteFTP group membership, although the set of groups of which they're a member may be added to using the CompleteFTP Groups column in this table. The home folder may also be overridden (see Home Folder).

## Home Folder Path Override

The path of the home folder of users authenticated by the AD/Windows group of this column may be overridden by setting the value of this column to that of a Windows directory. Clicking on the ellipsis button will open a server folder-browser. Macros, such as %UserName%, may be used in the path, e.g. C:\FTP\Group1\%UserName%.

Note that the way in which the value of this column is interpreted depends on the folder type of the home folder of the log-in-as user being used. If it's a Windows or Network/macro folder then it will be interpreted as a Windows path (with the optional inclusion of macros) because that's the way those folder types interpret that field, but if it's another folder type, such as a custom folder type, then it will be interpreted in the way native to that folder type.

For detailed instructions, please refer to Step-by-step guide: [Allow all users from a Windows group to log in](#).

As mentioned above, by default, users connecting via this authenticator are subject to the settings of the "defaultWindows" user. This user may be found in the Users panel if the "Show system users/folders/sites" in the main form's Options menu is checked, as shown below.

The screenshot shows the FileZilla application interface. On the left is a vertical toolbar with icons for various functions. The main window displays the 'General user settings' dialog box. This dialog has a table with columns: 'Enabled', 'Username', 'Type', 'Sites', and 'Groups'. The 'defaultWindows' user is selected, and its 'Enabled' checkbox is checked. Below the dialog box, a context menu is open, listing several options. The option 'Show system users/folders/sites' is highlighted with a red rectangle.

Enabled	Username	Type	Sites	Groups
<input checked="" type="checkbox"/>	admin	Non-Windows	Admin	admins
<input checked="" type="checkbox"/>	anonymous	None	Default Site	
<input checked="" type="checkbox"/>	defaultDatabase	Non-Windows	Default Site	users
<input checked="" type="checkbox"/>	defaultExtension	Non-Windows	Default Site	users
<input checked="" type="checkbox"/>	defaultWindows	Windows	Default Site	users
<input checked="" type="checkbox"/>	DilipGupta	Non-Windows	Default Site	users
<input checked="" type="checkbox"/>	FredBartlett	Non-Windows	Default Site	users
<input checked="" type="checkbox"/>	HeinzSchmidt	Non-Windows	Default Site	users
<input checked="" type="checkbox"/>	MaryJohnson	Non-Windows	Default Site	users
<input type="checkbox"/>	web	None		

- Change skin
- Show logging window
- Show context help
- Show guide window
- Show system users/folders/sites
- Import FileZilla Configuration

The administrator may, for example, enable only certain protocols for Automatic Windows user connections. If distinct settings are required for specific users then those users should be added explicitly as Windows users in the Users panel.



# How to configure gateway users

Gateway users are an Enterprise MFT feature of CompleteFTP. Gateway users are users who are authenticated by a different server - often residing in a different location.

Gateway users are analogous to [automatic Windows users](#) - they permit the login of any user located on a remote machine (rather than the current machine) without having to manually add each user to CompleteFTP. The remote users must be accessible via FTP, FTPS or SFTP on the remote machine.

To authenticate users on a different server, a [gateway authenticator](#) must be configured as described [here](#).

Users connecting via this method are subject to the settings of the "defaultExtension" user, which may be found in the Users panel (and by selecting "Show system users/folders/sites" in the main form's Options menu).

[Add user](#) [Remove](#) [General user settings](#) [More columns](#)

Enabled	Username	Email	Type	Sites	Groups
<input checked="" type="checkbox"/>	admin		Non-Windows	Admin	admins
<input checked="" type="checkbox"/>	anonymous		None	Admin, Default S...	
<input checked="" type="checkbox"/>	defaultDat...		Non-Windows	Default Site	users
<input checked="" type="checkbox"/>	defaultExt...		Non-Windows	Default Site	users
<input checked="" type="checkbox"/>	defaultWi...		Windows	Default Site	users
<input type="checkbox"/>	web		None		

[Select](#) [Groups](#) [Two-factor auth status](#)

**AUTHENTICATORS**

For site: Default Site

Authenticator	Enabled	Log in as	
Database	<input type="checkbox"/>	defaultDatabase	<a href="#">Configure...</a>
Gateway	<input checked="" type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
LDAP	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Single sign-on / SAML	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
		defaultWindows	<a href="#">Configure...</a>

Change skin

Show logging window

☒ Show context help

Show guide window

☒ Show system users/folders/sites

Import Filezilla Configuration

The administrator may, for example, enable only certain protocols for gateway user connections.

If gateway users are to access the remote filesystem that they are authenticated for, they need to change directory into a [gateway folder](#) or have their home directory set to the [gateway folder](#) corresponding to that server. In the latter case, the *defaultExtension* user must have its home directory set to the gateway folder. If the gateway folder set as the home directory is configured to use the gateway connection then each user authenticated on the remote server will get their own home directory on the remote directory (the same directory they would have if they logged into the remote server directory instead of via CompleteFTP).

# How to configure Single Sign-ON (SAML) users

## Background

### What is Single Sign-On?

Single Sign-On (SSO) is a concept in which multiple systems delegate authentication of users to a single authentication server. The main advantage of this is that accounts for access to many systems may be maintained in a single system. Since many different systems need to communicate, a standards-controlled protocol is required.

### What is SAML?

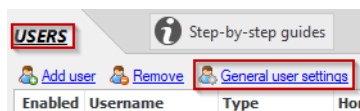
One of the most common SSO protocols is SAML (Security Assertion Markup Language). SAML involves three parties: a normal web-browser, a Service Provider (SP) and an Identity Provider (IDP). The SP is the web-server that the person using the browser wants to access, and the IDP is the server on which the person has an account.

The basic assumption of SAML is that there is a trust-relationship between the SP and the IDP. In particular, the SP trusts the IDP when the IDP says that the client is allowed to log in. This relationship is established through the exchange of SSL certificates wrapped up in packages referred to as metadata. The SP must almost always have the IDP's metadata installed before it can accept connections, so that it can verify messages from the IDP. The IDP doesn't always require the SP's metadata as it doesn't itself expose sensitive data. The exchange of metadata must happen before logins are possible.

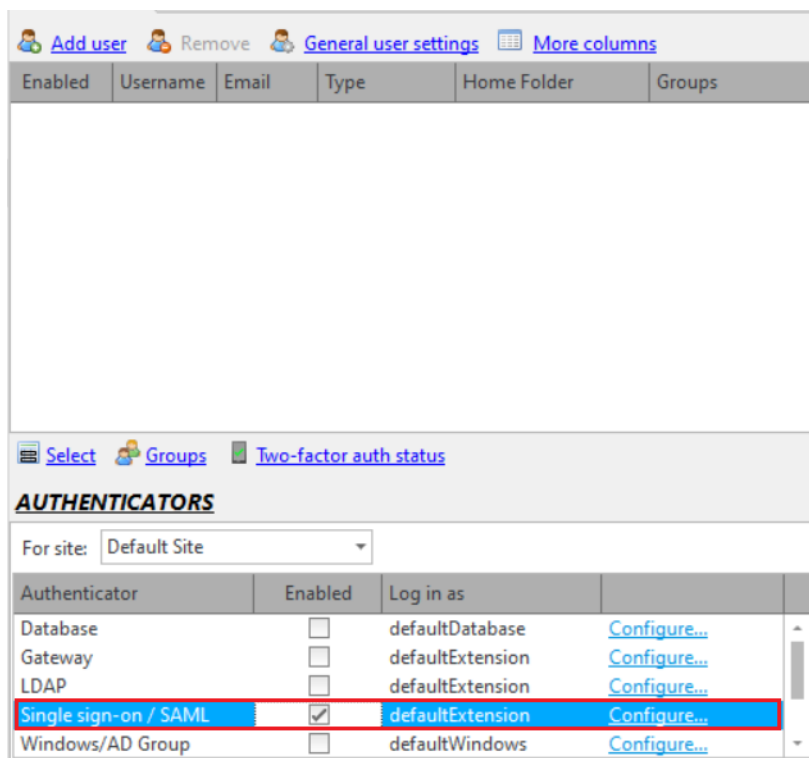
### SAML in CompleteFTP

CompleteFTP (Enterprise MFT only) can be configured as a SAML SP. This means that it can delegate authentication to a IDP. It can't itself act as an IDP.

SAML is configured in the *Other Authenticators* section of the *General User Settings* dialog, which is accessible from the *Users* panel.



The first step in configuring SAML in CompleteFTP is to enable the SAML authenticator in the *Other authenticators* section of the *General Users Settings* dialog.



After this, click the *Configure* link to open the SAML Single Sign-On Configuration dialog.

Single sign-on / SAML Configuration

Identity Providers (IDP)

Name	ID	User attribute	User prefix	User matching
OneLogin	https://app.on...		onelogin_	<input checked="" type="checkbox"/>

Add IDP Remove

☐ Validate certificates against the Windows Certificate Store

This site's URL

**Administrative contact**

First name:  Last name:  Company:

Email address:  Phone number:

Save Service Provider (SP) metadata Edit raw XML

OK Cancel

The first step in configuring SAML is to add the IDPs. For each added IDP, a button will be added to CompleteFTP's login form.

### Adding the IDP's metadata

IDPs are added using their metadata. The IDP's metadata must be either downloaded from the IDP server or sent by the IDP's administrators. Once received it must be added to the configuration by clicking the 'Add IDP' link, which allows for reading from a file or for direct download from a website.

After adding the IDP metadata the name must be entered. This name will appear on the IDP's button on the login page, as shown below:

U username

P password

☐ remember me

login

or log in via

OneLogin

In addition to the name, three other attributes may be defined for an IDP:

#### User attribute:

if the IDP doesn't send a user-name in the default format, or if a specific attribute is to be used instead of the default user-name, then the User attribute field may be set. It's the name of the 'SAML attribute' whose value holds the user-name. A SAML attribute is a name-value pair that's included in the message that the IDP sends to the SP once the authentication has succeeded. It can be something like a user-name, an ID of some sort or an e-mail address. Some attributes are defined by Uniform Resource Names (URNs). For example, urn:oid:0.9.2342.19200300.100.1.3 is an e-mail address. The attributes that are available depend on how the IDP is configured. CompleteFTP will log the available SAML attributes if the user-name could not be determined.

#### User prefix:

The user prefix is prepended to the user-name returned by the IDP. For example, if the user prefix is 'myidp-' the IDP returns 'marius' for the user-name then CompleteFTP will use the user-name, 'myidp-marius'.

#### User matching:

When enabled CompleteFTP will match the logged in user with a CompleteFTP user of the same name. If one is found then that user will be used as the **log-in-as user** for the session. For example, if the username of the user on the SAML IDP is 'marius', the prefix is 'myidp-' and there's a CompleteFTP user with the name 'myidp-marius' then that user will be used as the log-in-as user.

### Validating the IDP's certificate

As mentioned above, the IDP tells CompleteFTP that it's safe to give access to a user by means of a SAML assertion. SAML assertions are digitally signed using the IDP's private key allowing CompleteFTP to verify their origin using the IDP's certificate, which is included in the IDP's metadata. An additional level of security can be added by validating the IDP's certificate against the Windows Certificate Store.

If the checkbox labelled 'Validate certificates against the Windows Certificate Store' is ticked then the IDP's certificate will be verified against the Windows Certificate Store every time a SAML assertion is received. The main condition for a certificate to be considered valid is that the certificate of the Certificate Authority (CA) that issued it is installed in the Trusted Root Certification Authorities store for the machine in the Windows Certificate Store. This can of course only be done if the certificate was actually issued by a CA in the first place, which is often not the case.

The validation option is off by default placing the onus on the administrator who adds the IDP's metadata to **make certain that the metadata is from the intended source**. This can usually be done by downloading it over an HTTPS connection directly from the IDP website. If there's any doubt then the validation option should be enabled.

## Setting information about the SP

Some IDPs require the SP's metadata. The SP metadata is available for download from CompleteFTP at the URL that's presented when the 'Save SP metadata' link is clicked. The metadata won't be correct until (1) the site's URL has been set, (2) the administrative contact details have been filled in, and (3) the SAML configuration has been applied to the server.

The site's URL is the URL that users will be directed back to after they're successfully authenticated, so it must be a URL that resolves to root directory of the CompleteFTP server. It's important that the correct protocol is used; usually it will be HTTPS. Note that if your site is configured to use a non-standard port (80 for HTTP and 443 for HTTPS) then the site URL must include the port number (e.g. <https://mydomain.com:1443>).

## Custom configuration settings

The SAML configuration is stored in XML. A means of editing the XML directly is provided in case customizations other than those presented by the configuration form are required. This should usually not be attempted unless instructed to do so by EnterpriseDT support.

## Specifying group membership via IDP

CompleteFTP supports group claims and a custom attribute called 'group', which allows group membership to be controlled from the IDP. The values of the group claim and the group custom attribute are matched against the names of CompleteFTP groups and where there's a match the authenticated user will be a member of that group. For convenience, the value of the custom group attribute may be a comma-separated list, but since both the group claim and the custom attribute are arrays, multiple groups may also be specified as distinct elements in the array of values.

## Testing the SAML configuration

Once SAML has been configured and applied to the server, go to the CompleteFTP login page. There should be an additional button labelled with the name of the IDP. Click this button to login via the IDP. Check the real-time log in case of errors.

Note that there are many different implementations of the SAML protocol and there's a chance that your version of CompleteFTP is incompatible with some of them. Please [contact support](#) if you're unable to get an IDP to work with CompleteFTP.

# How to configure LDAP users

## Background

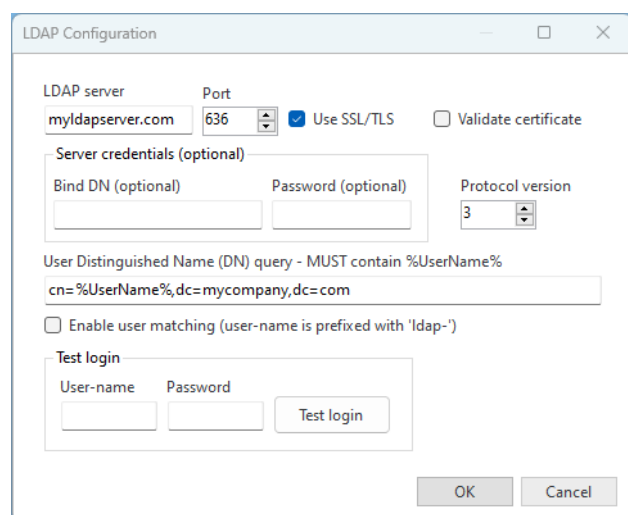
LDAP (Lightweight Directory Access Protocol) is a protocol used to access and manage directory information services over a network. These directories can store information about users, groups, and other resources, making LDAP a popular choice for authentication and directory lookups in organizations.

Configuration of the LDAP authenticator involves setting up the LDAP server details, user credentials, and query parameters to authenticate users. The form allows you to specify these settings and test the login credentials.

## Configuring LDAP in CompleteFTP

LDAP authentication can be configured in the General User Settings dialog, accessed from the Users panel. The LDAP authenticator must be enabled and configured via the 'Configure' link shown below:

Clicking the 'Configure' link will bring up the *LDAP Configuration* dialog.



## LDAP Configuration Fields

- **LDAP Server:** Enter the address of the LDAP server. This is the hostname or IP address where the LDAP service is running.
- **Port:** Enter the port number for the LDAP server. The standard port is 389 for non-SSL connections and 636 for SSL connections.
- **Use SSL/TLS:** Check this box to use SSL/TLS for secure communication with the LDAP server. This ensures that data exchanged with the LDAP server is encrypted.
- **Validate Certificate:** Check this box to validate the LDAP server's SSL certificate. This ensures that the server's certificate is trusted and prevents man-in-the-middle attacks. The certificate will be validated against the certificates in the machine's certificates store.
- **Bind DN (optional):** Enter the Distinguished Name (DN) to bind to the LDAP server, if required. This DN is used to authenticate the LDAP connection and should have the necessary permissions to perform searches in the LDAP directory. The Bind DN format is typically something like 'cn=admin,dc=example,dc=com'. Many LDAP servers allow anonymous access for authentication, in which case this field should be left blank.
- **Bind DN Password:** Enter the password for the Bind DN. This password is used to authenticate the Bind DN to the LDAP server. This field is also optional.
- **LDAP Version:** Select the LDAP protocol version. Typically, version 3 is used.
- **User DN Query:** Enter the DN query to locate the user in the LDAP directory. This query must include %UserName%, which will be replaced with the actual username during the authentication process. For example, 'cn=%UserName%,dc=example,dc=com'. This query is crucial for locating the user's DN in the directory and must be accurate to ensure successful authentication.
- **Enable User Matching:** When enabled, CompleteFTP will match the logged-in user with a CompleteFTP user whose username is prefixed with 'ldap-'. If one is found, that user will be used as the log-in-as user for the session. For example, if the user enters 'user1', user-matching will look for 'ldap-user1'.
- **Test Username:** Enter a username to test LDAP authentication. This is used to validate that the LDAP configuration is correct.
- **Test Password:** Enter the password for the test username. This is used to validate that the LDAP configuration is correct.
- **Test Login:** Once the LDAP server details and query parameters are configured, enter a username and password here to test the authentication settings. Click the 'Test login' button to perform the test.

## Valid User-Names

For security reasons, user names must not contain certain special characters that could be used in LDAP injection attacks. The following characters are prohibited: \ \* ( ) / + & | = ! > < ~ - , ; ' " ` .

## Example User DN Queries

- **Basic Query:** cn=%UserName%,dc=example,dc=com
- **With Organizational Unit:** cn=%UserName%,ou=users,dc=example,dc=com

## Testing the LDAP Configuration

Once LDAP has been configured and applied to the server, use the test login fields to validate the configuration. Enter a test username and password, then click 'Test login'. If the test is successful, the LDAP configuration is correct. If there are errors, check the LDAP server details and query parameters.

### Default User Settings

Users authenticated via LDAP are subject to the settings of the "defaultExtension" user, which may be found in the Users panel. The administrator may enable specific protocols or setting for LDAP user connections.

# How to set user quotas

CompleteFTP (Professional and Enterprise MFT only) supports the use of disk quotas and bandwidth quotas, so that users can be restricted in their use of system resources.

By default, no quotas are enforced - they have to be set explicitly for each user. This is done by going to the *Users* panel in the CompleteFTP manager, and setting the values in the *Quotas and Limits* section amongst the properties that are listed.

<b>User Properties (DilipGupta):</b>	
Username	DilipGupta
User type	Non-Windows
Home folder	/Home/DilipGupta
Enabled	<input checked="" type="checkbox"/>
Expires	[Never]
Authentication ▼	
File-System ▼	
Allowed Protocols ▼	
File-Sharing ▼	
Quotas and Limits ▲	
Disk usage (MB)	Unlimited
Download (kB/s)	Unlimited
Upload (kB/s)	Unlimited
Scripting and Shells ▼	
Personal Details ▼	
Timestamps ▼	

Disk usage is set in megabytes (MB) and is a cumulative limit across all sites.

Upload and download bandwidth limits are set in kilobytes/second (kB/s). Note that bandwidth is **not** set in kilobits/second (kbs), and that bandwidth limits apply per site (so that in Enterprise MFT, if a user is enabled on multiple sites then their total bandwidth is [number of sites] x [set bandwidth]).





# How to control user access by IP address

It's possible to configure CompleteFTP to allow specific users to connect only from specific IP addresses. This is done via the user property called 'Allowed IP addresses' in the User Properties panel (as shown below), but for this to have any meaning it's necessary to firstly deny access from other IP addresses.

<b>User Properties (DilipGupta):</b>	
Username	DilipGupta
User type	Non-Windows
Home folder	/Home/DilipGupta
Enabled	<input checked="" type="checkbox"/>
Expires	[Never]
<b>Authentication</b> ▲	
Methods (SSH)	All
Password	*****...
Public keys	No keys
Allowed IP addresses	All

## Denying access from other IP addresses

Denying access from IP addresses other than those allowed for specific users must be done via the IP filtering property in the Site Properties panel, as follows:

1. Open the Site Properties panel
2. Expand the IP Filtering and Auto-banning category
3. Select the IP filtering property
4. Click the ellipsis button on the right
5. Select 'Allow over deny'
6. If there's currently a rule that allows all connections, change this to deny all.

Once this is done, no connections will be allowed from any IP address. Note that more information on IP filters may be found in the section on [IP Filtering and Auto-Banning](#).

## Allowing specific IP addresses for specific users

Now that CompleteFTP has been configured to deny connections by default, we can go ahead and allow specific users to connect from specific IP addresses, as follows:

1. Go to the Users panel
2. Select one or more users
3. If you're in the Overview panel, right-click and select Properties to open the User Properties panel.
4. Select the property labelled 'Allowed IP addresses'.
5. Enter the IP address(es) from which the user is to be allowed to connect. Separate multiple IP addresses with commas. Domain-names may also be used.
6. Apply the changes

Note that entering allowed IP addresses simply automatically creates the appropriate user-specific IP filter rules (see [here](#)). More specifically, an 'allow rule' is created for the specified IP addresses the users on each site on which the user is enabled. Further customization of these rules may be done via the Site Properties panel.

For step by step instructions with pictures, please refer to Step-by-step guide: [Accept connections only from specific users at specific IP addresses](#).

# How to set up client certificates (FTPS)

For FTPS, clients can be forced to supply a valid client certificate (not available in the Standard Edition). This setting can be found by navigating to Settings->FTP/FTPS->Advanced FTP/FTPS Settings->Security Settings. The "Clients must supply certificates" setting is set to "Don't require valid certificate" by default, but can be changed to "Require valid certificate".

FTP / FTPS	
FTP enabled	<input checked="" type="checkbox"/>
FTPS explicit enabled (standard FTPS)	<input checked="" type="checkbox"/>
FTPS implicit enabled (legacy FTPS)	<input checked="" type="checkbox"/>
Anonymous Logins	
Advanced FTP/FTPS Settings	
Port for FTP / FTPS explicit	21
Port to use for FTPS implicit	990
Listening IP addresses	All available
Folder listing format	Unix
Character set	Default - Unicode (UTF-8)
Passive (PASV) Transfer Settings	
Security Settings	
Clients must supply certificate	Don't require valid certificate
Server certificate (also used in)	C=US, S=State, L=Localit...
Available Cipher Suites	SECURE_CIPHERS
Minimum SSL Version	TLS 1.0

If a valid certificate is required, all FTPS clients must send a certificate that may be validated against the Windows certificate store. This means that its issuer's certificate must be in the Windows 'Trusted root certificate authorities' store. This store may be managed using the Windows Administrative Tools, or by clicking '...' to the right of this security setting (this will bring up the *Certificate Management* dialog).

Certificate Management

Location

Local Machine

The certificate store for the local computer.

Store Name

Trusted root certification authorities

Specifies certificates in the Trusted Root Certification Authorities store.

Issued To	Issued By	Expiration Date
AddTrust External CA Root	AddTrust External CA Root	05/30/2020
Baltimore CyberTrust Root	Baltimore CyberTrust Root	05/13/2025
Certum CA	Certum CA	06/11/2027
COMODO RSA Certification Authority	COMODO RSA Certification Authority	01/19/2038
Deutsche Telekom Root CA 2	Deutsche Telekom Root CA 2	07/10/2019
DigiCert Assured ID Root CA	DigiCert Assured ID Root CA	11/10/2031
DigiCert Global Root CA	DigiCert Global Root CA	11/10/2031
DigiCert High Assurance EV Root CA	DigiCert High Assurance EV Root CA	11/10/2031
DST Root CA X3	DST Root CA X3	09/30/2021
Entrust Root Certification Authority	Entrust Root Certification Authority	11/28/2026
Entrust Root Certification Authority - G2	Entrust Root Certification Authority - G2	12/08/2030
Entrust Root Certification Authority - G3	Entrust Root Certification Authority - G3	07/24/2029

Import...

Remove

Close

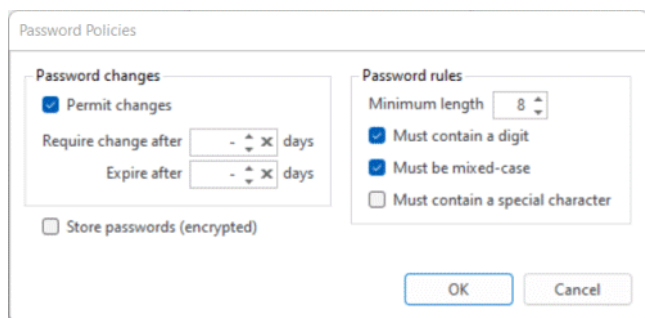
## How to make a user's home dir root

Typically users see their home directory as a path, if the defaults are used. This is the absolute virtual filesystem path, e.g. `/home/javaftp`.

It is sometimes desirable to make the user's home directory the root, i.e. `/`. This is a configuration setting for the site. See the *Users* panel's [General User Settings](#) for more details on enabling this feature.

# How to set password policies

System-wide password policies are accessible from the *Password policies* link in the General User Settings dialog box, that can be opened from the *Users* panel. The Password Policies dialog is shown below. For security reasons, it is recommended that *must have at least 8 characters*, *must be mixed-case* and *must contain a digit* are enabled.



## Password Changes and Expiry

*Permit password changes* is enabled by default in new trial installations, but are disabled by default otherwise. When enabled, non-Windows users are able to change their password, but none of the other inbuilt authentication methods (i.e. Windows, AD, gateway, external and SAML) allow password changes. *Custom authenticators* may define a way to change password.

There are multiple ways that a user can change their password:

- **HTTP/HTTPS:** users who have logged in can change their password by clicking the change password link on the account management page; users who are not logged in can use the forgot/reset password feature (see [below](#)).
- **FTP/FTPS:** once logged in users can change their password using the `SITE CPWD` command.
- **SSH (SFTP/SCP):** once logged in users can change their password using the `cpwd` command. Some SFTP and SCP clients permit opening an SSH terminal/shell.

CompleteFTP supports forced expiry of the passwords of non-Windows users. This functionality is controlled by two settings: *Require change after N days* and *Expire after N days*. Both of these are not set by default, meaning that passwords never expired.

When the *Require change after N days* setting is set, a non-Windows user will be forced to change their password if it's been more than N days since the last time they changed their password, but only if they're not logging in via FTP. FTP has no mechanism for forcing users to change their password, so they're simply shown a warning. In SFTP, SSH, HTTP or HTTPS, the user won't be given access to any files until they've changed their password.

When the *Expire after N days* is set, a non-Windows user will be blocked from accessing the server if it's been more than N days since they changed their password. This happens for all protocols.

The password expiry settings have no effect on Windows users or users who authenticate via other means, such as SAML, databases and gateways. The reason for this is that CompleteFTP doesn't control these passwords.

## Forgot/Reset Password

Users who have forgotten their passwords can use the forgot/reset password feature to set a new password. To do this they must:

1. Click the 'Forgot password?' link on the login web-page.
2. Enter their user-name or email address to have an email message containing a password reset link sent to them.
3. Look in the email inbox for the message; open it and click the link to open a password reset page.
4. Enter their new password twice.

The 'Forgot password?' link will be visible only if:

- permit password changes is enabled in CompleteFTP's password policy (see top of this page)
- **AND** a file named `customizations.js` is present in the directory, `C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Include\Custom`. If the file isn't already there then it must be created.
- **AND** the following line is in the `customizations.js` file.

```
passwordResetEnabled = true;
```

The password reset message will only be sent to the user if:

- CompleteFTP's SMTP configuration has been set (Email Settings category in the site/settings panel)
- their email address has been set
- the user is actually able to change their password via CompleteFTP (see [above](#))

Users won't be informed if the password reset message could not be sent as this would allow attackers to discover valid user-names. The reason for the failure will be written to CompleteFTP's log file, though.

## Password Storage

Store encrypted passwords is not enabled by default. Usually, passwords for non-Windows users are not stored for security reasons. Instead, a hash of the password is stored, and when users log in, the hash of the password they supply is compared with the stored hash.

If *Store encrypted passwords* is enabled, passwords for non-Windows users will be stored in encrypted form. This is less secure than using a hash, but does mean that the administrator is able to retrieve passwords.

Note that Windows users do not have passwords stored by CompleteFTP in any way. Instead, standard Windows authentication is used.

## Password Requirements

Must be mixed-case, must contain a digit, and must contain a special character are not enabled by default.

Any passwords that are less than the Minimum password length will be rejected.

# How to make user accounts expire

User accounts can be made to expire by setting an expiry date in the user properties. Once a user has expired they will no longer be able to log into any site using any protocol. The user's account remains in CompleteFTP, and the account's user-name remains reserved until the account is deleted. The account may be reactivated by clearing or modifying the expiry date.

To set, user expiry date, open the Users panel, select the user and look for the user property labelled 'Expires' and either enter the full expiry date, or click the down-arrow to open a calendar control and select the date there. The expiry date may be cleared by clicking the 'x'.

User Properties (JohnSmith):

Username	JohnSmith
User type	Non-Windows
Sites & home folders	Default Site
Home web-app	[Site default]
Enabled	<input checked="" type="checkbox"/>
Expires	5/31/2019

Authentication

Methods (SSH)

Password

Public keys

Allowed IP addresses

File-System

Home folder is root

Encrypt stored file

Access control

Read-only

Allowed Protocols

Tuesday, May 7, 2019

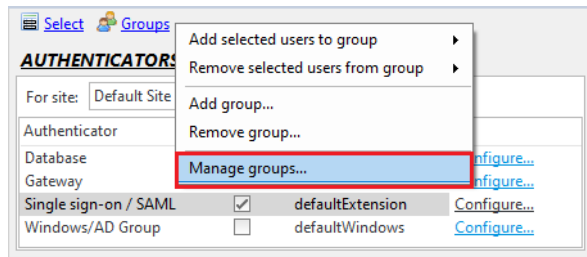
May 2019

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

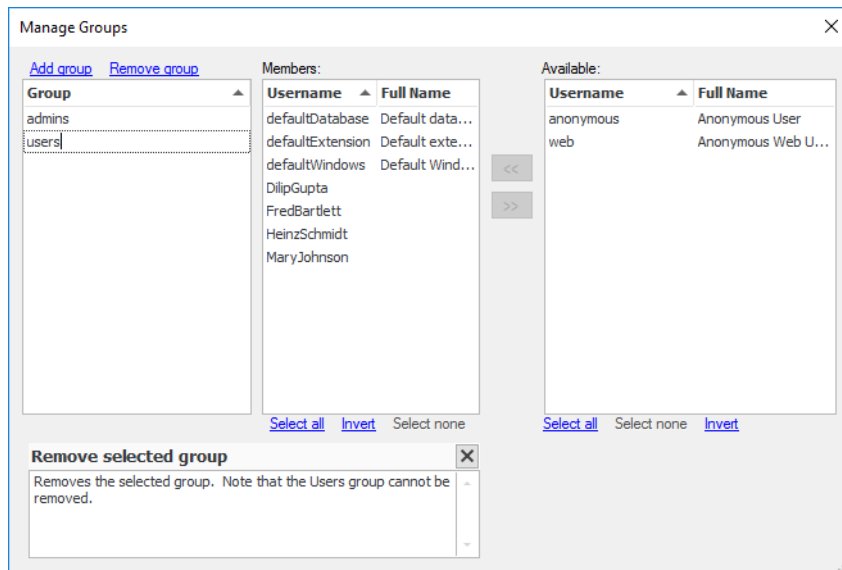
# How to manage groups

Groups may be used to simplify management of access of users to system resources, such as files or custom commands.

Groups of users can be managed via the *Manage Groups* dialog box which can be opened by selecting the *Groups* link at the bottom of the Users panel and then selecting *Manage groups* from the menu.



The *Manage Groups* dialog box is used for adding and removing users from groups.

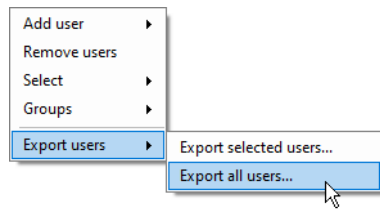


This dialog box allows you to create new groups and add users to those groups. Once groups have been created as desired, permissions may be assigned to those groups.

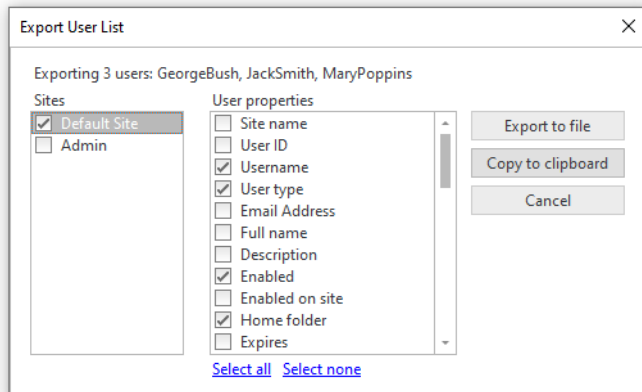
Group permissions for a given folder may be set by going to the Folders panel; selecting the folder; setting the Group of the folder; clicking on the Permission property and checking the checkboxes in the group section of the permission editor.

# How to Export Users

A list of users can be exported by right-clicking on the user list in the Users panel and selecting 'Export users' and then either 'Export selected users' or 'Export all users'.



A dialog is shown where the desired export can be specified. Since some user properties have different values for different sites, it's possible to select the sites for which data should be exported. Users have many properties, so only those that are selected will be exported.



The list can be exported to a file or copied to the clipboard. When copied to the clipboard, fields will be tab-delimited, which allows pasting into spreadsheets. When exported to a file the format can be either CSV or tab-delimited depending on the extension of the file selected (.csv or .tab, respectively). The default is .csv.



# File sharing

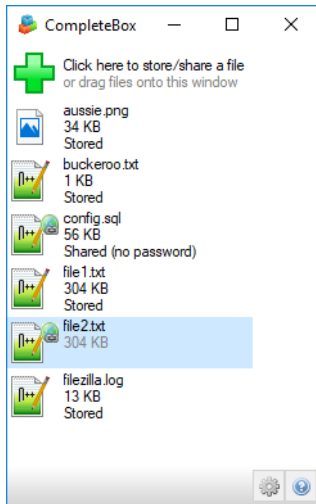
*CompleteBox* is a application installed on user's computers that allows them to easily store and share their files on a server machine running CompleteFTP Professional or Enterprise.

This is known as *collaborative file sharing and storing*, and it requires CompleteFTP Professional or Enterprise MFT.

To store or share files, file sharing must first be **enabled in CompleteFTP**. Users **install CompleteBox** and use it to **share their local files**. Files can be stored on the server for private use, or shared with others.

The Professional Edition includes 50 CompleteBox licenses and Enterprise MFT includes unlimited CompleteBox licenses.

The CompleteBox desktop application is shown below:

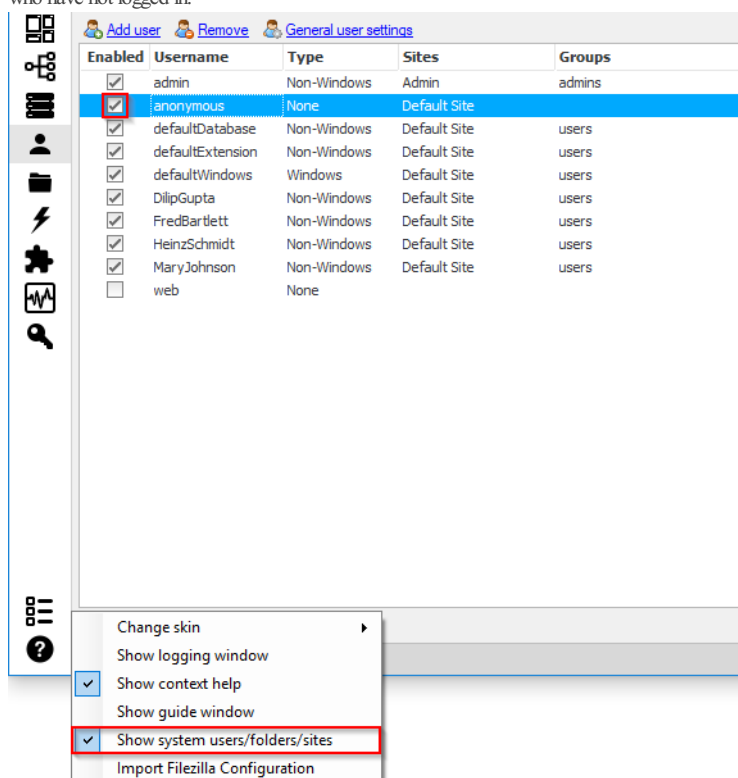


# How to enable file sharing

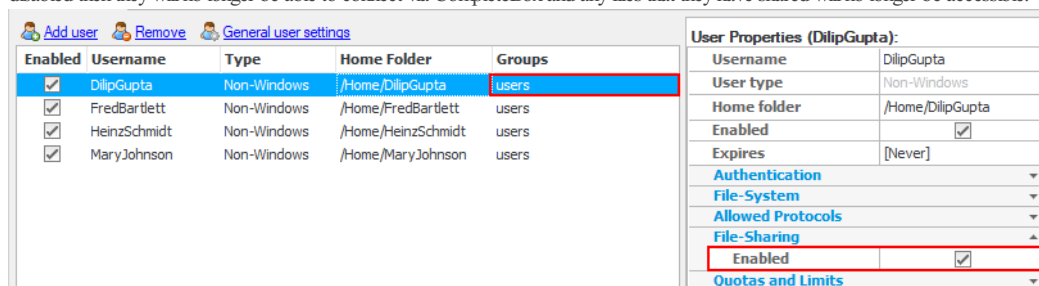
CompleteBox enables *storing or sharing* of local files on a CompleteFTP server instance, which must be either the Professional Edition or Enterprise MFT.

There are a few settings that must be enabled in CompleteFTP for CompleteBox users to begin using CompleteBox to share or store their files. If file-sharing is enabled during installation, all the settings below should already be enabled. The settings below should be checked if there are any problems getting file-sharing to work.

1. File sharing must be enabled in CompleteFTP. The CompleteFTP site must have sharing enabled, as described [here](#). If it's turned off then users will no longer be able to connect via CompleteBox and shared files will no longer be available via HTTP/S.
2. Public HTTP access must be enabled, as described [here](#). It is enabled by default. If it is disabled then files that have been shared via a link will not be accessible to users who have not logged in.
3. HTTPS must be enabled, as described [here](#). It is enabled by default. If it's disabled then shared files will not be accessible via a link.
4. The anonymous system user must be enabled, via the Users tab. Use 'Show system users/folders/sites' in the Options menu at the bottom left of the window to display it. It is enabled by default. As with public HTTP access above, if it is disabled then files that have been shared via a link will not be accessible to users who have not logged in.



5. If a user is to share files, then their account must be enabled for sharing and they must be a member of the *users* group (as shown below). If a user's sharing is disabled then they will no longer be able to connect via CompleteBox and any files that they have shared will no longer be accessible.



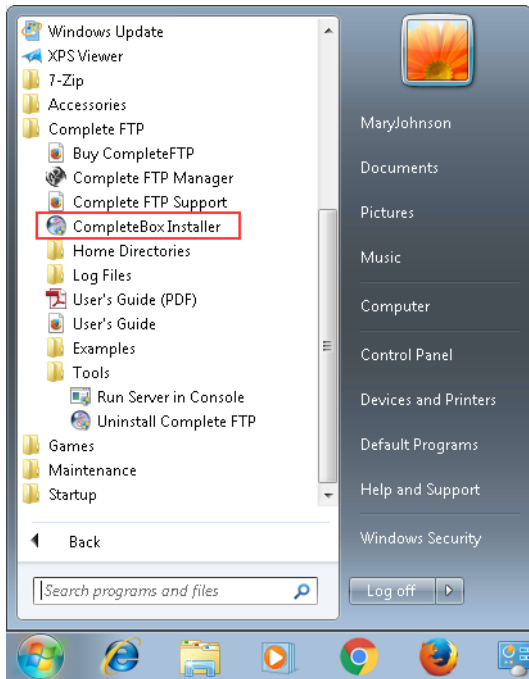
Finally, sufficient CompleteBox user licenses must be enabled in CompleteFTP so that every user who's file sharing has a user license. Each person requires a user license. The license count is based on the edition of CompleteFTP being used (from 10.0 onwards). The Professional Edition includes 50 user licenses, while Enterprise MFT includes unlimited licenses.

# How to install CompleteBox

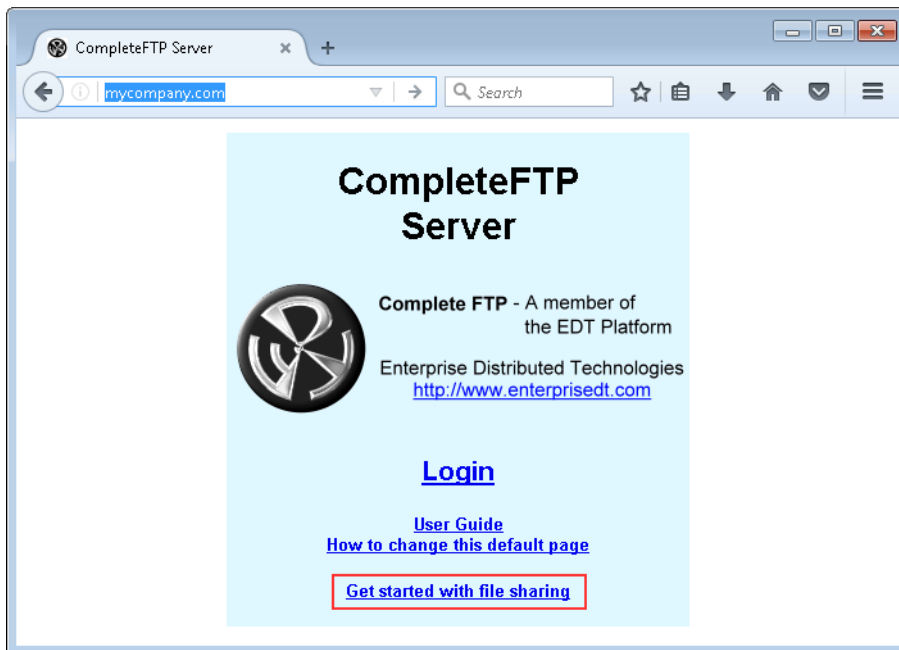
To share a local file, the *CompleteBox* client must be installed on the local machine that files are being shared from.

## Getting the installer

The CompleteBox installer is shipped with **CompleteFTP server**. After CompleteFTP has been installed, the CompleteBox installer can be found from the CompleteFTP program group in the Windows Start Menu.



The installer can also be downloaded from your company's CompleteFTP website. On the default index page of the server, a link "Get started with file sharing" provides installer links.



The CompleteBox installer can freely redistributed, although it may only be used with a CompleteFTP server.

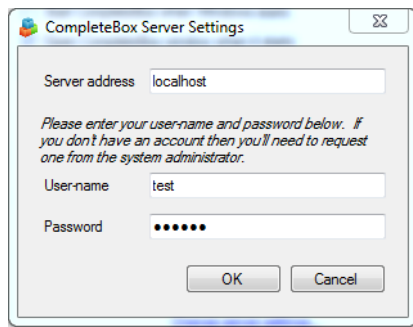
## Running the installer

The installer is an executable that requires administrator permissions to run.

If you are upgrading, it is important to upgrade CompleteFTP first. Older versions of CompleteBox will work with the latest version of CompleteFTP (as well as previous versions of course), but the latest version of CompleteBox **requires** the latest version of CompleteFTP.

Once installed, *CompleteBox* runs as a Windows system tray application. It can be opened by clicking on its icon in the Windows system tray.

An account on the file sharing server is required, and the credentials must be entered when prompted. The file sharing server must have sufficient user licenses free to permit an additional file sharer. The server settings dialog is shown below:



CompleteBox Server Settings

Server address: localhost

Please enter your username and password below. If you don't have an account then you'll need to request one from the system administrator.

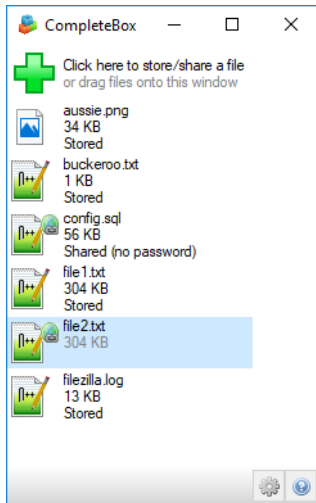
User-name: test

Password: ●●●●●●

OK Cancel

Once installed, files can now be *stored or shared*.

# File sharing vs file storing



*CompleteBox* is an application installed on user's computers that allows them to easily **store** and **share** their files on a CompleteFTP server machine set up for this purpose.

*Sharing* a file means that the file is uploaded to the server and a public URL is generated by which others can download the shared file. You share a file so others can have access to it. A shared file is marked as "Shared", and has a link icon superimposed on its file icon.

*Storing* a file means that the file is uploaded to the server for the user's private use. No URL is available, and no-one other than the user can access the file. A stored file is marked as "Stored".

## Why store and not share?

There are two reasons why a CompleteBox user might use file storage without sharing.

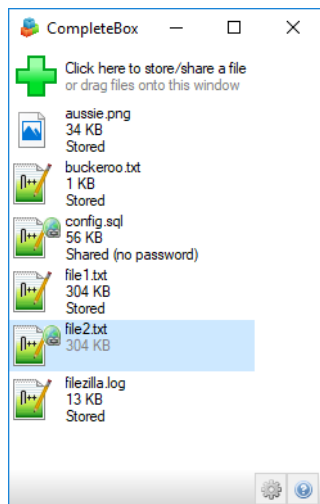
**Backup.** The first reason is that it is a useful way to quickly back up files onto a server. Select and upload, and your local files are now safely stored in case of a local machine failure.

**File synchronization.** The second reason is for private synchronization of files between machines. If you work in two locations (e.g. a work location and a home location), you can upload from one machine, and later download onto another. CompleteBox file synchronization will help you to keep files on both machines identical.

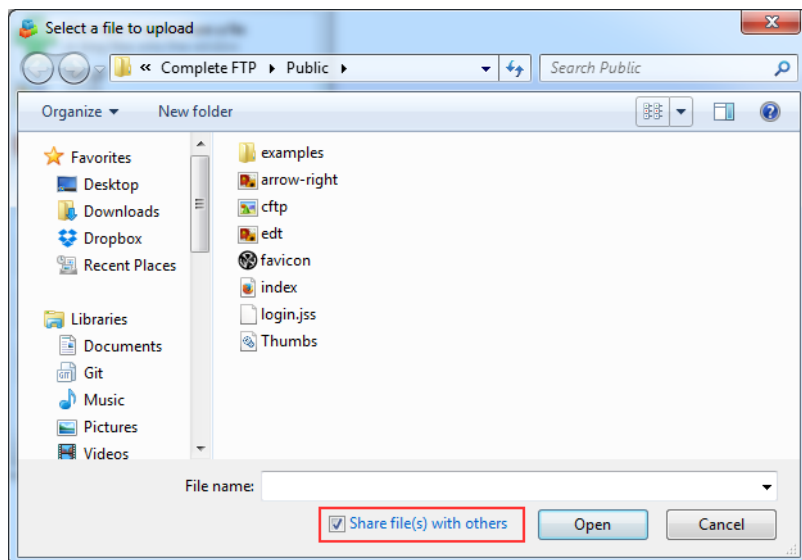
# How to share files with CompleteBox

Sharing a file with *CompleteBox* means that the selected file is uploaded to the server and a public URL is generated by which others can download the shared file. Provided someone is given the URL, the file is publicly available.

To share a local file, the *CompleteBox* client must be installed on the local machine that files are being shared from, as described [here](#). The main dialog is displayed by selecting the CompleteBox icon in the Windows system tray, shown below.

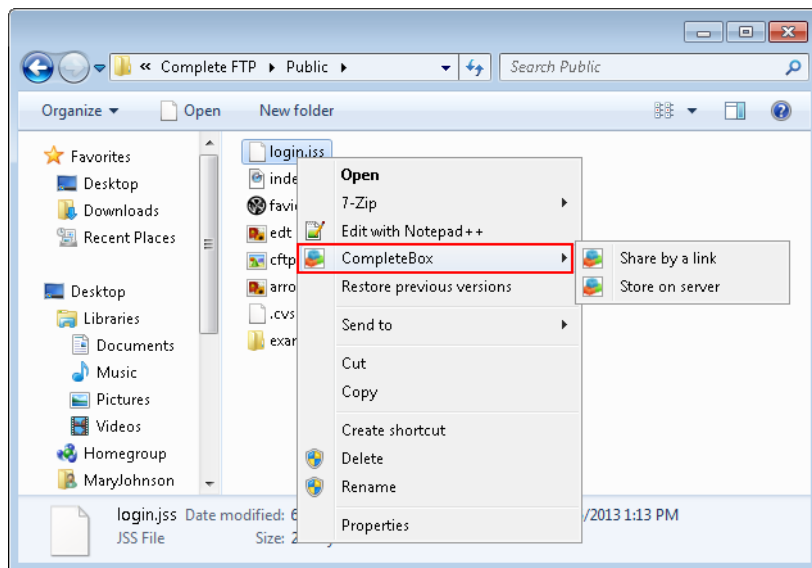


Click on the green cross to display a file open dialog box for selecting files to share, or drag and drop files onto the dialog. Make sure the "Share file(s) with others" checkbox at the base of the dialog is ticked - if it is not, the files are stored but not shared. Sharing is the default.



Multiple files can be selected in the file open dialog by holding down the control key and selecting each file. Once selected, the files will be shared on the file sharing server via a secure connection. A unique link (i.e. a URL) is generated that can be emailed to the end-user of the file. Links are configured to expire in 30 days by default. If a single file is shared, the share URL is sent to the clipboard.

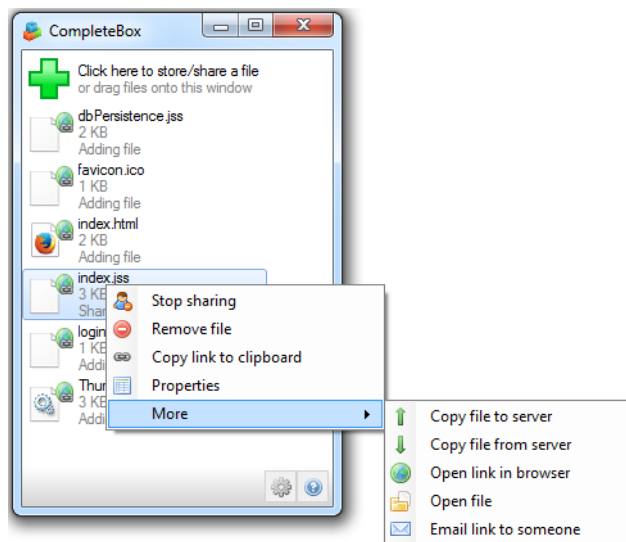
Note also that files can be drag-and-dropped or copy-and-pasted onto the main CompleteBox window, and Windows Explorer has a menu item "CompleteBox" which will store or share the selected files.



Right clicking on a shared file in the list will display the right-click menu shown below, which allows the sharing to be stopped, the file removed, the link copied to the clipboard, and the file properties to be displayed. Multiple files can be selected by holding down the control key and selecting each file. Control-A will select all files listed.

Additional options allow the forced copying of the local file to the server, or from the server to local file. This is in case you know the local or remote file to be the authoritative one, and you want the other copied over, irrespective of which is the most recent file. You can also open the link in a browser, open the local file, or email the URL to someone.

Finally, Control-C can be used to copy the selected files to the clipboard. Select a target directory in Windows Explorer, and Control-V will paste copies of these files into the directory.

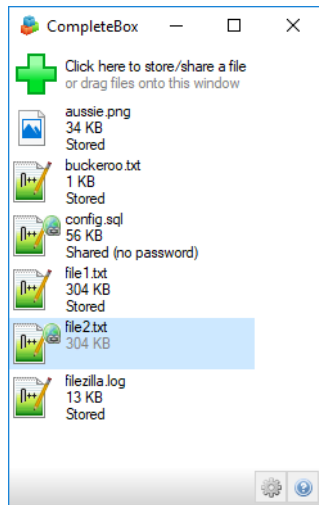


For step by step instructions on how to share a file with CompleteBox, please refer to Step-by-step guide: [Share a file using CompleteBox](#).

# How to store files with CompleteBox

Storing a file with *CompleteBox* means that the selected file is uploaded to the server, but no URL is generated and the file is not available to anyone except its owner. Typically this is done for backup, or for a user to share their own files with themselves on two different computers.

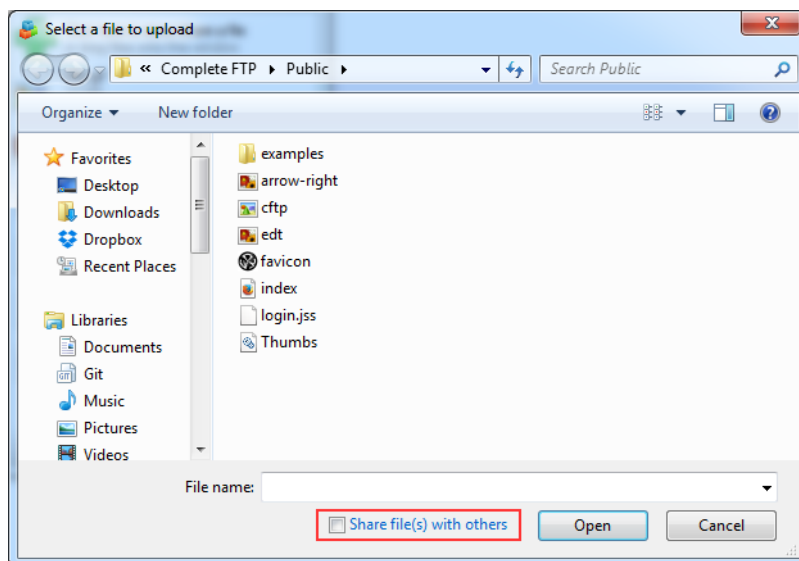
To store a local file, the *CompleteBox* client must be installed on the local machine that files are being shared from, as described [here](#). The main dialog is displayed by selecting the CompleteBox icon in the Windows system tray, shown below.



Click on the green cross to display a file open dialog box for selecting files to share, or drag and drop files onto the dialog. Make sure the "Share file(s) with others" checkbox at the base of the dialog is **unticked** as below (otherwise the file will be shared).

Multiple files can be selected in the file open dialog by holding down the control key and selecting each file. Once selected, the files will be stored on the file sharing server via a secure connection.

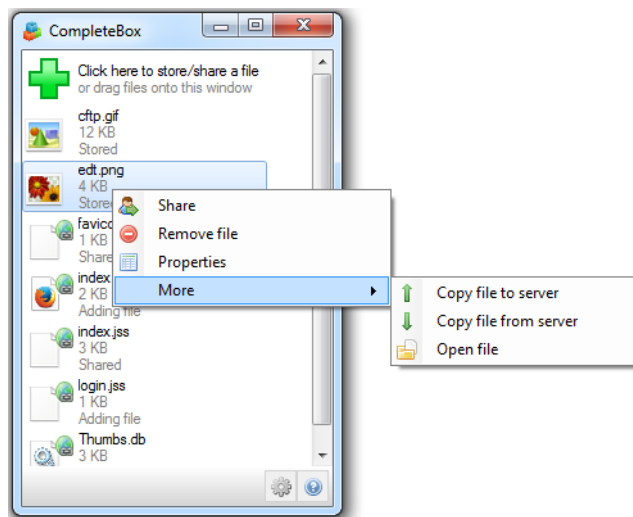
Note also that files can be drag and dropped onto the main CompleteBox window, and Windows Explorer has a menu item "Store in CompleteBox" which will store or share the selected files.



Right clicking on a stored file in the list will display the right-click menu shown below, which allows the stored file to be shared, the file to be removed, and the file properties to be displayed. Multiple files can be selected by holding down the control key and selecting each file. Control-A will select all files listed.

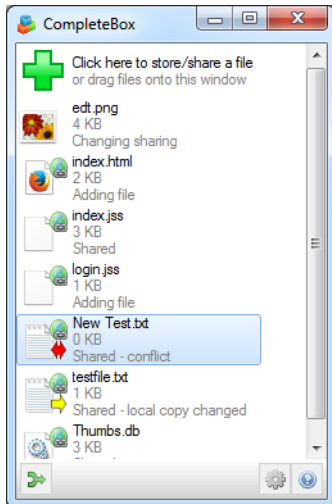
Additional options allow the forced copying of the local file to the server, or from the server to local file. This is in case you know the local or remote file to be the authoritative one, and you want the other copied over, irrespective of which is the most recent file. You can also open the local file.





# File synchronization

Once a local file has been **shared or stored** on the server, there is the possibility that either copy will become out of date. This means the file needs to be *synchronized*, i.e. the local copy and the server copy need to be made identical.



This can happen in two ways. The local file could be modified by the user, or the remote file could be modified. The remote file could be modified if the same user uses CompleteBox on a different machine to upload a change. If this happens, CompleteBox will show that the files are different by displaying a "needs synchronizing" message under the filename, and changing the icon for the file.

A red double arrow means both the local and remote file have changed since the last synchronization. In this case, you will be prompted to select which file should be the authoritative one.

A yellow arrow means either the local or the remote file have changed since the last synchronization. When you synchronize, the most recently changed file will be copied over the older file. The arrow indicates the direction of the copy. Pointing to the right means the local file will be copied over the server file, and pointing to the left means the server file will be copied over the local file.

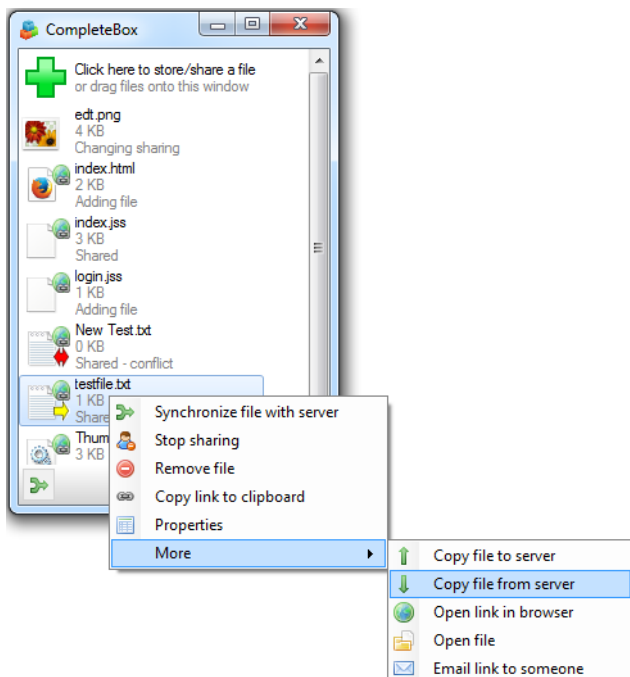
## How to synchronize files

To synchronize the local and server files, right-click on the filename and use the *Synchronize file with server* menu item, as shown below. This will ensure the latest copy is the local and the remote file - it works via the timestamp.

You can also synchronize **all** out-of-date files via the green synchronize button that appears on the bottom left of the CompleteBox file listing, shown on the left.

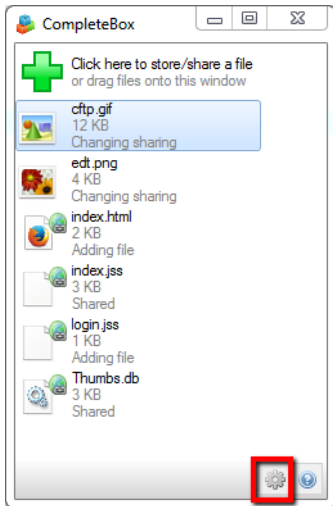
## Manually overriding synchronization

Occasionally, you may find that you want to overwrite local changes that you have not yet synchronized to the server. For example, you may have made some local changes, and wish to discard them. This can be done by right-clicking on the filename and choosing *More -> Copy file from server*. This copies the server file over the local file, erasing the local changes. The *Copy file to server* menu item does the opposite, forcing the local file to be copied to the server, overwriting any changes in the server file.

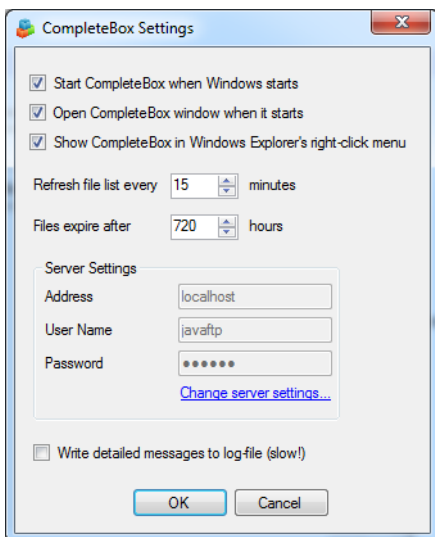


# How to configure CompleteBox

*CompleteBox's* configuration dialog is found by selecting the cog icon on the main dialog that displays the list of shares, shown on the left.



Clicking the cog icon shows the following dialog:



There are various options controlling if CompleteBox is started with Windows, as well as enabling or disabling the Windows Explorer integration.

The CompleteBox client automatically refreshes its shared files listing every 10 seconds from the server while the window is active, and this can be altered. The default share expiry time of 720 hours (30 days) can also be changed here.

The credentials can also be changed in this dialog - both the username and password, and the sharing server that is being used.

Finally, if any problems are experienced that require contacting support, detailed logging can be switched on, and the zipped log file emailed to support. Note that this slows down file transfers considerably, and should only be used temporarily. The log file can be found at

C:\Users\username\AppData\Local\EnterpriseDT\CompleteBox\CompleteBox.log

# CompleteBox Software License Agreement

We **Enterprise Distributed Technologies Ltd** grant you license to use this Software on the following terms:

## 1. License and support

1. This Software is licensed and not sold to you. Your use of this Software is governed solely by the terms of this License. This License is personal to you.
2. This Software is owned by us, save for certain elements of it which are owned by The KPD-Team; Chew Keong Tan; Routrek Networks, Inc; ComponentAce; and The Mentalis.org Team. This Software is protected by copyright law, and the owners reserve ownership of all Intellectual Property Rights in it, and all rights other than those expressly granted by this Agreement.
3. Technical support for CompleteBox must be obtained via the server administrator of the file sharing server that is being connected to by CompleteBox.
4. Where use is permitted by this License, such use includes
  - i. the making of no more than a reasonable number of backup copies of the Software; and
  - ii. the right to utilise and make prints for your own personal use (but not otherwise copy) any instructional and/or operational manuals relating to the Software.

## 2. Client License

The CompleteBox client application may be installed without charge on any machine, without restriction.

1. CompleteBox may only be used to connect to a licensed server instance of CompleteFTP.
2. You may not modify the Software. You may not disassemble, decompile, or reverse engineer the Software, or otherwise attempt to discover the source code of the Software.

## 3. Warranties, and Limitation of Liability:

1. You acknowledge that
  - i. you have had the opportunity to evaluate the Software without charge, and that you have satisfied yourself that it meets your requirements in all material respects
  - ii. it is not technically practicable to guarantee software to be error-free, and you agree that if any such errors are found to exist they shall not constitute a breach of this Agreement
  - iii. the Software is provided for use by you as an expert, and that where you decide to incorporate the Software in an application created by you, it is solely your responsibility to carry out all appropriate testing and to determine its suitability for your intended purpose. You will therefore indemnify us against any and all claims that may be brought by a third party in relation to any applications distributed by you which incorporate the Software, whether or not the terms of the License you have bought permit such distribution.
2. THE SOFTWARE IS NOT WARRANTED TO BE FAULT-TOLERANT, AND IS NOT INTENDED FOR THE DESIGN, CONSTRUCTION, MAINTENANCE, OPERATION, CONTROL, OR ANY OTHER USE IN CONNECTION WITH HIGH RISK SYSTEMS. WE SPECIFICALLY DISCLAIM ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR PURPOSE IN CONNECTION WITH HIGH RISK SYSTEMS. 'High Risk Systems' means systems in environments requiring fail-safe performance (such as nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems), in which the failure of the software could lead directly to death, personal injury, or severe physical or environmental damage. You agree that you will not use the software for any purpose in connection with High Risk Systems.
3. THIS SOFTWARE IS PROVIDED "AS IS", AND TO THE EXTENT PERMITTED BY THE APPLICABLE LAW, WE DISCLAIM ALL WARRANTIES WITH RESPECT TO THE SOFTWARE, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.
4. ALTHOUGH WE DO NOT WARRANT THAT THE SOFTWARE SUPPLIED HEREUNDER SHALL BE FREE FROM ALL KNOWN VIRUSES WE HAVE USED COMMERCIALY REASONABLE EFFORTS TO CHECK FOR THE MOST COMMONLY KNOWN VIRUSES. YOU ARE NEVERTHELESS SOLELY RESPONSIBLE FOR VIRUS SCANNING THE SOFTWARE.
5. WE SHALL NOT BE LIABLE TO YOU OR TO ANYONE ELSE FOR ANY LOSS OR DAMAGE WHATSOEVER OR HOWSOEVER CAUSED ARISING DIRECTLY OR INDIRECTLY IN CONNECTION WITH THIS LICENSE, THE SOFTWARE, ITS USE OR OTHERWISE, EXCEPT TO THE EXTENT THAT SUCH LIABILITY MAY NOT BE LAWFULLY EXCLUDED UNDER THE APPLICABLE LAW.
6. NOTWITHSTANDING THE GENERALITY OF THE ABOVE, IN NO EVENT WILL WE BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL LOSS OR DAMAGE WHICH MAY ARISE IN RESPECT OF THE SOFTWARE, ITS USE, OR IN RESPECT OF OTHER EQUIPMENT OR PROPERTY, OR FOR LOSS OF PROFIT, BUSINESS, REVENUE, GOODWILL OR ANTICIPATED SAVINGS, EVEN IF WE HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSS.
7. IF ANY EXCLUSION OR LIMITATION CONTAINED IN THIS LICENSE SHALL BE HELD TO BE INVALID FOR ANY REASON AND WE BECOME LIABLE FOR LOSS OR DAMAGE THAT MAY LAWFULLY BE LIMITED, SUCH LIABILITY SHALL BE LIMITED TO THE LICENSE FEE PAID BY YOU FOR THE SOFTWARE.
8. We do not exclude liability for death or personal injury to the extent only that it arises as a result of our negligence or that of our employees, agents or authorized representatives.
9. So far as any parts of this Software which were not created by us are concerned, THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ANY PERSON WHO HAS CONTRIBUTED TO OR IS THE OWNER OF ANY PART OF THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. This provision shall operate for the benefit of any and all such persons.

## 4. Miscellaneous

1. You acknowledge that these terms supersede all prior agreements, and are complete and exclusive. No oral or written information given by us or on our behalf shall create a warranty or collateral contract, or in any way increase the scope of this warranty in any way, and you may not rely on any such advice.
2. If any provision in this Agreement shall be determined to be invalid, such provision shall be deemed omitted; the remainder of this Agreement shall stand.
3. This License shall be governed by the laws of Australia.

# Clustering: Introduction

## 1. Overview of Clustering

Clustering in CompleteFTP refers to the ability to run multiple instances of CompleteFTP servers across different machines, working together to provide a unified service. This setup allows for load balancing, failover, and centralized administration capabilities, ensuring that client connections are distributed across servers and that the service remains available even if one server fails. While CompleteFTP does not provide native clustering, when combined with a hardware or software load balancer (e.g., Microsoft NLB), load balancing and failover can be achieved.

### Benefits of Clustering:

- **Load Balancing:** By distributing client connections across multiple servers, clustering helps to manage the load effectively. This ensures that no single server becomes a bottleneck, enhancing performance and scalability.
- **High Availability and Failover:** Clustering provides redundancy. If one server fails, the load balancer can redirect clients to other operational servers in the cluster, ensuring continuous service availability.
- **Centralized Administration:** The configurations of multiple CompleteFTP servers may be done via a single server (the primary), and changes are automatically distributed to the other servers in the cluster (the secondaries).

### CompleteFTP's Role in Clustering:

- **Configuration Synchronization:** CompleteFTP manages the synchronization of server configurations across the cluster. A primary server is designated to control all configuration changes, which are instantly propagated to all secondary servers in the cluster. This makes the active/active cluster setup ideal, as an active/passive setup would prevent configuration propagation to dormant servers.
- **No File Synchronization:** CompleteFTP does **not** handle the synchronization of user files across the servers. External solutions such as Windows DFS (Distributed File System) or network storage are required to synchronize files and directories between servers.

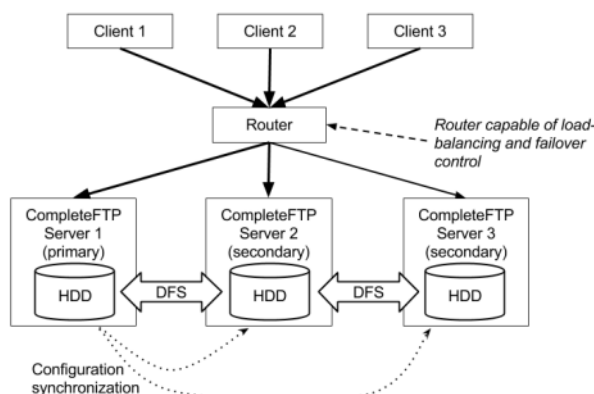
## 2. Clustering Architecture

CompleteFTP clustering can be configured using different architectures depending on your organization's needs. Below are the main options.

### 2.1. Active/Active Clustering with Local Storage

In this architecture, each server has its own local storage. File synchronization is managed externally through tools like Windows DFS, which replicates files and directories across all servers in near real-time. This ensures that users can access their files regardless of which server they connect to.

#### Architecture Diagram:



#### Explanation:

- The router directs incoming client requests to one of the servers in the cluster.
- Each server is running CompleteFTP and is configured to respond to the same requests.
- The primary server is responsible for distributing configuration changes to the secondary servers.
- File synchronization is handled by Windows DFS, which ensures that user files are consistent across all servers.

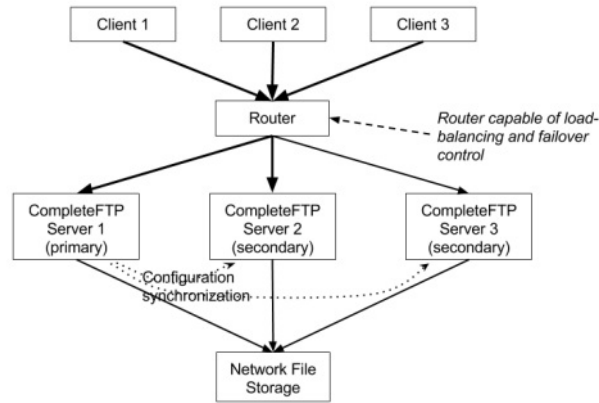
**Pros:** This architecture offers redundancy and ensures that user files are available across all servers.

**Cons:** There is added complexity due to the external file synchronization solution.

### 2.2. Active/Active Clustering with Network Storage

In this architecture, all servers in the cluster access a shared network storage system, such as NAS or SAN, eliminating the need for file synchronization. This simplifies file management as all servers access the same centralized storage for user files.

#### Architecture Diagram:



#### Explanation:

- All servers in the cluster connect to a single shared storage, ensuring data consistency without the need for DFS.
- The primary server manages configuration changes, which are synchronized across the cluster.

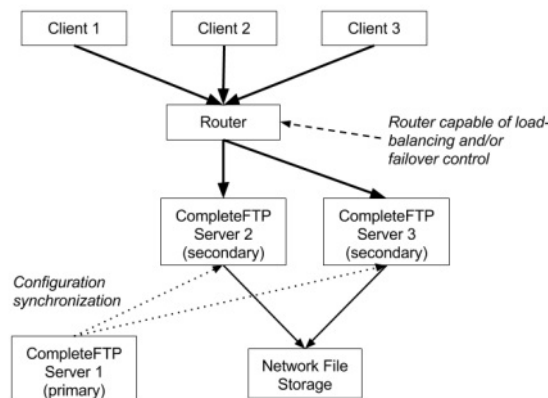
**Pros:** Simplifies file management by removing the need for complex file synchronization tools.

**Cons:** The shared storage system can become a single point of failure unless it has built-in redundancy.

### 2.3. Load Reduction on Primary Server

In high-load environments, it may be necessary to reduce the load on the primary server. In this architecture, the load balancer directs most client traffic to secondary servers, allowing the primary server to focus on configuration management and synchronization tasks.

#### Architecture Diagram:



#### Explanation:

- Client traffic is primarily directed to secondary servers, reducing the load on the primary server.
- The primary server remains the central point for configuration management and synchronization.

**Pros:** Improves performance by offloading client traffic from the primary server.

**Cons:** More complex load balancer configuration is required.

## 3. Primary and Secondary Servers

**Primary Server:** The primary server is the central point for configuration management within the CompleteFTP cluster. All configuration changes are made through the CompleteFTP Manager on the primary server. It is responsible for synchronizing these changes to all secondary servers in near-real-time. The primary server also acts as the point of contact for new secondary servers joining the cluster, ensuring they are updated with the latest configurations.

**Secondary Servers:** Secondary servers handle client connections and apply the configuration changes received from the primary server. Although secondary servers are capable of handling client traffic, they depend on the primary server for all configuration updates.

In some architectures, to reduce the load on the primary server, most client traffic can be directed to secondary servers while the primary focuses on administrative tasks. This architecture still ensures that the primary retains control over all configuration synchronization.

## 4. Load Balancers

Load balancers are critical in clustering setups as they distribute incoming client connections across the servers in the cluster, managing traffic and providing failover. They help ensure that

no single server becomes overloaded and that the cluster remains available even if a server fails.

**Role:** The load balancer can be configured to manage traffic distribution and failover, redirecting client connections to operational servers if a server goes down. In scenarios where HTTP/S protocols are in use, load balancers should be configured for session stickiness to ensure that client sessions remain with the same server.

**Types:** Hardware load balancers or software solutions such as Microsoft Network Load Balancing (NLB) can be used to manage traffic across a CompleteFTP cluster.

## 5. Protocol-Specific Considerations

Some protocols require specific configurations in a clustered environment. CompleteFTP supports multiple protocols, and different strategies must be applied for high availability and failover based on the protocol.

### 5.1 HTTP/S Session Stickiness

CompleteFTP supports HTTP and HTTPS file transfers, but each HTTP/S session is local to the specific server a client connects to. The load balancer must ensure that all requests from a client during a session are directed to the same server. This can be achieved by configuring session stickiness, typically using the client's IP address or cookies. A future release of CompleteFTP may include HTTP/S session cache replication across the cluster.

### 5.2 FTP, SFTP, FTPS Protocols

For FTP, SFTP, and FTPS, all requests happen within a single connection. Once a connection is established, it remains on the same server for the duration of the session. Load balancing is applied only at the connection level, ensuring that new connections are distributed evenly across servers. While secondary servers handle client connections, all servers must remain synchronized with the primary server for configuration data to ensure a seamless experience across sessions.

## 6. Prerequisites and Requirements

Before setting up clustering in CompleteFTP, it is important to understand the prerequisites and requirements to ensure a successful deployment.

**Licensing:** Clustering is available in the *CompleteFTP Enterprise MFT* edition. Ensure that you have the appropriate license to use clustering features.

**Supported Protocols:** Clustering supports all protocols available in CompleteFTP, including SFTP, FTPS, FTP, and HTTPS.

### System Requirements:

- **Hardware:** CompleteFTP servers require adequate hardware resources to handle expected client connections and data transfer loads. Please refer to the *CompleteFTP System Requirements* for detailed hardware specifications.
- **Software:** CompleteFTP runs on Windows operating systems. Ensure that all servers in the cluster are running a supported version of Windows Server.

### Network Considerations:

- **Firewall Settings:** Ensure that firewall rules allow communication between the primary and secondary servers on the necessary ports. By default, servers in a CompleteFTP cluster use port **14983** to communicate, so this port must be open for access between the servers in the cluster.
- **IP Filtering:** Adjust the IP filtering rules on each secondary server to allow connections from the primary server. By default, secondary servers only permit connections from localhost or LAN IP addresses.

**Version Compatibility:** All servers in the cluster must be running the same version of CompleteFTP. Ensure that all installations are up to date and consistent to avoid synchronization issues.

# Clustering: Setting Up

## 1. Preparing the Environment

Before setting up clustering in CompleteFTP, ensure that your environment meets all the necessary prerequisites.

### 1.1 Installation

- Install **CompleteFTP Enterprise MFT** on all machines that will participate in the cluster, including both the primary and secondary servers.

### 1.2 Network Configuration

- Verify that the primary and secondary servers can communicate over the network.
- Open the necessary ports in your firewall settings to allow communication between servers. By default, CompleteFTP uses the SFTP protocol for server-to-server communication, which requires port **14983** to be open.

### 1.3 Adjusting IP Filters

By default, secondary servers may only permit connections from localhost or LAN IP addresses. To allow the primary server to communicate with secondary servers, adjust the IP filtering rules on each secondary server:

1. Open CompleteFTP Manager on the secondary server.
2. In the bottom-left corner, click on the **Options** menu and select **Show system users/folders/sites**.
3. Navigate to the **Sites** panel and select the **Admin Site**.
4. In the **IP Filtering** section, add a rule that allows connections from the primary server's IP address.

## 2. Configuring the Primary Server

### 2.1 Designating the Primary Server

The primary server is the central point for configuration management in the cluster. To designate a server as the primary:

1. Open CompleteFTP Manager and connect to the server you intend to use as the primary server.

### 2.2 Enabling Clustering

To enable clustering on the primary server:

1. In CompleteFTP Manager, navigate to the **Overview** panel.
2. Locate the **Servers** window, which is used to control the servers in the cluster.
3. Click on the **Add server** link. This action will automatically designate the current server as the primary server.

### 2.3 Synchronization Settings

After adding secondary servers (described in the next section), you can configure synchronization settings:

- Ensure that the **Sync enabled** option is checked for automatic synchronization of configuration changes to secondary servers.
- The default synchronization frequency is every 60 seconds. You can adjust this interval if necessary by modifying the synchronization settings in the **Servers** panel.

## 3. Adding Secondary Servers

### 3.1 Step-by-Step Process

To add a secondary server to the cluster, follow these steps:

1. In CompleteFTP Manager connected to the primary server, navigate to the **Servers** panel.
2. Click on the **Add server** link to open the **Add Server** dialog box.



3. Enter the secondary server's details:
  - **Hostname/IP Address:** The network address of the secondary server.
  - **Port:** The port number for the admin protocol (default is 14983 for SFTP).
  - **Username:** The username of an administrative user on the secondary server (usually *admin*).
  - **Password:** The password for the administrative user.
4. Click **Next** to proceed to the registration summary.
5. Review the settings and click **Add** to add the secondary server to the cluster.
6. A progress report will be displayed, detailing the registration process. The secondary server's configuration will be overwritten with a copy of the primary server's configuration.

### 3.2 Post-Addition Tasks

After successfully adding a secondary server:

- Verify that the secondary server appears in the **Servers** list in the **Overview** panel.
- Check the **Last synchronized** column to confirm that the secondary server has been synchronized with the primary server's configuration.

### 3.3 Troubleshooting Tips

If you encounter issues when adding a secondary server:

- **Check IP Filters:** Ensure that the IP filtering rules on the secondary server allow connections from the primary server's IP address.
- **Firewall Settings:** Verify that firewalls on both servers are configured to allow communication on the necessary ports (default is 14983 for SFTP).
- **Admin User Credentials:** Confirm that the admin user exists on the secondary server and that you have entered the correct username and password.
- **Network Connectivity:** Test the network connection between the primary and secondary servers to ensure they can communicate.

## 4. Configuring the Load Balancer

To distribute client requests across the servers in the cluster, configure your load balancer accordingly:

- **Set Up Load Balancing:** Configure the load balancer to distribute incoming client connections among the primary and secondary servers. The method of distribution (e.g., round-robin, least connections) can be chosen based on your requirements.
- **Health Checks:** Implement health checks to monitor the availability of each server. This ensures that the load balancer directs traffic only to servers that are operational.

### 4.2 Protocol Considerations

Different protocols may have specific considerations when configuring the load balancer:

#### HTTP/S

- **Session Stickiness (Sticky Sessions):** HTTP/S sessions are maintained per server. Configure the load balancer to use sticky sessions based on the client's IP address or cookies to ensure that all requests from the same client are directed to the same server. This is important to maintain session continuity.

#### Other Protocols (FTP, SFTP, FTPS)

- **Standard Load Balancing:** These protocols are generally stateless or handle state within each connection, so standard load balancing techniques can be applied without special considerations.

Ensure that your load balancer is configured to handle the specific requirements of each protocol used by your CompleteFTP servers.

# Clustering: Security Considerations

## 1. Access Controls and IP Filtering

In a CompleteFTP cluster, proper access control and IP filtering are essential to ensure secure communication between servers and prevent unauthorized access.

### 1.1 Adjusting IP Filters

By default, each server in the cluster has strict IP filtering rules that may only allow connections from localhost or LAN IP addresses. To ensure the primary and secondary servers can communicate securely:

1. Open CompleteFTP Manager on each secondary server.
2. Go to the **Sites** panel and select the **Admin Site**.
3. In the **IP Filtering** section, add a rule to allow connections from the primary server's IP address.
4. Ensure that only trusted IP addresses (including the primary and other secondary servers) are allowed to communicate with each server in the cluster.
5. Remember that each secondary server in the cluster must have open two-way communication with the primary.

Using strict IP filtering helps protect your cluster from unauthorized external access while allowing internal servers to synchronize configurations and data.

### 1.2 User Permissions

User permissions are synchronized across the cluster, and CompleteFTP uses a Linux-like security model where user permissions are based on **groups** rather than individual roles. This simplifies managing permissions for large numbers of users. Best practices for managing user access include:

- **Group-Based Access Control:** Assign users to groups based on their access needs. Managing permissions at the group level ensures consistency and simplifies permission management across the cluster.
- **Limit Administrative Access:** Ensure that only authorized users have administrative privileges in CompleteFTP. Regularly review and audit access to administrative accounts.

## 2. Securing Communication

Securing communication between clients and servers, as well as between servers in the cluster, is critical for preventing data interception and unauthorized access. CompleteFTP supports SSL/TLS encryption for specific protocols to protect these communications.

### 2.1 SSL/TLS Certificates

For secure connections using FTPS or HTTPS, it is important to install valid SSL/TLS certificates on the relevant **sites** (not individual servers):

1. Obtain an SSL/TLS certificate from a trusted certificate authority (CA) or create a self-signed certificate for internal use.
2. Install the certificate on each site that requires encrypted connections via FTPS or HTTPS using CompleteFTP Manager.
3. Ensure that the certificates are up to date and renew them before they expire to avoid service interruptions.

Installing SSL/TLS certificates on the relevant sites ensures that communication between clients and servers is encrypted and secure. Note that SSL/TLS does not apply to SFTP, which uses its own encryption mechanism.

### 2.2 Securing the Admin Site

The Admin Site is used for server management and configuration. To prevent unauthorized access, it is important to secure the Admin Site:

- **Strong Passwords:** Use strong, complex passwords for all administrative accounts. Ensure that password policies require a mix of letters, numbers, and special characters.
- **IP Filtering:** Restrict access to the Admin Site to trusted IP addresses only, such as internal network addresses or specific management machines.

### 2.3 Firewall Configurations

Firewalls play a crucial role in securing communication between servers and clients. Ensure that firewalls are configured to:

- **Limit Open Ports:** Only open the necessary ports for the protocols you are using (e.g., SFTP, FTPS, HTTPS). Close all other unused ports to minimize attack vectors.
- **Restrict Incoming Connections:** Restrict incoming connections to trusted IP addresses or networks, ensuring that only authorized users and servers can communicate with the CompleteFTP cluster.
- **Monitor Firewall Logs:** Regularly monitor firewall logs for unusual activity or attempts to access the server from unauthorized IP addresses.

## 3. Securing Server-to-Server Communication

In a CompleteFTP cluster, the primary server communicates with secondary servers to synchronize configuration changes. It is essential to secure this communication to prevent tampering or unauthorized access.

### 3.1 Using Encrypted Protocols

By default, CompleteFTP uses the SFTP protocol (port 14983) for server-to-server communication. This protocol is encrypted, ensuring that all data transferred between the primary and secondary servers is secure. If needed, you can configure CompleteFTP to use other encrypted protocols like FTPS.

- **Port Configuration:** Ensure that the firewall allows communication over the designated protocol and port (e.g., 14983 for SFTP).

- **Use Strong Encryption:** Configure CompleteFTP to use strong encryption algorithms for all communication. Avoid using outdated or weak ciphers.

Additionally, ensure that network security protocols are in place to prevent man-in-the-middle attacks or eavesdropping on server-to-server communication.

### 3.2 Regular Security Audits

It is important to regularly audit the security of your CompleteFTP cluster:

- **Log Monitoring:** Enable audit logs for all administrative actions and regularly review them for suspicious activities, especially around administrative access and configuration changes.
- **Update Management:** Keep all servers in the cluster up to date by installing the latest version of CompleteFTP. Unlike patches, CompleteFTP releases involve upgrading the entire product to the latest version.
- **Security Testing:** Conduct periodic vulnerability assessments or penetration testing to identify and address potential security issues in the cluster.

By following these security considerations, you can ensure that your CompleteFTP cluster is protected against unauthorized access and data breaches, while maintaining secure communication between servers and clients.

# Clustering: Management

## 1. Monitoring and Administration

### 1.1 Using CompleteFTP Manager

Effective management of your CompleteFTP cluster involves monitoring server status and administering servers through the CompleteFTP Manager. The **Servers** panel within the **Overview** tab provides a centralized view of all servers in the cluster.

#### Monitoring Server Status:

- **Server List:** The Servers panel displays a list of all servers in the cluster, including the primary and all secondary servers.
- **Synchronization Status:** For each server, you can view the last synchronization time and status, indicating whether the server is up-to-date with the primary server's configuration.
- **Server States:** Servers may have different states such as *Online*, *Offline*, or *Synchronizing*. These states help you understand the current status of each server.

### 1.2 Server Management

#### Managing Secondary Servers:

Once a secondary server has been added to the cluster, the primary server manages its configuration synchronization, but there are several actions you can take to manage the secondary servers effectively:

- **Adding Secondary Servers:** You can add more secondary servers to the cluster by using the **Add server** link in the **Servers** panel. This allows you to scale the cluster as needed.
- **Removing Secondary Servers:** If a secondary server is no longer required or is to be replaced, it can be removed from the cluster. To do this:
  1. Right-click on the secondary server in the **Servers** panel.
  2. Select **Remove server** from the context menu to remove it from the cluster.

Removing a server from the cluster will stop the synchronization of configuration data between the primary and that server. It will revert to a standalone server with the same configuration as it had prior to being removed.

#### Viewing and Managing Sites:

- **Accessing Sites:** In the CompleteFTP Manager, navigate to the **Sites** panel to view and manage the sites configured on each server.
- **Site Configuration:** You can add, modify, or delete sites as needed. Remember that any changes made on the primary server will be synchronized to the secondary servers.

## 2. Synchronization Settings

### 2.1 Sync Enabled Option

The **Sync enabled** option controls whether configuration changes made on the primary server are automatically synchronized to the secondary servers.

- **Importance of Sync Enabled:** Keeping this option checked ensures that all servers in the cluster remain consistent with the primary server's configuration. It is recommended to keep synchronization enabled unless there is a specific reason to disable it temporarily.
- **How to Enable/Disable:** In the **Servers** panel, you can enable or disable synchronization for each secondary server by checking or unchecking the **Sync enabled** checkbox next to the server's name.

### 2.2 Adjusting Synchronization Frequency

By default, synchronization occurs every 60 seconds. You may adjust this interval based on your network performance and the frequency of configuration changes.

- **Modifying the Interval:** To change the synchronization frequency:
  1. In the CompleteFTP Manager, select the secondary server in the **Servers** panel.
  2. Right-click and choose **Properties** from the context menu.
  3. In the server properties dialog, adjust the **Synchronization Interval** to the desired number of seconds.
- **Considerations:** A shorter interval means changes are propagated more quickly but may increase network traffic. A longer interval reduces network load but may delay the application of configuration changes on secondary servers.

### 2.3 Manual Synchronization

If you need to force a synchronization outside of the regular interval, you can manually synchronize a secondary server.

- **Forcing Synchronization:**
  1. In the **Servers** panel, right-click on the secondary server you wish to synchronize.
  2. Select **Force configuration update** from the context menu.
- **Use Cases:** Manual synchronization is useful when immediate propagation of changes is necessary or if a secondary server has fallen out of sync due to network issues.

## 3. Changing the Primary Server

There may be situations where you need to change which server acts as the primary in your cluster, such as decommissioning the current primary server or promoting a more capable server to primary status.

### 3.1 Promoting a Secondary to Primary

To promote a secondary server to become the new primary:

1. In the CompleteFTP Manager connected to the current primary server, navigate to the **Servers** panel.
2. Right-click on the secondary server you wish to promote and select **Select as primary** from the context menu.

The selected secondary server will now become the primary server. All configuration changes should henceforth be made on this new primary server.

### 3.2 Removing the Old Primary

If the old primary server is to be decommissioned or removed from the cluster:

1. Connect to the new primary server using the CompleteFTP Manager.
2. In the **Servers** panel, right-click on the old primary server (now a secondary) and select **Remove server**.

This will remove the old primary server from the cluster. You can now proceed to decommission or repurpose the server as needed.

### 3.3 Considerations

- **Avoid Configuration Changes During Transition:** It is important not to make any configuration changes during the transition of primary servers to prevent synchronization conflicts.
- **Confirm Synchronization:** After promoting a new primary server, verify that all secondary servers are synchronizing correctly with the new primary.
- **Update Load Balancer Settings:** If your load balancer has been configured to minimize load on the primary server, you may need to adjust its settings to reflect the change in primary server.

# Clustering: Upgrading

## 1. Steps to Upgrade

Upgrading your CompleteFTP cluster is a straightforward process, but it must be done carefully to avoid any disruption in service. All servers in the cluster must be upgraded to the same version of CompleteFTP for compatibility and proper synchronization. Below are the steps to perform an upgrade in your cluster.

### 1.1 Preparation

Before starting the upgrade, follow these preparation steps:

- **Backup Configurations and Data:** Before proceeding with the upgrade, it is critical to back up your server configurations and data on all servers. This ensures that if anything goes wrong during the upgrade process, you can restore the previous configuration and data. While CompleteFTP automatically backs up configurations daily, it's advisable to perform a manual backup before upgrades.
- **Verify Licensing and Downloads:** Ensure that you have valid licenses for the new version. Download the installer from your official EnterpriseDT account to ensure access to the correct version and proper licensing.
- **Notify Users:** Inform users of potential downtime or service interruptions during the upgrade process. Though the process is generally quick, users should be made aware in case of short interruptions.
- **Test in a Non-Production Environment:** If possible, perform the upgrade on a test or staging environment before applying it to your production cluster. This helps identify and resolve potential issues without impacting users.
- **Plan for Rollback:** Prepare a rollback plan in case severe issues arise after the upgrade. This includes understanding the downgrade process and having backups readily available.

### 1.2 Upgrade Process

Follow these steps to upgrade the CompleteFTP cluster:

1. **Avoid Configuration Changes:** Do not make any configuration changes on any server during the entire upgrade process. This prevents synchronization issues and potential data loss.
2. **Stop Primary Server:** Close CompleteFTP Manager and stop the CompleteFTP Service on the primary server to prevent failures when upgraded secondaries contact the primary post-upgrade. Note that such failures should not actually cause any issues, but may as well be avoided.
3. **Upgrade Secondary Servers First:**
  - On each secondary server:
    - Close CompleteFTP Manager if it is open.
    - Stop the CompleteFTP service.
    - Run the new installer directly over the existing installation. There's no need to uninstall the old version, though no harm in doing so, as configuration data is not deleted.
    - During installation, ensure you use the same administrator password as on the primary server.
    - The CompleteFTP service will automatically start after installation. Each secondary server will attempt to reconnect with the primary server. If the primary is still running the old version, synchronization will not occur until the primary is upgraded.
4. **Upgrade the Primary Server:**
  - Run the new installer directly over the existing installation. Do not uninstall the old version.
  - Ensure you use the same administrator password during installation.
5. **Restart the Cluster:**
  - Start the CompleteFTP service on the primary server.
  - The primary server will now synchronize configurations with the secondary servers.

### 1.3 Post-Upgrade Verification

Once the upgrade is complete:

- **Check Synchronization:** Verify that all secondary servers are synchronized with the primary server. You can check the **Servers** panel in CompleteFTP Manager for synchronization status.
- **Test Server Functionality:** Perform comprehensive tests to ensure that the servers are functioning correctly after the upgrade. This includes testing all protocols (e.g., FTP, SFTP), verifying user authentication, and conducting file transfers.
- **Monitor Logs:** Review CompleteFTP logs for any errors or warnings that may have occurred during or after the upgrade to identify and address potential issues promptly.

## 2. In-Place Upgrades

An in-place upgrade allows you to upgrade individual servers in the cluster without taking the entire cluster offline. This can help minimize downtime, especially in production environments where constant availability is critical.

### 2.1 Steps for In-Place Upgrades

To perform an in-place upgrade, keeping in mind that the primary server should be stopped while the secondary servers are upgraded and upgraded last, follow these steps:

1. **Avoid Configuration Changes:** Do not make any configuration changes on any server during the entire upgrade process. This prevents synchronization issues and potential data loss.
2. **Redirect Traffic Away from the Primary Server:** Configure your load balancer to redirect all client traffic away from the primary server to the secondary servers. This ensures continuous service availability while the primary server is offline.
3. **Stop the Primary Server:**

- Close CompleteFTP Manager on the primary server.
  - Stop the CompleteFTP service on the primary server to prevent failures when upgraded secondary servers attempt to contact the primary server post-upgrade.
4. **Upgrade Secondary Servers First:**
- On each secondary server, one at a time:
    - Close CompleteFTP Manager if it is open.
    - Stop the CompleteFTP service.
    - Run the new installer directly over the existing installation. There's no need to uninstall the old version, though there's no harm in doing so, as configuration data is not deleted.
    - During installation, ensure you use the same administrator password as on the primary server.
    - The CompleteFTP service will automatically start after installation.
    - Each secondary server will attempt to reconnect with the primary server. Since the primary server is stopped, synchronization will not occur until the primary is upgraded and restarted.
    - Reintroduce the upgraded secondary server into the load balancer to handle client traffic.
5. **Upgrade the Primary Server:**
- Run the new installer directly over the existing installation on the primary server. There's no need to uninstall the old version.
  - Ensure you use the same administrator password during installation.
6. **Restart the Cluster:**
- Start the CompleteFTP service on the primary server.
  - The primary server will now synchronize configurations with the secondary servers.
  - Reintroduce the primary server into the load balancer to resume normal operation.

## 2.2 Considerations During In-Place Upgrades

- **Avoid Configuration Changes:** Do not make any configuration changes during the in-place upgrade process to prevent conflicts or synchronization issues between servers running different versions.
- **Expect Temporary Communication Disruptions:** Since the primary server is stopped during the upgrade, secondary servers will be unable to synchronize configurations until the primary server is upgraded and restarted. This is expected and will be resolved once the primary server is back online.
- **Capacity Planning:** Ensure your cluster has sufficient capacity to handle the client load with the primary server offline during the upgrade.

## 2.3 Post-Upgrade Verification

Once the in-place upgrade is complete:

- **Check Synchronization:** Verify that all secondary servers are synchronized with the primary server by checking the **Servers** panel in CompleteFTP Manager.
- **Test Server Functionality:** Perform comprehensive tests to ensure that all servers are functioning correctly after the upgrade, including testing all protocols and conducting file transfers.
- **Monitor Logs:** Review CompleteFTP logs for any errors or warnings that may have occurred during or after the upgrade to identify and address potential issues promptly.

## 3. Common Upgrade Issues and Troubleshooting

### 3.1 Version Mismatch

One of the most common issues during an upgrade is a version mismatch between the primary and secondary servers. All servers in the cluster must run the same version of CompleteFTP to maintain proper synchronization.

**Solution:** Ensure that each server is upgraded in the correct order (secondary servers before the primary server), and verify that all servers are running the same version after the upgrade process. If a version mismatch occurs, upgrade the out-of-sync servers immediately.

### 3.2 Synchronization Failures After Upgrade

If secondary servers fail to synchronize with the primary server after an upgrade, the issue may be related to network configuration or firewall rules.

**Solution:** Check that firewalls and network settings are configured correctly to allow communication on the necessary ports between the primary and secondary servers. Ensure that IP filtering rules on the secondary servers allow connections from the upgraded primary server.

### 3.3 Failed Installation

In rare cases, an installation might fail due to a corrupted installer file or permission issues.

**Solution:** Re-download the installer from your official EnterpriseDT account to ensure file integrity. Run the installation as an administrator to avoid permission-related issues.

## 4 Best Practices for Cluster Upgrades

To ensure a smooth and trouble-free upgrade of your CompleteFTP cluster, consider the following best practices:

- **Perform Upgrades During Maintenance Windows:** Schedule upgrades during planned maintenance windows to minimize the impact on users.
- **Test in a Staging Environment:** Perform the upgrade process in a staging or test environment before applying it to your production cluster to identify potential issues.
- **Backup Configurations and Data:** Always back up your configurations and data before starting the upgrade process.
- **Avoid Configuration Changes During Upgrade:** Do not make any changes to configurations during the upgrade to prevent synchronization conflicts.
- **Monitor Servers Post-Upgrade:** After completing the upgrade, monitor the servers closely for any issues such as synchronization problems or performance degradation. Address

any problems promptly to ensure stability.

- **Verify Firewall and Network Settings:** Ensure that firewall rules and network configurations allow for proper communication between cluster nodes after the upgrade.
- **Plan for Rollback:** Have a rollback plan in case severe issues arise after the upgrade, including backups and knowledge of the downgrade process.

By following these steps and best practices, you can upgrade your CompleteFTP cluster smoothly and with minimal downtime, ensuring that your servers remain up to date and fully synchronized.



# Clustering: Troubleshooting

Clustering in CompleteFTP can provide significant advantages, such as load balancing and high availability, but like any distributed system, it may occasionally encounter issues. This section addresses common problems, their causes, and solutions to help you effectively maintain a stable and reliable CompleteFTP cluster.

## 1. Common Clustering Issues

### 1.1 Failure to Add a Secondary Server

#### Symptoms:

- Unable to add a secondary server to the cluster.
- Error messages related to connection failures or invalid credentials.

#### Potential Causes:

- **IP Filtering Rules:** The secondary server may not allow connections from the primary server's IP address.
- **Firewall Settings:** The firewall on either the primary or secondary server may be blocking required ports (default: 14983 for SFTP-based communication).
- **Incorrect Admin Credentials:** The admin username or password entered when attempting to add the secondary server may be incorrect.
- **Version Mismatch:** The primary and secondary servers may be running different versions of CompleteFTP.

#### Solutions:

1. **Verify IP Filtering Rules:** Check that the secondary server allows connections from the primary server's IP address. Adjust the IP filtering settings in the Admin Site on the secondary server to permit connections from the primary server.
2. **Check Firewall Settings:** Ensure the firewalls on both the primary and secondary servers are configured to allow communication on port 14983 for SFTP-based server-to-server communication.
3. **Double-check Admin Credentials:** Ensure the admin account exists on the secondary server and the correct username and password are being used. Verify that the account has the necessary permissions to join the cluster.
4. **Confirm Version Compatibility:** Verify that both the primary and secondary servers are running the same version of CompleteFTP. If not, upgrade the servers to the same version before proceeding.

### 1.2 Synchronization Failures Between Primary and Secondary Servers

#### Symptoms:

- Secondary servers fail to synchronize with the primary server.
- Error messages such as "Synchronization failed" or "Unable to connect to primary server."

#### Potential Causes:

- **Network configuration:** Ensure that both the primary and secondary servers are allowed to communicate over the network. Common causes include closed firewall ports or incorrect IP filtering.
- **Version mismatch:** All servers in the cluster must be running the same version of CompleteFTP. Version differences can cause synchronization to fail.
- **IP Filtering:** The IP filtering settings on the secondary servers may not allow incoming connections from the primary server.

#### Solutions:

1. **Verify firewall and network settings:** Ensure the necessary ports (default: 14983 for SFTP) are open between the primary and secondary servers. Also, ensure the network allows bidirectional communication.
2. **Check IP filtering rules:** On each secondary server, allow the primary server's IP address through the IP filtering settings. To modify these settings, connect to each secondary server, go to the Admin Site settings in the Sites panel, and ensure the primary's IP is whitelisted.
3. **Confirm version compatibility:** Check that all servers in the cluster are running the same version of CompleteFTP. If needed, upgrade all servers to the same version before attempting to synchronize again.
4. **Force synchronization:** In the CompleteFTP Manager, go to the **Servers** panel and select "Force configuration update" on the problematic secondary server.

### 1.3 File Synchronization Issues

#### Symptoms:

- Files are not synchronized between servers in the cluster.
- Inconsistent file availability across different servers.

#### Potential Cause:

- **Misunderstanding of the clustering feature:** CompleteFTP clustering **does not synchronize files** between servers—it only synchronizes server configurations.

#### Solutions:

- **External file synchronization:** Use an external method such as **DFS (Distributed File System)** or shared network storage (NAS/SAN) to manage file synchronization across servers. Ensure that your file synchronization solution is properly configured and suited to your network architecture.

## 1.4 Primary Server Fails or Becomes Unavailable

### Symptoms:

- The primary server crashes or becomes unresponsive.
- Secondary servers can no longer receive configuration updates.

### Potential Causes:

- **Hardware failure** or software crash on the primary server.
- **Network issues** causing loss of connectivity between the primary and secondary servers.

### Solutions:

1. **Promote a secondary server to primary:** In the CompleteFTP Manager, right-click the desired secondary server and select "**Select as primary**". This promotes the secondary to act as the new primary server, restoring configuration synchronization across the cluster.
2. **Fix or replace the old primary server:** Once the new primary is functioning, troubleshoot the failed server or decommission it. If the original primary server is restored, you may want to re-add it as a secondary.

## 1.5 Licensing Issues During Clustering Setup

### Symptoms:

- License activation errors during the addition of secondary servers.
- Secondary servers fail to synchronize licensing from the primary.

### Potential Causes:

- **Missing licenses** for secondary servers.
- **Network connectivity issues** preventing license validation.

### Solutions:

1. **Check licensing requirements:** Each server in a cluster requires its own **Enterprise edition** license. Verify that licenses are valid and activated on both primary and secondary servers.
2. **License troubleshooting:** If licensing issues persist, contact support for assistance with reactivating or validating your licenses.

## 1.6 Unbalanced Server Load

### Symptoms:

- Client connections are not properly distributed across servers in the cluster.
- Some servers experience heavy loads while others remain underutilized.

### Potential Causes:

- **Load balancer configuration:** An external load balancer may not be properly routing traffic across all secondary servers, resulting in uneven load distribution.

### Solutions:

1. **Review the external load balancer configuration:** Ensure that the load balancer is configured to distribute traffic evenly across all servers in the cluster. This includes setting up appropriate rules and ensuring health checks are working correctly.
2. **Sticky sessions for HTTP/S:** If you are using HTTP or HTTPS, ensure that sticky sessions (session persistence) are enabled to route clients to the same server for the duration of their session.
3. **Test and adjust load distribution:** Regularly test how traffic is being distributed by the load balancer to ensure that no server is being overloaded while others are idle. Adjust load balancer rules as necessary.

Note: CompleteFTP itself is not responsible for managing the distribution of client connections. Proper traffic balancing must be handled by the external load balancer used in your environment.

## 2 Diagnostic Tools and Logs

### 2.1 Reviewing Logs

CompleteFTP provides detailed logs that can help diagnose issues in the cluster. Logs can be accessed directly from the server's file system.

**Log Location** Logs for each server are located in `C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Logs` on each server. These logs can provide insights into any network-related or configuration errors affecting the clustering setup.

It is recommended that users access secondary server logs directly from the file system, as the CompleteFTP Manager typically only shows logs for the server to which it is connected (usually the primary server).

### 2.2 Enabling Debug Logging

In the event of recurring issues, you can increase the level of detail captured in the logs by enabling debug logging. This can help capture more specific information about the server's operations and errors.

**How to Enable Debug Logging:** In the CompleteFTP Manager, go to **Settings > Logging** and set the logging level to **Debug** for more detailed diagnostic information.

### 3. Preventative Maintenance and Best Practices

- **Regular Backups:** Regularly back up the primary server's configuration to ensure you can quickly restore your cluster in the event of a failure.
- **Monitor Server Health:** Use monitoring tools to keep track of server performance, network connectivity, and storage usage to prevent resource exhaustion.
- **Test Failover Procedures:** Periodically test your failover procedures by promoting a secondary server to primary and verifying that synchronization and configuration management work as expected.
- **Keep Servers Up-to-Date:** Ensure all servers are running the latest version of CompleteFTP to benefit from bug fixes and performance improvements.

By following these troubleshooting steps and best practices, you can maintain a robust and resilient CompleteFTP cluster that meets your high availability and performance needs.

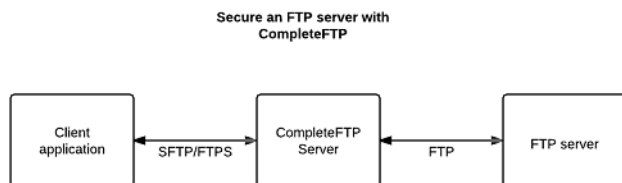
# Gateway

CompleteFTP (Enterprise MFT only) can act as a *multiprotocol gateway* for other servers. It can present any of its own protocols to a client, and translate them into a different protocol to talk to another server.

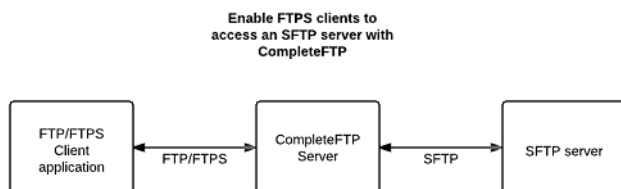
The multiprotocol gateway is an extremely powerful feature. A number of possible uses are described below. [Gateway configuration](#) is described [here](#).

## Gateway examples

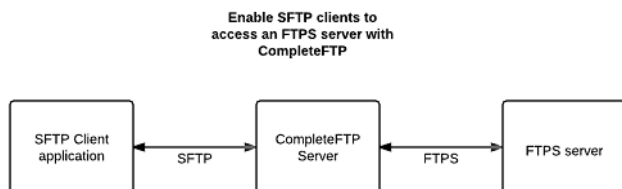
1. Provide secure access to another non-secure FTP server located on the same machine or network. CompleteFTP provides a secure wrapper to the secure server. This means SFTP or FTPS clients can connect to CompleteFTP, which proxies the requests on to the non-secure server.



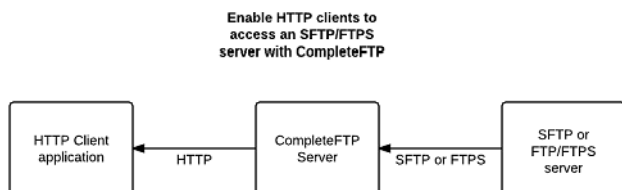
2. Add FTP or FTPS support to an existing SFTP server. This means FTP or FTPS clients can access the SFTP server without any other changes than setting up CompleteFTP to proxy the requests.



3. Add SFTP support to an existing FTP or FTPS server. This means SFTP clients can access the FTP or FTPS server without any other changes than setting up CompleteFTP to proxy the requests.



4. Add HTTP support to an existing FTP, FTPS or SFTP server. This means HTTP clients can access the FTP or FTPS server without any other changes than setting up CompleteFTP to proxy the requests.



# Gateway configuration

There are two ways of configuring the gateway to access a particular server. Both cases require a **gateway folder** to be configured, which is analogous to a remotely mounted directory. When users login and navigate to a gateway folder, they are actually accessing the remote file system. A gateway folder can be set as a user's home directory.

To configure the gateway to access a remote server, either:

1. Set up a **gateway authenticator**, where the remote server is used to authenticate a user that is trying to log in. The authenticated connection to the server is saved in the user's session. Then set up the **gateway folder** to use the gateway connection, or
2. Set up a **gateway folder** that uses a specified connection, i.e. the gateway folder has the proxied server's credentials added, and every user that accesses the gateway folder uses these credentials.

# How to configure a gateway authenticator

Gateway users require a gateway authenticator to be configured to allow users to be authenticated using a different server. This is a feature of the Enterprise MFT only of CompleteFTP.

A gateway authenticator is an **authentication extension** that authenticates user attempting to log in via another server, using whatever protocol is selected. When the user details are received in CompleteFTP, a login attempt to the remote server is attempted using the supplied credentials.

To configure a gateway authenticator, select "General user settings" link from the Users tab. "Gateway" is listed amongst "Other authentication methods", and has a "Configure" link that must be selected (below).

Authenticator	Enabled	Log in as	
Database	<input type="checkbox"/>	defaultDatabase	<a href="#">Configure...</a>
<b>Gateway</b>	<input checked="" type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Single sign-on / SAML	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Windows/AD Group	<input type="checkbox"/>	defaultWindows	<a href="#">Configure...</a>

This brings up the "Gateway configuration" dialog box which allows the addition and removal of gateways (shown below). Multiple gateways can be configured, meaning multiple different machines can be tried for authentication, in the order in which they are listed.

Gateway Configuration

Add gateway Remove gateway(s)

New Gateway 1

Connection Properties:

Basic

Name

Protocol

Server address

Server port

Initial directory

Advanced

Character Encoding

Connections

Directory listings

Transfer

Basic Connection Settings

Shows the basic connection settings

OK Cancel

There are both "Basic" and "Advanced" connection properties to be configured for each gateway. Usually the basic properties are all that is required - the protocol, the server address and port, and optionally the initial directory to change to on the remote machine.

If the basic settings are not sufficient, the advanced settings should be examined. These include character encoding; connection settings such as proxies, timeouts and active and passive settings; directory listing settings such as caching of listings; and transfer settings. These settings permit a fine degree of configuration control over a gateway authenticator.

If SFTP with public key authentication is required and the users' public keys are to be stored on the remote server then the remote server must (1) be a CompleteFTP server, (2) have SFTP enabled and (3) have the password authentication method enabled. Even though password authentication must be enabled on the remote server, the user is indeed verified on that server via their public key.

# How to use CompleteAgent

## Overview

**CompleteAgent** is a cross-platform client that is designed to perform robust, automatic uploads and downloads to and from CompleteFTP. CompleteAgent can run as a standalone application on Windows and Linux, but on Windows is typically run as a service. It monitors a designated local folder for new files and securely uploads them to a specified CompleteFTP server when they appear. It also monitors a designated remote folder for new files, and securely downloads them to a local directory.

## Requirements

CompleteAgent is a free client application that requires an Enterprise MFT server to connect to. It can be freely installed on any machine. As well as being a Windows service, CompleteAgent includes a .NET standard binary that can be run on any platform (including Linux) that supports .NET standard.

## Installation

CompleteAgent is installed as part of the CompleteFTP trial, and is optionally installed by the production installer. In both cases, it is installed as a Windows service, but only the trial starts the service running.

CompleteAgent requires user credentials to connect to the user - the trial creates a user called `completeAgent`, and sets up a private key, so that it is ready to go once the trial is installed. The trial also is set up to monitor a remote directory called `downloads` in the `completeAgent` user's home directory - see the configuration file. This directory will need to be created on the server and files placed in it for automatic downloads to begin.

## Configuration

By default, CompleteAgent is installed in a subdirectory of the main CompleteFTP installation directory, called `CompleteAgent`. Typically then, it can be found at `C:\Program Files\Complete FTP\CompleteAgent`. The Windows-specific binaries are found in the `net` subdirectory, while the cross-platform .NET standard binaries are found in `standard`.

The main configuration file for CompleteAgent is found at `C:\ProgramData\Enterprise Distributed Technologies\CompleteAgent\CompleteAgent.ini`. The various options that can be set in the configuration file are also available as command-line options if the binary is run from the command-line rather than as a service. The long names of the command-line options are the same names used in the configuration file.

If the configuration file is altered, the CompleteAgent service must be restarted.

## Command-line options

Although CompleteAgent is installed as a Windows service, it can also be run from the command-line - and installed as a Windows service from the command-line.

```
install - install CompleteAgent as a Windows service
uninstall - uninstall the CompleteAgent service
start - start the CompleteAgent service
stop - stop the CompleteAgent service
```

The options below are used when running CompleteAgent from the command-line. Please note that these options have changed slightly from the original 1.0 options (which was when CompleteAgent only supported uploads). For example, `RootDirectoryPath` is now `RootDirectory`. Most of the settings are not required as they have defaults.

The most important options are shown below, and can also be obtained to running `Completeagent.exe` on the command line with the `-h` option. The `-hh` option displays the less frequently used options.

```
-rd or -RootDirectory - the base directory under which all files and directories will be placed by default - defaults to C:\ProgramData\Er
-s or -ServerAddress - IP address or host-name of server
-sp or -ServerPort - server port - defaults to 22
-u or -UserName - user-name of account on server
-p or -Password - password of account on server
-kf or -PrivateKeyFilePath - path of private key file to use
-kp or -PrivateKeyPassword - password for private key file
-su or -RemoteUploadDir - remote target directory on server for uploads - defaults to user's home directory
-sd or -RemoteDownloadDir - remote target directory on server for polling for downloads - if not supplied, no polling for downloads is ini
-l or -LogLevel - logging level (All, Debug, Information, Warning, Error) - defaults to Information
-h or -help - show help for most important options
-hh - show help for all options
```

## Using CompleteAgent

By default, CompleteAgent is configured to detect files being placed in `C:\ProgramData\Enterprise Distributed Technologies\CompleteAgent\Uploads\ToBeUploaded`. Once files are placed here, they are securely uploaded to the location specified in the configuration, using the named user and their credentials.

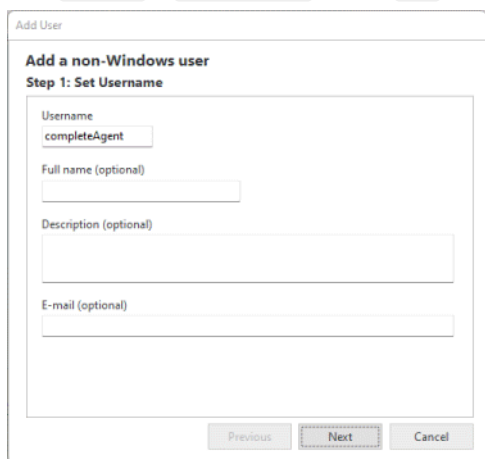
Successfully uploaded files are moved to the `C:\ProgramData\Enterprise Distributed Technologies\CompleteAgent\Uploads\Completed`. If an upload fails, the file is eventually moved to `C:\ProgramData\Enterprise Distributed Technologies\CompleteAgent\Uploads\Failed`. Diagnostic information on the failure can be found in the logs, which are written to `C:\ProgramData\Enterprise Distributed Technologies\CompleteAgent\Logs`.

If CompleteAgent is configured for automatic downloads, when found on the remote server, files are initially downloaded into the `C:\ProgramData\Enterprise Distributed Technologies\CompleteAgent\Downloads\InProgress` directory. When the download has successfully completed, the file will be moved into `C:\ProgramData\Enterprise Distributed Technologies\CompleteAgent\Downloads\Completed`. If a download fails, the file is moved to `C:\ProgramData\Enterprise Distributed Technologies\CompleteAgent\Downloads\Failed`. In the case of a failure, this will generally be a partially downloaded file. Diagnostic information on the failure can be found in the logs.

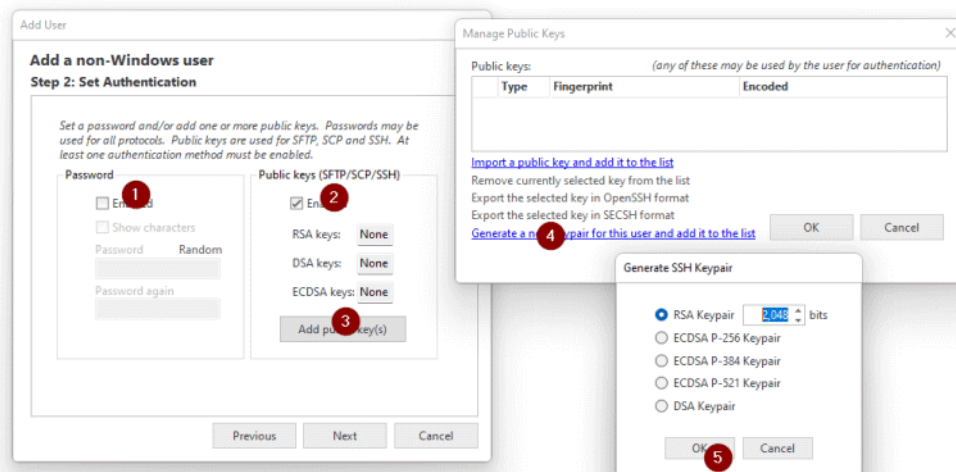
## Setting up CompleteAgent in Production Installations

Instances of CompleteFTP that were initially installed using the production installer (rather than the trial installer) don't have an account set up for CompleteAgent, so it's necessary to create such an account before starting the CompleteAgent Service. By default, the CompleteAgent configuration file authenticates using the username `completeAgent`, using a private key that's in the CompleteAgent data directory. The steps for creating such an account and private key are given below:

1. Open CompleteFTP Manager
2. Go to the `Users` panel
3. Click `Add user` → `Non-Windows user`
4. Set the `Username` to `completeAgent` and click `Next`

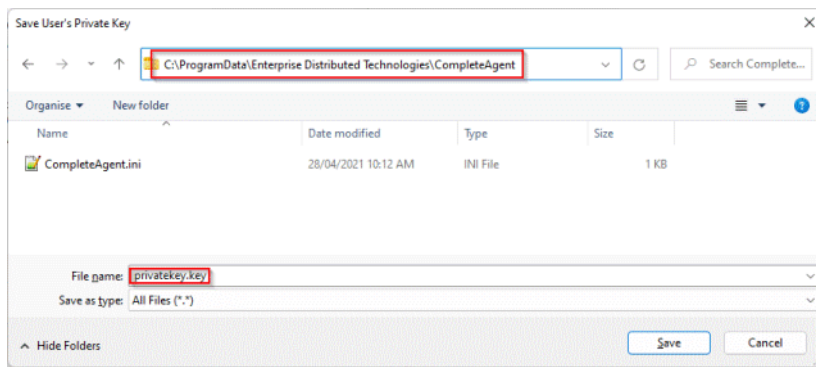


5. Uncheck `Enabled` under `Password` and check `Enabled` under `Public keys`, click `Add public key(s)`, click `Generate a new keypair` and click `OK`



6. After you click `OK`, you'll be prompted for a location to save the private key file. Save it in `C:\ProgramData\Enterprise Distributed Technologies\CompleteAgent` and name it `privatekey.key`





7. When prompted for a password, just click **OK**, leaving it blank.
8. Finish the Add non-Windows user wizard
9. Click **Apply changes**
10. Use the Windows Services console to start the **CompleteAgent** service.

Once you've done that, CompleteAgent should have created two SFTP-based 'file pipelines':

#### Uploads:

C:\ProgramData\Enterprise Distributed Technologies\CompleteAgent\Uploads\ToBeUploaded →  
C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Users\completeAgent

#### Downloads:

C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Users\completeAgent\Downloads →  
C:\ProgramData\Enterprise Distributed Technologies\CompleteAgent\Downloads\Completed

You can test it by copying a file into C:\ProgramData\Enterprise Distributed Technologies\CompleteAgent\Uploads\ToBeUploaded and confirming that it appears in C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Users\completeAgent. Similarly, you can confirm that any file copied into C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Users\completeAgent\Downloads will appear in C:\ProgramData\Enterprise Distributed Technologies\CompleteAgent\Downloads\Completed.

Of course, the point is that the client and server can each be anywhere on the Internet, rather than on the same computer, but the transport mechanism is the same as in this example.

Any configuration changes required for CompleteAgent must be done by editing the file, C:\ProgramData\Enterprise Distributed Technologies\CompleteAgent\CompleteAgent.ini.

# Events

In CompleteFTP Professional and Enterprise MFT, an *event* is an action performed by a logged in user on the server, or an action performed by the server itself (a [scheduled event](#)). For example, uploading or downloading a file is an event.

**Email notifications** can be attached to these events, so that when, for example, a file is uploaded, an email is sent to a specified user providing details of the upload. Email notifications are described in more detail [here](#).

**Process triggers** can also be attached to these events, so that when the event is triggered, a program can be launched. Process triggers are described in more detail [here](#).

These events can also be recorded in the [audit log](#). Note that email notifications, process triggers and auditing are not supported in the Standard Edition.

# How to set up email notifications

CompleteFTP supports highly customizable email notifications. This means that when an event such as a file upload occurs, an email can be sent.

The screenshot shows the 'Email Notification' configuration window. On the left is a list of notifications with columns 'Enabled' and 'Name'. Two notifications are listed: 'New video file' and 'New audio file', both with the 'Enabled' checkbox checked. At the top of the main panel are buttons: '+ Add', '- Remove', and a 'Copy' icon. The 'SMTP Server' section contains fields for 'Host-name / IP address' (smtp.myserver.com), 'Port' (25), an 'SSL/TLS' checkbox, 'Username' (myuser), and 'Password' (masked with asterisks). Below this is a 'Show password characters' checkbox. The 'Name' field is set to 'New video file'. The 'Events' section has a checked 'Upload file' checkbox and a 'Choose' button. The 'Errors' section has checked 'Trigger on success' and unchecked 'Trigger on error' checkboxes. The 'Folder filter' is '/media/video/\*', 'User filter' is 'Mary', and 'Site filter' is 'Default Site'. Each filter has an 'Inverse' checkbox. The 'From' field is 'completeftp@myserver.com', 'To' is 'webmaster@myserver.com', and 'Subject' is 'New video has been uploaded: %FileName%'. The 'Message' field contains a template: 'A new video has been uploaded: User = %LoginUserName% Path = %VirtualPath%'. At the bottom is a 'Notifications' section with a close button and a description: 'Shows the list of email notifications. The checkbox indicates/controls whether or not a process trigger is enabled.'

To set up email notifications, the mail server (SMTP server) settings must be entered. The server hostname, port number (normally port 25) and optional username and password are entered in the Email Notification tab found in the *Events* panel. SSL connections are also permitted by checking the "Enabled SSL/TLS" checkbox. For SSL, the port number will most likely need to be changed - the default SMTP SSL port is 465.

The list of current notifications is displayed in the Email Notification tab. Use *Remove* to delete notifications, and *Add* to create a new notification.

Enter an appropriate name for the notification, and select what Events the notification is applicable for. The various fields are described below:

## Events

Processes may be triggered by any of the types of events listed below. By default, events are only triggered when an operation, such as a file upload, succeeds, but they may also be triggered on failure. This is determined by the checkboxes labelled 'Trigger on success' and 'Trigger on error'.

### Upload file

When a file has been uploaded. All macros are available.

### Download file

When a file has been downloaded. All macros are available.

### Move file

When a file has been renamed or moved. All macros are available.

### Delete file

When a file has been deleted. All macros are available.

### Create folder

When a new folder has been created by a user. All macros are available.

### Move folder

When a folder has been renamed or moved by a user. All macros are available.

### Delete folder

When a folder has been deleted by a user. All macros are available.

### Log in

When a user logs in. All macros are available.

### Log out

When a user logs out. All macros are available.

### Server start

When the CompleteFTP Windows service starts. By default, the service is configured to restart automatically in case of failure, so this event can be useful to alert administrators of such failures.

### Server stop

When the CompleteFTP Windows service stops gracefully. This event will not trigger if the service stops ungracefully.

### Site start

[Enterprise MFT only] When a site starts.

### Site stop

[Enterprise MFT only] When a site stops.

#### Auto-ban

When an IP address has been auto-banned.

#### SSL/TLS certificate expiring

When a user tries to connect via FTPS or HTTPS and the server certificate is due to expire in less than 30 days. Note that this event is triggered at most once every 24 hours or after the CompleteFTP service has been restarted.

#### SSL/TLS certificate expired

When a user tries to connect via FTPS or HTTPS and the server certificate has expired. Note that this event is triggered at most once every 24 hours or after the CompleteFTP service has been restarted.

#### Cluster synchronization failed

[Enterprise MFT only] When the primary fails to synchronize with a secondary.

## Filters

Filters restrict events based on a set of criteria that are defined by the administrator. These criteria may be based on: (1) the folder in which a file-operation is taking place, (2) the user performing the operation, and/or (3) the site on which the operation is taking place.

### Folder filter

The folder- or file-path filter may be used to select the files/folders for which the event should be triggered. By default Windows-style wildcards are used for matching. For example, if the event should be triggered only for files with the extension `.dat` in the folder `/Home/MyUser` then the filter should be set to `/Home/MyUser/*.dat`. If the event should be triggered for files named `test.dat` in any directory, then the filter should be set to `*/test.dat`.

Filters may be inverted by checking the checkbox labelled 'Inverse'. If this is done then the event will occur for all files/folders that **don't match** the filter.

Regular expressions (.NET-style) may also be used. To use these, the filter should be prefixed by 'regex:'. Information on .NET regular expressions may be found on the Microsoft(tm) website. Note that multiple filters may be defined using regular expressions using the syntax, `regex:^(/Folder1/.*/Folder2/.*)`, which would match anything underneath the folders, `/Folder1` and `/Folder2`.

### Example

To test this filter to see if it can filter the files/folders whose name has only got a number in it.

1. Under folder `user1`, create 2 folders: `abc` and `123`.
2. Folder filter: `regex:^(/Home/user1/[0-9]/*)`.
3. Upload any file to folder `123` (or its subfolder - `123/456`), the email will be sent.
4. Upload a file to `abc` or its subfolder, no email will be sent because this `abc` folder is not match the `/Home/user1/[0-9]/*` filter.

The main point here is it check the folder path, so that without the path (`/Home/user1/...`), it won't work.

User can change the `[0-9]` to `[a-z]` to see the opposite result of 3. and 4.

### User filter

A user filter may be used to select the users for which an event should occur, or, if it's inverted, the users for which it shouldn't occur. By default it will occur for all users. Users may be selected simply by checking the box next to them in the dropdown list that's displayed, when the user filter control is clicked.

### Example

How to test this filter so that it can be used to filter for the user, 'user1' only:

1. User filter: set as `user1`.
2. Login to server via FileZilla using the user, 'user1', and upload a file.
3. Check email, there should be an email sent and received.
4. Continue to login to the server with FileZilla using user, 'user2', and upload a file.
5. Check email again, no email should have been sent.

### Site filter

[Enterprise MFT only] A site filter may be used to select the sites for which an event should occur or, if it's inverted, the sites on which it shouldn't occur. By default it will occur on all sites. Sites may be selected simply by checking the box next to them in the dropdown list that's displayed when the site filter control is clicked.

### From address

Enter the email address that the email notification is set to originate from. The `%OwnerEmail%` (email address of folder owner) or `%LoginEmail%` (email address of logged-in user) macros may be used here.

### To address

Enter the email address that the email notification is sent to. The `%OwnerEmail%` (email address of folder owner) or `%LoginEmail%` (email address of logged-in user) macros may be

used here.

### **Subject**

Enter the subject of the notification message. [Various macros](#) can be used here.

### **Message**

Enter the body of the notification message. [Various macros](#) can be used here.

For an example on setting up an email notification, please refer to Step-by-step guide: [E-mail notification when a file is uploaded](#).

# How to set up process execution triggers

CompleteFTP supports highly customizable, process execution triggers. This means that when an event such as a file upload occurs, a process can be launched (e.g. to move a file after it has been uploaded).

The screenshot shows the 'Process Execution' configuration window in CompleteFTP. On the left is a list of triggers with columns 'Enabled' and 'Name'. The first trigger, 'Copy uploaded files', is checked and highlighted. Above this list are buttons for 'Add', 'Remove', and 'Copy'. Below the list are 'Move up' and 'Move down' buttons. The main configuration area on the right is for the selected trigger. It includes a 'Name' field with 'Copy uploaded files'. The 'Events' section has a checked 'Upload file' event with a 'Choose' button. The 'Errors' section has 'Trigger on success' checked and 'Trigger on error' unchecked. The 'Folder filter' is '/Upload/\*' with an 'Inverse' checkbox. The 'User filter' is 'Mary' and the 'Site filter' is 'Default Site', both with 'Inverse' checkboxes. The 'Type' section has radio buttons for 'Program', 'Batch script', 'Powershell script', 'JSS script', and 'FTP script' (which is selected). The 'Script' text area contains: '# Copy all files uploaded to /Uploads to other SFTP server', 'set remotehost=otherftpserver.com', 'set user=myusername', 'set password=mypassword', 'set protocol=sftp', and '# connect to the server'. The 'Directory' field is empty with a browse button. The 'Limits' section has 'Max processes' set to 200 and 'Max run time' set to 0 seconds. The 'Working directory' section has a text area with the instruction: 'Full path of working directory to use when the program is being executed. If it's left blank then the program's directory will be used.'

## Process control

In the Process Execution tab, the maximum number of concurrent processes (default 200) and the maximum run time can be set. The latter setting has a default of 0 which means no limit.

## Creating and editing triggers

The list of current process triggers is displayed in the Process Execution tab. Use *Remove* to delete triggers, and *Add* to create a new trigger.

When adding, enter an appropriate name for the trigger, and select which Events the trigger is applicable for. The various fields are described below:

### Events

Processes may be triggered by any of the types of events listed below. By default, events are only triggered when an operation, such as a file upload, succeeds, but they may also be triggered on failure. This is determined by the checkboxes labelled 'Trigger on success' and 'Trigger on error'.

#### Upload file

When a file has been uploaded. All macros are available.

#### Download file

When a file has been downloaded. All macros are available.

#### Move file

When a file has been renamed or moved. All macros are available.

#### Delete file

When a file has been deleted. All macros are available.

#### Create folder

When a new folder has been created by a user. All macros are available.

#### Move folder

When a folder has been renamed or moved by a user. All macros are available.

#### Delete folder

When a folder has been deleted by a user. All macros are available.

#### Log in

When a user logs in. All macros are available.

#### Log out

When a user logs out. All macros are available.

#### Scheduled

Events occur at a predefined schedule, such as once per hour or at midnight every weekday. See [here](#) for details. Since this event is not related to any particular session macros related to users and files are not available.

#### Server start

When the CompleteFTP Windows service starts.

#### Server stop

When the CompleteFTP Windows service stops gracefully. This event will not trigger if the service stops ungracefully.

#### Site start

[Enterprise MFT only] When a site starts.

#### Site stop

[Enterprise MFT only] When a site stops.

#### Auto-ban

When an IP address has been auto-banned.

#### SSL/TLS certificate expiring

When a user tries to connect via FTPS or HTTPS and the server certificate is due to expire in less than 30 days. Note that this event is triggered at most once every 24 hours or after the CompleteFTP service has been restarted.

#### SSL/TLS certificate expired

When a user tries to connect via FTPS or HTTPS and the server certificate has expired. Note that this event is triggered at most once every 24 hours or after the CompleteFTP service has been restarted.

#### Cluster synchronization failed

[Enterprise MFT only] When the primary fails to synchronize with a secondary.

### Filters

Filters restrict events based on a set of criteria that are defined by the administrator. These criteria may be based on (1) the folder in which a file-operation is taking place, (2) the user performing the operation, and/or (3) the site on which the operation is taking place.

#### Folder filter

The folder- or file-path filter may be used to select the files/folders for which the event should be triggered. By default Windows-style wildcards are used for matching. For example, if the event should be triggered only for files with the extension `.dat` in the folder `/Home/MyUser` then the filter should be set to `/Home/MyUser/* .dat`. If the event should be triggered for files named `test.dat` in any directory, then the filter should be set to `*/test.dat`.

Filters may be inverted by checking the checkbox labelled 'Inverse'. If this is done then the event will occur for all files/folders that **don't match** the filter.

Regular expressions (.NET-style) may also be used. To use these, the filter should be prefixed by 'regex'. Information on .NET regular expressions may be found on the Microsoft(tm) website. Note that multiple filters may be defined using regular expressions using the syntax, `regex:^(/Folder1/.*/Folder2/.*)`, which would match anything underneath the folders, `/Folder1` and `/Folder2`.

#### Example

To test that this filter can filter the files/folder whose name only has a number in it:

1. Under folder `user1`, create 2 folders: `abc` and `123`.
2. Folder filter: `regex:^(/Home/user1/[0-9]/*)`.
3. Upload any file to folder `123` (or its subfolder - `123/456`), the email will be sent.
4. Upload a file to `abc` or its subfolder, no email will be sent because this `abc` folder is not match the `/Home/user1/[0-9]/*` filter.

The main point here is it checks the folder path, so without the path (`/Home/user1/...`), it won't work.

User can change the `[0-9]` to `[a-z]` to see the opposite result of steps 3. and 4. above.

#### User filter

A user filter may be used to select the users for which an event should occur, or if it's inverted, the users for which it shouldn't occur. By default it will occur for all users. Users may be selected simply by checking the box next to them in the dropdown list, which is displayed when the user filter control is clicked.

#### Example

To test that this filter can filter for the user 'user1' only:

1. User filter: set as `user1`.
2. Login to server via FileZilla using user 'user1' and upload a file.
3. Check email, as there is an email sent and received.
4. Continue to login to the server with FileZilla using user 'user2' and upload a file.
5. Check email again, no email is sent.

#### Site filter

[Enterprise MFT only] A site filter may be used to select the sites for which an event should occur or, if it's inverted, the sites on which it shouldn't occur. By default it will occur on all sites. Sites may be selected simply by checking the box next to them in the dropdown list, which is displayed when the site filter control is clicked.

### Type

CompleteFTP supports five types of process triggers:

#### Program

Execute a specified executable. The full path of the executable must be set in the 'Program file' field. Any required arguments may be specified in the 'Arguments' field. These may include any of the macros displayed when the button to the right of the field is clicked (e.g. %FileName%, %LoginUserName%). The working directory may be specified in the 'Directory' field. Note that if there's a possibility that there's a space character in a macro variable then the macro should be in quotes.

#### Batch script

Batch scripts use the common DOS/Windows batch commands. Scripts should be entered directly into the 'Script' field. **Various macros** may be entered manually or inserted by selecting them from the menu which is displayed when the button to the right of the 'Script' field is clicked (e.g. %FileName%, %LoginUserName%). The working directory may be specified in the 'Directory' field. Note that if there's a possibility that there's a space character in a macro variable, then the macro should be in quotes. If an existing batch file is to be executed then the 'Program' type should be used with the path of the batch file as the 'Program file' argument.

#### Powershell script

Powershell scripts use the standard Microsoft Powershell syntax. Scripts should be entered directly into the 'Script' field. **Various macros** may be entered manually or inserted by selecting them from the menu which is displayed when the button to the right of the 'Script' field is clicked (e.g. %FileName%, %LoginUserName%). The working directory may be specified in the 'Directory' field. Note that if there's a possibility that there's a space character in a macro variable then the macro should be in quotes. If an existing Powershell file is to be executed then the 'Program' type should be used, with the path of the batch file as the 'Program file' argument.

#### JSS script

JSS (Javascript Server-Side) is Javascript combined with a set of API's for common tasks and integration with CompleteFTPISS scripts are executed within the CompleteFTI service.

The **JSS API's** provide access to CompleteFTP's **virtual file-system** and commonly used functionality, such as **FTP/SFTP**, **HTTP**, **e-mail** and **databases**. Various **macros** are available within a JSS process trigger script via properties of the variable called `event`. These can be inserted into the script via the button to the right of the editor.

JSS permits use of .NET classes (e.g. `var xmlDoc = new System.Xml.XmlDocument()`) so that the standard .NET libraries are available if required.

#### FTP script

FTP scripts are a sequence of commands similar to those used in command-line FTP applications (e.g. GET and PUT), preceded by a set of commands to set connection parameters. Full details may be found [here](#). Note that JSS includes an FTP/SFTP client (see [here](#)) so, since Javascript allows proper programming as opposed to just a fixed sequence of commands, in many cases using JSS scripts is recommended above that of FTP scripts.

For an example on setting up a process trigger, please refer to Step-by-step guide: [Move a file to another directory after it's been uploaded](#).



# Scheduled Events

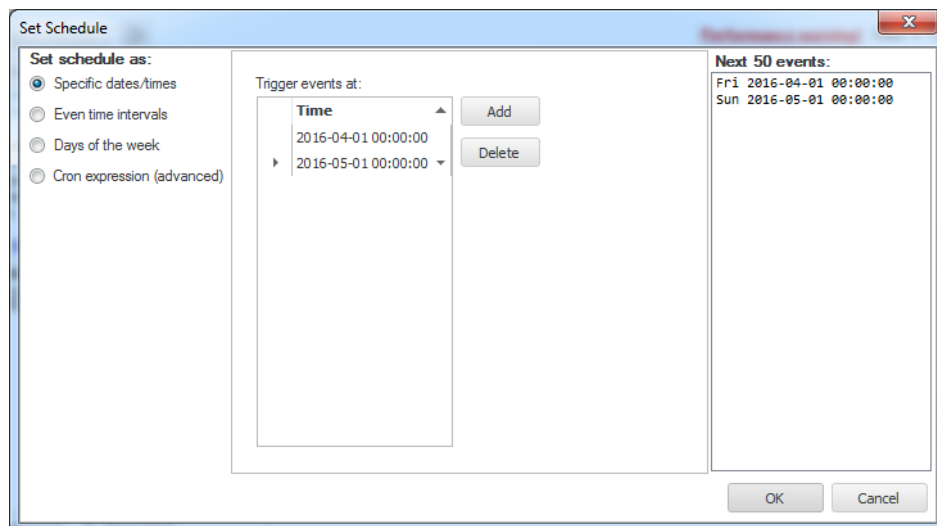
CompleteFTP allows events to be triggered according to a user-specified schedule. This schedule may consist of a sequence of specific times or of a repeated pattern. There are three ways to define a pattern: by even time intervals, by days of the week or by a [cron expression](#).

Schedules may be defined using the Set Schedule dialog. This is accessed on the Process Triggers panel by selecting the *Change* link next to the "Scheduled" checkbox of the event.

The four methods of setting a schedule are listed on the left. When one of these is selected the editor for that method will be displayed in the centre. On the right is a list of the next (up to) 50 times that the defined event will be triggered.

## Specific times

A schedule of specific times is defined by creating a list of dates and times using the following interface:

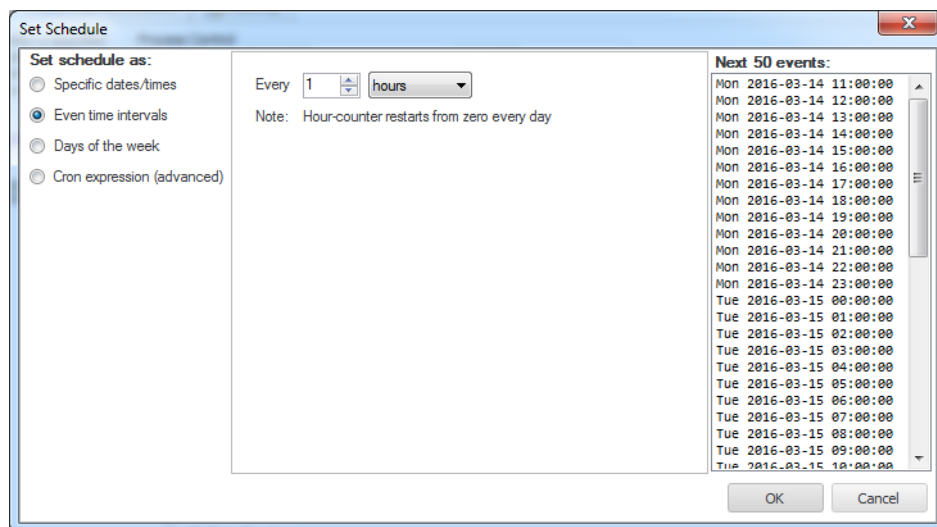


The 'Set Schedule' dialog box shows the 'Specific dates/times' method selected. The 'Trigger events at:' list contains two entries: '2016-04-01 00:00:00' and '2016-05-01 00:00:00'. The 'Next 50 events:' list shows the next two events: 'Fri 2016-04-01 00:00:00' and 'Sun 2016-05-01 00:00:00'.

The 'Next 50 events' list will only list future events and a warning will be shown when OK is pressed if there are no future events.

## Even time intervals

This is the easiest way to define an event that is to be triggered every  $n$  seconds, minutes, hours, days or months.



The 'Set Schedule' dialog box shows the 'Even time intervals' method selected. The 'Every' field is set to '1' and the unit is 'hours'. A note states: 'Hour-counter restarts from zero every day'. The 'Next 50 events:' list shows a sequence of events starting from 'Mon 2016-03-14 11:00:00' to 'Tue 2016-03-15 10:00:00'.

It's important to note that the counter restarts from zero when it passes the maximum value of the particular unit that is selected. For example, if 23 minutes is selected and the time is currently 13:11 then the event will trigger at 13:23, then 13:46 and then again at 14:00. It's therefore best to use factors of the maximum value, such as 30 seconds, 15 minutes or 6 hours.

## Days of the week

Events may be specified to trigger at a certain time on specific days of the week. For example, at noon on weekdays:

Set Schedule

Set schedule as:

- Specific dates/times
- Even time intervals
- Days of the week**
- Cron expression (advanced)

At 12:00:00

On

- ☒ Mondays
- ☒ Tuesdays
- ☒ Wednesdays
- ☒ Thursdays
- ☒ Fridays
- ☐ Saturdays
- ☐ Sundays

Next 50 events:

Mon	2016-03-14	12:00:00
Tue	2016-03-15	12:00:00
Wed	2016-03-16	12:00:00
Thu	2016-03-17	12:00:00
Fri	2016-03-18	12:00:00
Mon	2016-03-21	12:00:00
Tue	2016-03-22	12:00:00
Wed	2016-03-23	12:00:00
Thu	2016-03-24	12:00:00
Fri	2016-03-25	12:00:00
Mon	2016-03-28	12:00:00
Tue	2016-03-29	12:00:00
Wed	2016-03-30	12:00:00
Thu	2016-03-31	12:00:00
Fri	2016-04-01	12:00:00
Mon	2016-04-04	12:00:00
Tue	2016-04-05	12:00:00
Wed	2016-04-06	12:00:00
Thu	2016-04-07	12:00:00
Fri	2016-04-08	12:00:00
Mon	2016-04-11	12:00:00
Tue	2016-04-12	12:00:00
Wed	2016-04-13	12:00:00
Thu	2016-04-14	12:00:00

OK

Cancel

## Cron expression (advanced)

If none of the simpler methods of defining schedules cover your requirements then cron expressions probably will. Cron expressions are a powerful concept often used in Unix-like operating systems. Please refer to [this Wikipedia article](#) for an explanation. Note however that CompleteFTP's cron expressions also includes seconds.

The interface for specifying cron expressions allows you to select a common pattern from the list at the top. If this doesn't meet your requirements then it may be customized by modifying each of the components in turn. The button to the right of each component text-field allows you to select from a set of common values for that field. You may also edit the cron expression directly in the text-field labelled 'Cron pattern'. The field at the bottom provides a verbal description of the current cron expression.

Set Schedule

Set schedule as:

- Specific dates/times
- Even time intervals
- Days of the week
- Cron expression (advanced)**

First choose a common pattern
Common pattern: Custom pattern

Then modify the pattern

Second: 0

Minute: 0

Hour: 12,18

Day: \*

Month: \*

Weekday: 1,2,3,4,5

You can also enter the pattern directly in cron format
Cron pattern: 0 0 12,18 \* \* 1,2,3,4,5

Double-check with a verbal description
At 12:00 PM and 6:00 PM, only on Monday, Tuesday, \

Next 50 events:

Mon	2016-03-14	12:00:00
Mon	2016-03-14	18:00:00
Tue	2016-03-15	12:00:00
Tue	2016-03-15	18:00:00
Wed	2016-03-16	12:00:00
Wed	2016-03-16	18:00:00
Thu	2016-03-17	12:00:00
Thu	2016-03-17	18:00:00
Fri	2016-03-18	12:00:00
Fri	2016-03-18	18:00:00
Mon	2016-03-21	12:00:00
Mon	2016-03-21	18:00:00
Tue	2016-03-22	12:00:00
Tue	2016-03-22	18:00:00
Wed	2016-03-23	12:00:00
Wed	2016-03-23	18:00:00
Thu	2016-03-24	12:00:00
Thu	2016-03-24	18:00:00
Fri	2016-03-25	12:00:00
Fri	2016-03-25	18:00:00
Mon	2016-03-28	12:00:00
Mon	2016-03-28	18:00:00
Tue	2016-03-29	12:00:00
Tue	2016-03-29	18:00:00

OK

Cancel

For an example on creating a process trigger for scheduled event, please refer to Step-by-step guide: [Add a scheduled task](#).

# How to write FTP scripts

FTPScript is a powerful, yet simple scripting language specifically for FTP, FTPS and SFTP.

## Command overview

Most commands are fairly self explanatory, and are described in the [script command reference](#).

The set command is used to set various parameters that are used during the session, such as username, password, timeout and so on. Many of these parameters are optional (e.g. port, timeout). The most important parameter is the protocol.

Various macros can be inserted into the script at execute time.

The local working directory may be set in the process trigger definition inside the CompleteFTP Manager, or it may be set using the localdir command inside the script. All uploads and downloads that do not specify a full path, will use the set local working directory.

## Sample code

A sample script is shown below:

```
# Sample SFTP script for uploading a file to another server
set remotehost=otherftpserver.com
set user=myusername
set password=mypassword
set protocol=sftp

# connect to the server
open

# upload the file for which the trigger occurred to the server
# note quotes around path as it may contain spaces
put "%WindowsPath%" %FileName%

# disconnect
close
```

# Script command reference

Scripting and shell commands are described in the table below. Abbreviations for commands are noted in the command description.

Where the command has optional arguments, they are shown in square brackets, e.g. *open [host[:port]]* means that for the open command, the host and port are optional parameters (but if the port is supplied, the host must be supplied).

command		Description												
help [command]		Prints out help for a specific command, or if no command is specified, a list of supported commands is provided.												
version	Prints out version details.  Can use <i>ver</i> as an abbreviation.													
debug	Switches debug on or off. If debug is already on, it is switched off (and vice versa). When debug is enabled, more information is displayed when commands are executed.													
set	Sets various parameters used during the session, many of which are optional. Syntax is <code>set name=value</code>  Note that 'set' can be omitted, and thus <code>protocol=ftp</code> is a valid command.  Variables can be created this way. If an unknown variable is set using the 'set' command, then a variable of that name and value is created, e.g. <code>myprotocol=ftp</code> creates a variable called 'myprotocol' set to the string value 'ftp'.  The value of variables can be printed out using <i>echo</i> .  The value of all set variables can be printed out using <i>set</i> without any arguments.  The set command also supports a feature originally found on Unix systems called backquoting. This is where a variable is assigned the value of the output of a command, which must be surrounded by back quotes (the <code>"</code> symbol).  For example, to set the variable 'currentdir' to the value of the current remote working directory, use <code>set currentdir=`pwd`</code> .													
set connectmode	Sets the connection mode for data transfers and listings. Not applicable for SFTP protocol. Possible values are <i>active</i> or <i>passive</i> ., e.g. <code>set connectmode=active</code> .													
set connectmode	Sets the connection mode for data transfers and listings. Not applicable for SFTP protocol. Possible values are <i>active</i> or <i>passive</i> ., e.g. <code>set connectmode=active</code> .													
set protocol	Set the protocol to be used. Note that the protocol cannot be changed while connected. Supported protocols are shown below: <table><tr><th>Protocol</th><th>Description</th></tr><tr><td>ftp</td><td>"Plain" FTP protocol</td></tr><tr><td>ftps</td><td>FTPS, i.e. FTP over SSL (explicit mode)</td></tr><tr><td>ftpse</td><td>FTPS, i.e. FTP over SSL (explicit mode)</td></tr><tr><td>ftpsi</td><td>FTPS, i.e. FTP over SSL (implicit mode)</td></tr><tr><td>sftp</td><td>SFTP, i.e. FTP over SSH</td></tr></table> Note that server validation is switched off by default.		Protocol	Description	ftp	"Plain" FTP protocol	ftps	FTPS, i.e. FTP over SSL (explicit mode)	ftpse	FTPS, i.e. FTP over SSL (explicit mode)	ftpsi	FTPS, i.e. FTP over SSL (implicit mode)	sftp	SFTP, i.e. FTP over SSH
Protocol	Description													
ftp	"Plain" FTP protocol													
ftps	FTPS, i.e. FTP over SSL (explicit mode)													
ftpse	FTPS, i.e. FTP over SSL (explicit mode)													
ftpsi	FTPS, i.e. FTP over SSL (implicit mode)													
sftp	SFTP, i.e. FTP over SSH													
set connectmode	Sets the connection mode for data transfers and listings. Not applicable for SFTP protocol. Possible values are <i>active</i> or <i>passive</i> ., e.g. <code>set connectmode=active</code> .													
set remotehost	Sets the remote host to be connected to, e.g. <code>set remotehost=ftp.gnu.org</code>													
set user	Sets the username, e.g. <code>set user=javaftp</code> . This username is used for all supported protocols.													
set password	Sets the user's password, e.g. <code>set password=javaftp</code> . This password is used for all supported protocols.													
set localdir	Sets the local working directory. All uploads and downloads that do not specify a full path will use this directory. e.g. <code>set localdir=D:\work\tmp</code>													
set servervalidation	Enables or disables server validation (disabled by default). Possible values are <i>true</i> or <i>false</i> .													
set clientcert	Sets the path of the client certificate to use for client validation for FTPS. Currently supports PEM, CER and PFX file formats for certificates. Note that the password must also be set via <i>certpassword</i> .  Once set, client validation is automatically switched on.													
set certpassword	Sets the password (or passphrase) of the client certificate used for client validation (FTPS).													
set serverpublickey	Sets the path of the server's public key to use for server validation for SFTP. Supports OpenSSH and SECSH file formats. More typically <code>set knownhosts</code> is used.  Can also use <i>serverkey</i> .  Server validation must also be switched on via <code>set servervalidation=true</code>													
set knownhosts	Sets the path of the <i>known_hosts</i> file to use for server validation for SFTP.  Server validation must also be switched on via <code>set servervalidation=true</code>													
set clientprivatekey	Sets the path of the client private key file to be used for public key authentication in SFTP. Supports PuTTY, OpenSSH and SSH.com private key formats. Note that the <i>clientprivatekeypass</i> must also be set.  Can also use <i>clientkey</i> .													
set clientprivatekeypass	Sets the passphrase of the client private key file to be used for public key authentication in SFTP.  Can also use <i>clientkeypass</i> .													
set timeout	Optional. Sets the timeout in seconds for read or write operations. The default is 0 (which is an infinite timeout). e.g. <code>set timeout=10</code>													
set port	Optional. Defaults are determined by the protocol. Sets the port number to connect to, e.g. <code>set port=21</code>													
set dataports	Optional. Only applicable for active mode. Sets the data port range that the client will listen on for listings and transfers. The low port number must be listed first, e.g. <code>set dataports=10100-10140</code>													
set loglevel	Optional. Sets the logging level. 0 means no logging, while 5 is the maximum log level (the default). e.g. <code>set loglevel=4</code>													
set logfile	Sets the name or full path of the logfile to direct logging to, e.g. <code>set logfile=C:\temp\ftp.log</code>													
echo	Echoes the value of a variable to standard output, whether user-defined or a system variable such as 'protocol'. The variable must be preceded by a '\$' to identify it as a variable, e.g. <code>echo \$protocol</code> .  A synonym for echo is <i>print</i> .													
open [host[:port]]	Opens the connection to the remote server. The user and password must already be set via the set command. If the remote host is not already set, it must be supplied here (and will override any previous setting for this connection attempt). The port number can also be optionally supplied (but the host must be provided). e.g. <code>open edtmobile:21</code>  Can also use connect.													
binary	Change the current transfer mode to binary mode. Binary is the default.  Can use <i>bin</i> as an abbreviation.													

<b>ascii</b>	Change the current transfer mode to ASCII mode. Binary is the default.  Can use <i>asc</i> as an abbreviation.
<b>pwd</b>	Print the current remote working directory.
<b>dir [remotedir]</b>	Perform a detailed listing of the current directory if no directory is supplied as an argument, or of the supplied remote directory.  See <i>ls</i> for a similar command. A synonym of <i>dir</i> is <i>ls -l</i> .
<b>ls [remotedir]</b>	Perform a simple listing (just filenames) of the current directory if no directory is supplied as an argument, or of the supplied remote directory. If <i>ls -l</i> is used, a detailed listing is supplied.  See <i>dir</i> for a similar command. A synonym of <i>ls -l</i> is <i>dir</i> .
<b>cd remotedir</b>	Change the current remote working directory to <i>remotedir</i> .
<b>cdup</b>	Change the current remote working directory to its parent directory.
<b>mkdir remotedir</b>	Create the remotedir directory.
<b>rmdir remotedir</b>	Delete the remotedir directory. It must be empty.
<b>get remotefile [localfile]</b>	Download a remotefile to the local host. The localfile parameter can be a filename or the full path of a local file. If it is a filename, the current local working directory is prepended.  Note that the localfile parameter is optional - if not supplied, the downloaded file is saved in the current local working directory with the name of remotefile.  The remotefile parameter can be a filename or a path. Not all servers support the use of a path - in this case navigate to the correct remote directory using <i>cd</i> .
<b>put localfile [remotefile]</b>	Upload a local file to the remote host. The localfile parameter can be a filename or the full path of a local file. If it is a filename, the current local working directory is prepended.  The remotefile parameter should be a filename or a path. Not all servers support the use of a path - in this case navigate to the correct remote directory using <i>cd</i> .  If the remotefile parameter is not supplied, the local file name will be used.
<b>append localfile [remotefile]</b>	Append the contents of a local file to a remote file (if it exists) on the remote host. The localfile parameter can be a filename or the full path of a local file. If it is a filename, the current local working directory is prepended.  The remotefile parameter should be a filename or a path. Not all servers support the use of a path - in this case navigate to the correct remote directory using <i>cd</i> .  If the remotefile parameter is not supplied, the local file name will be used.
<b>rename remotefile1 remotefile2</b>	Rename a remote file from remotefile1 to remotefile2. Note that both parameters should be filenames, not full paths.  Can use <i>ren</i> as an abbreviation.
<b>lrename localfile1 localfile2</b>	Rename a local file from localfile1 to localfile2. Note that paths can be used as well as filenames.  Can use <i>lren</i> as an abbreviation.
<b>delete remotefile</b>	Delete a remote file in the current remote working directory.  Can use <i>del</i> as an abbreviation.
<b>ldelete localfile</b>	Delete a local file in the current local working directory, or specify a path for the file to be deleted.  Can use <i>ldel</i> as an abbreviation.
<b>mdelete wildcard</b>	Delete multiple remote files that match the wildcard in the current remote working directory.  Wildcards supported are '?' for a single matching character, and '*' for multiple matching characters. e.g. to delete all text files in the current remote directory, use <i>mdel *.txt</i>  Can use <i>mdel</i> as an abbreviation.
<b>mput wildcard</b>	Upload multiple local files that match the wildcard in the current local working directory to the current remote working directory.  Wildcards supported are '?' for a single matching character, and '*' for multiple matching characters. e.g. to upload all text files from the current local working directory, use <i>mput *.txt</i>
<b>mget wildcard</b>	Download multiple local files that match the wildcard in the current remote working directory to the current local working directory.  Wildcards supported are '?' for a single matching character, and '*' for multiple matching characters. e.g. to download all text files from the current remote working directory use <i>mget *.txt</i>
<b>close</b>	Closes the connection to the remote server.  Can also use <i>bye</i> , <i>quit</i> , <i>exit</i> or <i>disconnect</i> .
<b>quote command</b>	Sends the quoted command to the server and displays the response.
<b>site parameters</b>	Sends a SITE command to the FTP server with the supplied parameters. These parameters can vary widely between FTP servers. This command is not applicable to SFTP servers.  For example <i>site CHMOD 0600 /home/user/privatefile</i> might be used to change the permissions of a file on an FTP server that supports this feature.
<b>auth</b>	Change from unencrypted 'plain' FTP into secure FTP.
<b>protect clear or protect private</b>	Change the protection level of the data channel in FTPS.  Supported options are <i>clear</i> (or <i>c</i> ) to set data channels to unencrypted, and <i>private</i> (or <i>p</i> ) to set data channels to encrypted. e.g. <i>protect clear</i>  Alternatively <i>private</i> has the same effect as <i>protect private</i> , and <i>clear</i> has the same effect as <i>protect clear</i> .  An abbreviation for <i>protect</i> is <i>prot</i> , which combined with the argument abbreviations means that to set the data channels to unencrypted could be done with <i>prot c</i> , and to set to encrypted could be done with <i>prot p</i> .
<b>clear</b>	In FTPS, change the protection level of the data channel to private. Identical to <i>protect private</i> .
<b>private</b>	In FTPS, change the protection level of the data channel to private. Identical to <i>protect private</i> .
<b>ccc</b>	In FTPS, clears the control channel, setting it back to plain unencrypted text. This can be useful when using firewalls, which need to be able to inspect the control channel to open data channel ports.  Once <i>ccc</i> has been called, no more <i>protect</i> commands can be used. This is a security measure inherent in the protocol.
<b>record recordfile</b>	Record subsequent commands to the supplied filename. Most useful from within the shell to record a session.
<b>stop</b>	Stop recording commands to file.

# Notification macros

Email notifications and process triggers support a variety of macros. Email notifications can use these macros in the *Subject* and *Message* fields, while process triggers can use them in scripts or program arguments. The macros are described in the table below.

Macro	Description
%FilePath%	Full path of file in the virtual file-system
%FileName%	Name of file
%FolderPath%	Path to folder in the virtual file-syste,
%WindowsPath%	Path of file in the Windows file-system
%WindowsFolder%	Path to folder in the Windows file-system
%WindowsFileName%	Name of file in Windows
%OwnerEmail%	Email address of owner of folder
%OwnerUserName%	Usemame of owner of folder
%OwnerFullName%	Full name of owner of folder
%LoginEmail%	Email address of logged-in user
%LoginUserName%	Usemame of logged-in user
%LoginFullName%	Full name of logged-in user
%Event%	Event that caused the notification
%Notification%	Name of this notification
%PathFilter%	File filter of this notification
%SiteName%	Name of site
%ClientIP%	IP address of client
%TransferStatus%	Indicates whether or not an operation succeeded. If it succeeded the value will be 'success', otherwise it will be 'failure'.
%Time%	Time of event
%Date%	Date of event
%DateAndTime%	Date/time of event
%Timestamp:format%	Formatted timestamp of event. The word 'format' should be replaced by a .NET date/time formatting string (see <a href="#">here</a> and <a href="#">here</a> ). For example, the macro %Timestamp:yyyy/MM/dd HH:mm:ss% would produce a string like "2012-02-08 16:33:21".

## How to write your own events (Enterprise MFT only)

CompleteFTP supports email and process-trigger event handlers. Event extension classes can be written so that custom events are invoked when particular events occur. For example, it may be desirable to implement a custom auditing system that logs events to a proprietary database. See [event extensions](#) for more details.

# How to write Batch Scripts

A batch file is a plaintext file or script file, which contains multiple commands to achieve a certain task. It contains a series of commands that is executed by a command line interpreter. Some basic **commands** of batch files:

- **echo** â€“ Prints out the input string. It can be ON or OFF, allowing ECHO to turn the echoing feature on or off. If ECHO is ON, the command prompt will display the command it is executing.
- **copy** â€“ Copy a file or files.
- **move** â€“ Move a file or files.
- **del** â€“ Delete a file or files.
- **mkdir** â€“ Create a directory.
- **rmdir** â€“ Delete a directory.
- **if** â€“ Conditional statement.

A batch script can be used in process triggers in the Events panel of the CompleteFTP Manager, to check that the information of the events is saved in the created file in the script.

## Sample code

This script checks that the Move file event works properly with a Batch script type.

```
:: Create a BatchScript_Result.txt file in C:\Temp which contains information of the newly moved file
echo File '%FileName%' has been moved to another directory at %Time% > "C:\Temp\BatchScript_Result.txt"
```

## Steps to apply the above script:

1. Go to the **Process Triggers** tab, and add a new process trigger for a Move file event.

2. Configure the process trigger as follows:

- Errors: Trigger on success.
- Folder filter: no restriction (\*).
- User filter: Select all.
- Site filter: Select all.
- Type: Batch script.
- Directory: leave it blank.
- Limits: leave it as default.

3. Input the **above** script with Batch script type, then **Apply change**.

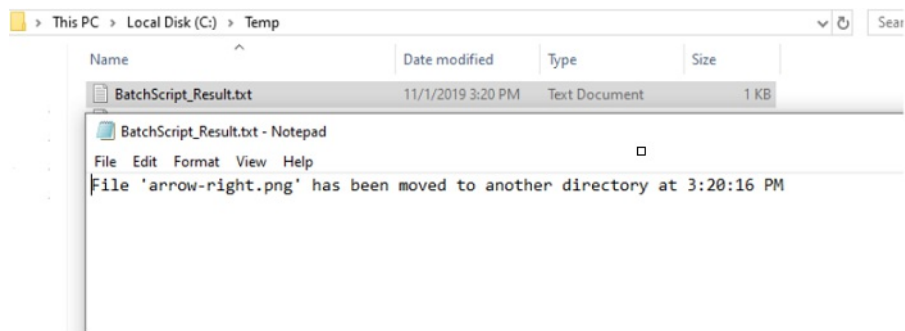
The screenshot shows the 'New process trigger' configuration window in CompleteFTP Manager. The 'Name' field is 'New process trigger 1'. The 'Events' section has 'Move file' selected. The 'Errors' section has 'Trigger on success' selected. The 'Folder filter' is '\*' with 'Inverse' unchecked. The 'User filter' is '(All users)' with 'Inverse' unchecked. The 'Site filter' is '(All sites)' with 'Inverse' unchecked. The 'Type' section has 'Batch script' selected. The 'Script' field contains the batch script code. The 'Directory' field is empty. The 'Limits' section shows 'Max processes' as 200 and 'Max run time' as 0 sec.

Name:	New process trigger 1		
Events:	<input checked="" type="checkbox"/> Move file	<input type="button" value="Choose"/>	
Errors:	<input checked="" type="checkbox"/> Trigger on success	<input type="checkbox"/> Trigger on error	
Folder filter:	*	<input type="button" value="Invert"/>	<input type="checkbox"/> Inverse
User filter:	(All users)	<input type="button" value="Invert"/>	<input type="checkbox"/> Inverse
Site filter:	(All sites)	<input type="button" value="Invert"/>	<input type="checkbox"/> Inverse
Type:	<input type="radio"/> Program <input checked="" type="radio"/> Batch script <input type="radio"/> Powershell script <input type="radio"/> JSS script <input type="radio"/> FTP script		
Script:	<pre>:: Create a BatchScript_Result.txt file in C:\Temp which contains information of the newly moved file echo File '%FileName%' has been moved to another directory at %Time% &gt; "C:\Temp\BatchScript_Result.txt"</pre>		
Directory:			
Limits:	Max processes: 200 Max run time: 0 sec (0=no limit)		

4. From the FTP client, connect to the CompleteFTP server with user`user1` and either move a file to another folder, or rename a file -> Ensure that the user is logged in successfully and the file is moved/rename successfully.

5. Navigate to C:\Temp and ensure that there is a BatchScript\_Result.txt file, which contains the information of the newly moved file.





# How to write PowerShell scripts

PowerShell is a command-line tool which allows you to run commands and scripts, to change system settings and automate tasks.

It can, for example, be used in the Process Triggers tab of the Events panel of the CompleteFTP Manager, to check that the information of the events is saved in the created file in the script.

## Sample code:

This script checks that the Upload file event works properly, with the PowerShell script type.

```
# Create a PowerShell_Result.txt file in C:\Temp contains the information of the newly uploaded file
$path = 'C:\Temp\PowerShell_Result.txt'
$value = '%FileName% has been uploaded by %LoginUserName% at %Time%'

Add-Content -path $path -Value $value
```

## Steps to apply script above

1. Go to the "Process Triggers" tab, add a new process trigger for an Upload file event.

2. Configure the process trigger as follows:

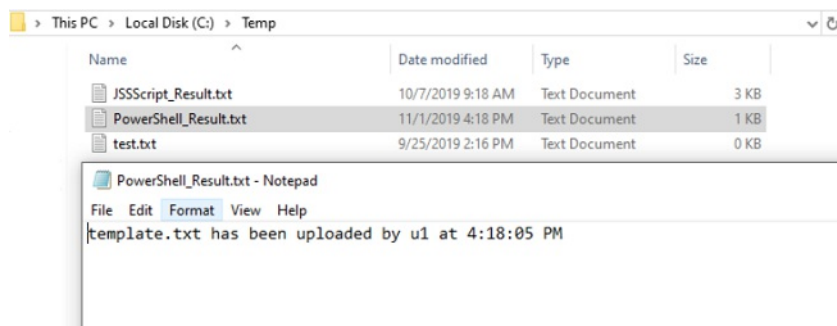
- Errors: Trigger on success.
- Folder filter: no restriction (\*).
- User filter: Select all.
- Site filter: Select all.
- Type: PowerShell script.
- Directory: leave it blank.
- Limits: leave it as default.

3. Input the above script with PowerShell Script, then 'Apply changes'.

The screenshot shows the 'Process Triggers' configuration window in CompleteFTP Manager. The 'Name' field is 'New process trigger 1'. The 'Events' section has 'Move file' checked. The 'Errors' section has 'Trigger on success' checked. The 'Folder filter' is set to '\*'. The 'User filter' is set to '(All users)'. The 'Site filter' is set to '(All sites)'. The 'Type' is set to 'Powershell script'. The 'Script' field contains the PowerShell code from the sample code block. The 'Directory' field is empty. The 'Limits' section shows 'Max processes' as 200 and 'Max run time' as 0 seconds.

4. From your FTP client, connect to the CompleteFTP server with the user `user1` and upload a file to the server -> Verify that the user is logged in and that the upload is successful.

5. Navigate to `C:\Temp` and verify that there is a `PowerShell_Result.txt` file, which contains the information of the newly uploaded file.



# Custom Extensions

## (requires Enterprise MFT)

With CompleteFTP custom extensions you can customize the behaviour of CompleteFTP by integrating your own .NET or Javascript code. Developing an extension involves [building a .NET assembly](#) or [developing Javascript code](#) and registering it with CompleteFTP.

### Types of Extensions

Six types of extensions are supported:

#### Authentication extensions

[[.NET](#) or [JSS](#)] Use this type of extension to implement any authentication scheme. The server calls methods in your class to find out whether the credentials presented by the client are valid. Your class can do whatever it needs to do to work out whether or not it is valid, such as calling a webservice or invoking an RPC.

#### File-system extensions

[[.NET](#) or [JSS](#)] Replace the regular file-system with one of your own design. Once your file-system extension is associated with a virtual folder, it will be called by the server whenever a client performs file operations (e.g. reading, writing, directory-listing) in that folder. Your class has complete control over how the operations are performed so it could, for example, tap directly into your company database without accessing any real files.

#### Custom command extensions

[[.NET](#) or [JSS](#)] Add your own commands simply by adding methods to your custom command extension. These commands will be available to clients via SITE commands in FTP/FTPS, via the shell in SSH and via the SSH\_FXP\_EXTENDED command in SFTP.

#### Event extensions

[[.NET](#)] CompleteFTP comes with e-mail and process-trigger event-handlers. With Event Extensions you can invoke methods on your own .NET classes when particular events occur. Currently event extensions can be written in .NET only.

#### IP filter extensions

[[.NET](#) or [JSS](#)] If the inbuilt IP filtering feature doesn't match your requirements you can develop your own.

#### JSS-to-.NET bridge extensions

[[.NET](#)] This type of extension facilitates invocation of .NET methods from JSS. These extensions must be written in .NET.

### Support

Ticket-based support is available to developers of extensions. Please direct questions to [Enterprise support](#). In particular, if you find that data and operations that your extensions require are not available from the classes described here then don't hesitate to request them to be made available.

### Development

We can write extensions for you! Please submit your requirements to [sales@enterprisedt.com](mailto:sales@enterprisedt.com) for a quote.

## Authenticator Class Reference

### Class: Authenticator

Authentication extensions must extend this class. Only `CheckUserName` and `Authenticate` must be overridden. The only job of `CheckUserName` is to set `userInfo.IsValidUserName` to `true` if the user-name (`userInfo.UserName`) is valid. `Authenticate` must set `userInfo.IsCorrectPassword` to `true` if the password (`authInfo.Password`) is correct for the given user-name (`userInfo.UserName`). Alternatively, `userInfo.IsValidKey` must be `true` if public key authentication is being attempted. If the home folder of the template user (usually `defaultExtension` - see below) is set to `%ExternalHomeFolder%` the value of `HomeDirectory` should be set to the path of the Windows directory that you wish to be the home of the user.

By default, users that are authenticated by authenticator extensions use the `defaultExtension` as a 'template user'. This allows the administrator to set the properties (e.g. permissions) of these users using the CompleteFTP Manager. The user that is used as a template can be changed by overriding the `TemplateUserName`. This property should return the name of the user that is to be used as a template.

The value of the Configuration field entered in the Manager is accessible via the `ExtensionConfiguration` property.

`LoadedUserInfo LoadUserInfo(IUserInfo suppliedUserInfo)`

Return an object containing the loaded user details if the user-name (`userInfo.UserName`) is valid. This method is called automatically in the default implementation of `CheckUserName`.

`void CheckUserName(IUserInfo userInfo)`

Set `userInfo.IsValidUserName` to `true` if the user-name (`userInfo.UserName`) is valid.

`void Authenticate(IAuthenticationInfo authInfo)`

Set `userInfo.IsCorrectPassword` to `true` if the password (`authInfo.Password`) is correct for the given user-name (`userInfo.UserName`). Or, if public key authentication is being used, set `userInfo.IsValidKey` to `true` if the key details supplied in the `IAuthenticationInfo` are valid. If the home folder of the template user (usually `default_external` - see above) is set to `%ExternalHomeFolder%` the value of `HomeDirectory` should be set to the path of the Windows directory that you wish to be the home of the user.

`void Initialize(IPlugInInfo info)`

Perform any required initialization that your extension requires.

`void Dispose()`

Perform any clean-up that your extension requires.

`string LogInAsUserName`

Optionally override this property to set a template user other than `default_external` (see above)

`string ExtensionName`

Name of the extension entered in the CompleteFTP Manager.

`Guid ExtensionID`

Identifier of the extension.

`string ExtensionConfiguration`

Configuration entered for the extension in CompleteFTP Manager.

### Class: LoadedUserInfo

`string HomeDirectory`

The user's home directory if available.

`List<byte[]> DSAPublicKeys`

The list of DSA public key blobs for this user. Each key is a byte array of the standard OpenSSH or SECSH formats for public keys.

`List<byte[]> RSAPublicKeys`

The list of RSA public key blobs for this user. Each key is a byte array of the standard OpenSSH or SECSH formats for public keys.

`string PasswordHash`

The MD5 hash of the user's password

`string PasswordSalt`

The salt prepended to the user's password prior to the MD5 hash. This can be null if a salt has not been used.

`string Password`

Setter only to set the user's password. This populates the `PasswordHash` field.

`bool MustChangePassword`

Set to `true` if the user must change their password

`List<string> Groups`

Names of groups of which the user is a member (in addition to those of the log-in-as user).

### Interface: IUserInfo

`string UserName`

User-name that the client is presenting.

`string Protocol`

Protocol that the client connect with

`IPEndPoint RemoteEndPoint`

Client's IP address and port-number.

`IPEndPoint LocalEndPoint`

IP address and port-number that the client connected to

`X509Certificate2 ClientCertificate`

SSL certificate presented by the client. The value will be `null` for all cases except where all of the following are true:

1. the protocol is FTPS or HTTPS
2. CompleteFTP is configured to require client certificates
3. the client certificate has already been validated via the Windows Certificate Store

`bool IsValidUserName`  
Set to `true` if the user-name is valid.

`string SiteName`  
Name of site that is requesting authentication

`Guid SiteID`  
Identifier of site that is requesting authentication. To obtain the ID for a given site you need to look at the log file; the ID is logged each time a site is started.

`List<string> Groups`  
Names of groups of which the user is a member (in addition to those of the log-in-as user).

`string HomeDirectory`  
Home folder override

## **Interface: IAuthenticationInfo**

IAuthenticationInfo extends IUserInfo so it has all the properties of IUserInfo plus:

`string Password`  
Password that the client is presenting.

`IsCorrectPassword`  
Set to `true` if the password is correct for the given user.

`bool MustChangePassword`  
Set to true if the user must change their password.

`bool IsValidKey`  
Set to `true` if public key authentication is successful for the given user.

`AuthenticationMethod AuthenticationMethod`  
Authentication being used.

`PublicKeyAlgorithm KeyAlgorithm`  
Provides the public key algorithm used if this is public key authentication.

`byte[] KeyCheckData`  
Provides the key check data if this is public key authentication. To authenticate using public key authentication, the client uses its private key to sign a block of data, producing SignatureBlob which is sent to the server. The server calculates the same block of data (the KeyCheckData) and verifies using the stored copy of the client's public key that SignatureBlob was indeed produced by the client signing the same block of data with its private key. The contents of KeyCheckData are defined in [RFC 4252 Section 7](#).

`byte[] SignatureBlob`  
Provides the signature blob if this is public key authentication. See KeyCheckData.

`X509Certificate2 ClientCertificate`  
SSL certificate presented by the client. The value will be `null` for all cases except where all of the following are true:

1. the protocol is FTPS or HTTPS
2. CompleteFTP is configured to require client certificates
3. the client certificate has already been validated via the Windows Certificate Store

## **Interface: IPlugIn**

`PlugInName`  
Name of plug-in as entered in CompleteFTP Manager.

`PlugInConfiguration`  
Configuration entered by administrator.

# JSS-to-.NET Bridge Extensions

JSS-to-.NET bridge extensions may be used to call into .NET from JSS, thus offering wide extensibility to the JSS platform. It might be used, for example, to wrap a simple .NET function, so that it may be used in JSS or as a bridge between JSS and a .NET API for a web-service.

A JSS-to-.NET bridge extension is a .NET assembly (usually a DLL) that contains a class, which extends the class `EnterpriseDT.Net.FtpServer.Core.PlugIn` and has one or more methods with the `JavascriptInvokable` attribute. The assembly can be developed in C#, VB.NET or any other .NET language. If you don't already have Visual Studio then you can use one of Microsoft's free Visual Studio Express or Community products.

## Creating a JSS-to-.NET Bridge Extension

General instructions on building CompleteFTP extensions may be found [here](#).

JSS-to-.NET bridge extension classes must extend `EnterpriseDT.Net.FtpServer.Core.PlugIn`. Each method that is to be exposed to JSS must have a `JavascriptInvokable` attribute. If the `JavascriptInvokable` attribute specifies a name, then this is the name by which the method must be called from JSS. If no name is specified, then the name of the .NET method is used.

Arguments may be passed to a method and values returned. Javascript numbers will be cast to .NET doubles; Javascript strings will be cast to .NET strings; Arrays will be cast to `List<object>`; and Javascript objects will be cast to `Dictionary<string, object>`.

## Calling .NET methods from Javascript

Once a JSS-to-.NET Bridge has been added the methods it contains may be called using the following pattern:

```
PlugInName.methodName()
```

where `PlugInName` is the name of the plug-in set in the `CompleteFTPManager` and `methodName` is either the name of the method or the name specified by the `JavascriptInvokable` attribute if one is provided for the method.

## Example

The following example implements methods that calculate hashes using two algorithms, MD5 and SHA1. Notice that the `JavascriptInvokable` attributes specify the names that should be used to call them from JSS, i.e. `calculateMD5` and `calculateSHA1`.

```
using System;
using EnterpriseDT.Net.FtpServer.Http.Javascript;
using System.Web.Security;
using EnterpriseDT.Net.FtpServer.Core;

public class HashFunctions : PlugIn
{
    [JavascriptInvokable("calculateMD5")]
    public string CalcMD5(string s)
    {
        return FormsAuthentication.HashPasswordForStoringInConfigFile(s, "MD5");
    }

    [JavascriptInvokable("calculateSHA1")]
    public string CalcSHA1(string s)
    {
        return FormsAuthentication.HashPasswordForStoringInConfigFile(s, "SHA1");
    }
    public override void Dispose()
    {
    }

    public override void Initialize(IPlugInInfo info)
    {
    }
}
```

Add the extension as type "JSS-to-.NET Bridge"™ in the Extensions panel in the Manager. Make sure that the name you use, only has letters and no spaces, .e.g. "HashFunctions".

Extensions	Permissions
<a href="#">Add extension</a> <a href="#">Remove</a>	
<div>Name</div> <div>HashFunctions</div>	<div>Name</div> <div>HashFunctions</div>
	<div>Type</div> <div>JSS-to-.NET bridge</div>
	<div>DLL location (use full path if not in Extensions directory)</div> <div>F:\FTP\Extension\SampleExtensions\bin\Debug\SampleExtensions.dll</div> <div>...</div>
	<div>Extension class (including namespace)</div> <div>HashFunctions</div> <div>...</div>
	<div>Editor class (including namespace)</div> <div></div>
	<div>Configuration</div> <div></div> <div>...</div>

Once youâ€™ve added the extension, you can use the class in a JSS upload process trigger like the following:

```
var ftp = new Ftp("Servername", "Username", "Password");
ftp.connect();
ftp.upload(event.virtualPath, event.fileName);
ftp.close();
console.log(HashFunctions.calculateMD5(event.virtualPath));
```

This should write the MD5 hash of the file-name to the log, each time a file is uploaded.

# How to write a .NET extension

A CompleteFTP .NET Extension is a .NET assembly, usually a DLL, which contains a class that extends one of the CompleteFTP extension base-classes (see [below](#)). The assembly can be developed in C#, VB.NET or any other .NET language. If you don't already have Visual Studio then you can use one of Microsoft's free Visual Studio Express or Community products. This feature is only available in Enterprise MFT.

Please note that from CompleteFTP 11.0, the .NET 4 framework is required to build .NET extensions. All extension DLLs will need to be recompiled with the target framework set to .NET 4.

## Types of .NET Extensions

For further details on the particular type of .NET extension which you are developing please go to:

- [Authentication extensions](#)
- [File-system extensions](#)
- [Custom command extensions](#)
- [Event extensions](#)
- [IP filter extensions](#)
- [JSS-to-.NET bridge extensions](#)

## Creating a .NET extension

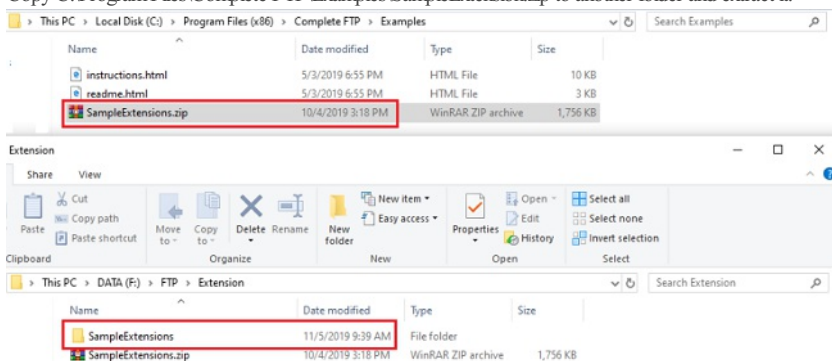
To create a CompleteFTP .NET extension:

1. Open Visual Studio.
2. Create a new project of type Class Library.
3. Add a reference to the CompleteFTPServer.dll. You can find this in the directory C:\Program Files\CompleteFTP\Server. Make sure the `Specific Version` property is set to `false` so that you don't have to rebuild the assembly every time you upgrade CompleteFTP.
4. If you are using CompleteFTP 11.0 or later, ensure the target framework in your application settings is .NET 4.
5. Add references (using `using` or `import`) to the namespaces required for your type of extension.
6. Make sure your new class extends the appropriate base-class.
7. Override the appropriate methods.
8. Add any [logging](#) you require.
9. Build the project.

## Using the existing .NET extension

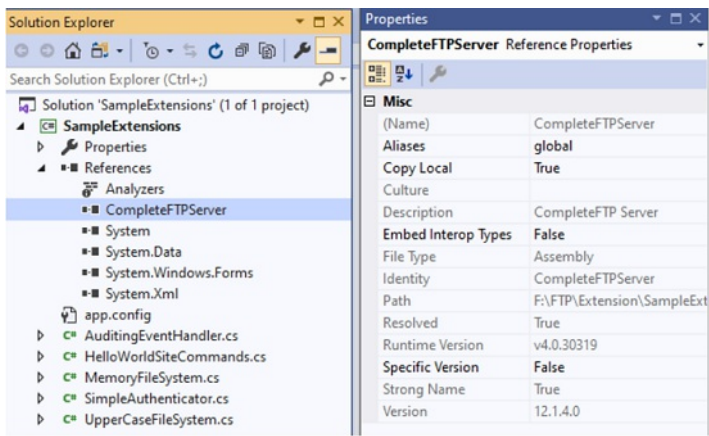
To use the existing .NET extension:

1. Open Visual Studio.
2. Copy C:\Program Files\Complete FTP\Examples\SampleExtension.zip to another folder and extract it.

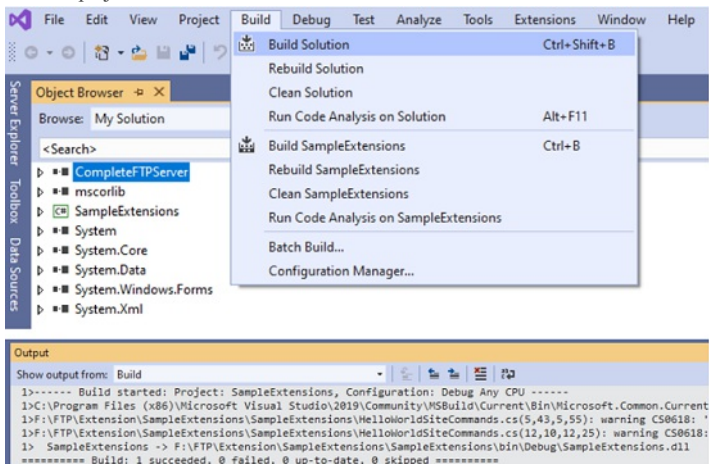


3. Navigate to the SampleExtension folder, then open SampleExtensions.sln file in Visual Studio.
4. Under the project tree, click References, if there is no CompleteFTPServer reference, you can add it, the path is: "C:\Program Files\Complete FTP\Examples\SampleExtensions\lib\CompleteFTPServer.dll".





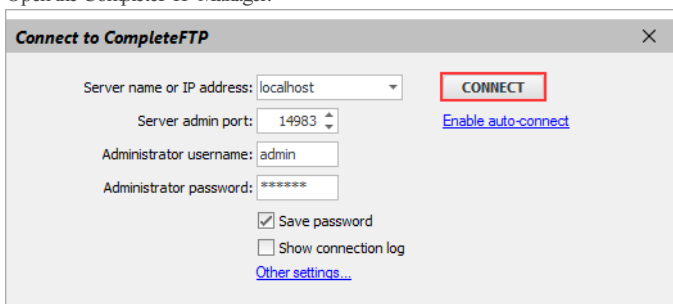
5. Build the project.



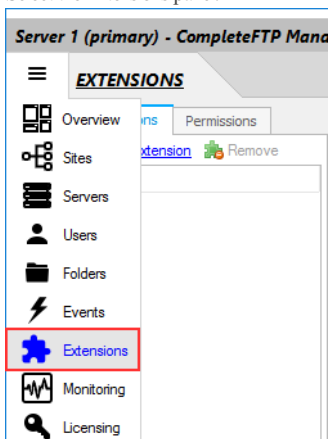
## Registering a .NET extension in CompleteFTP

Extensions must be registered with CompleteFTP. To do this:

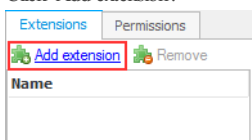
1. Open the CompleteFTP Manager.



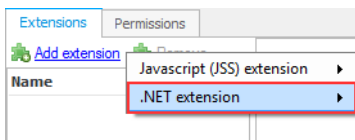
2. Select the Extensions panel.



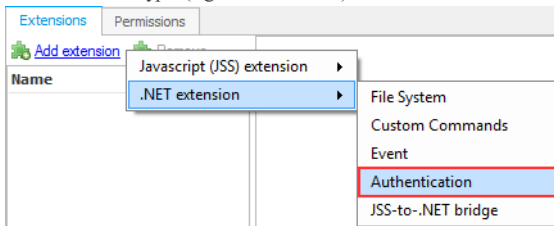
3. Click 'Add extension'.



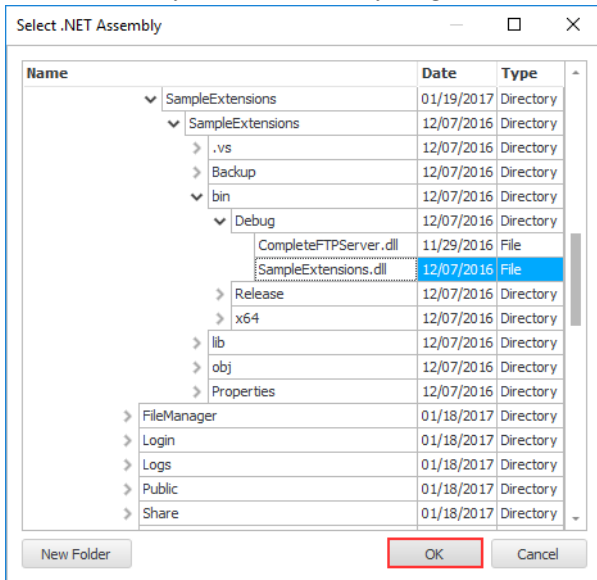
4. Select '.NET Extension'.



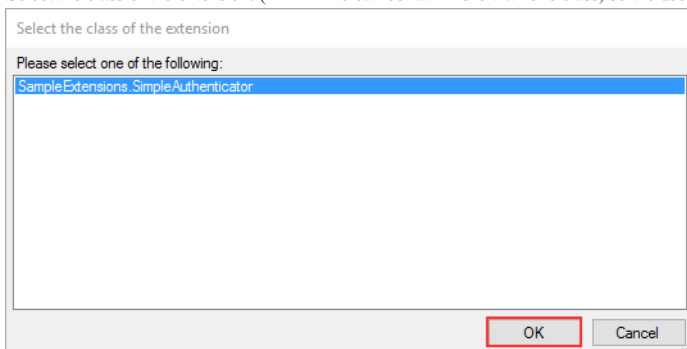
5. Select 'Extension type' (e.g. 'Authentication').



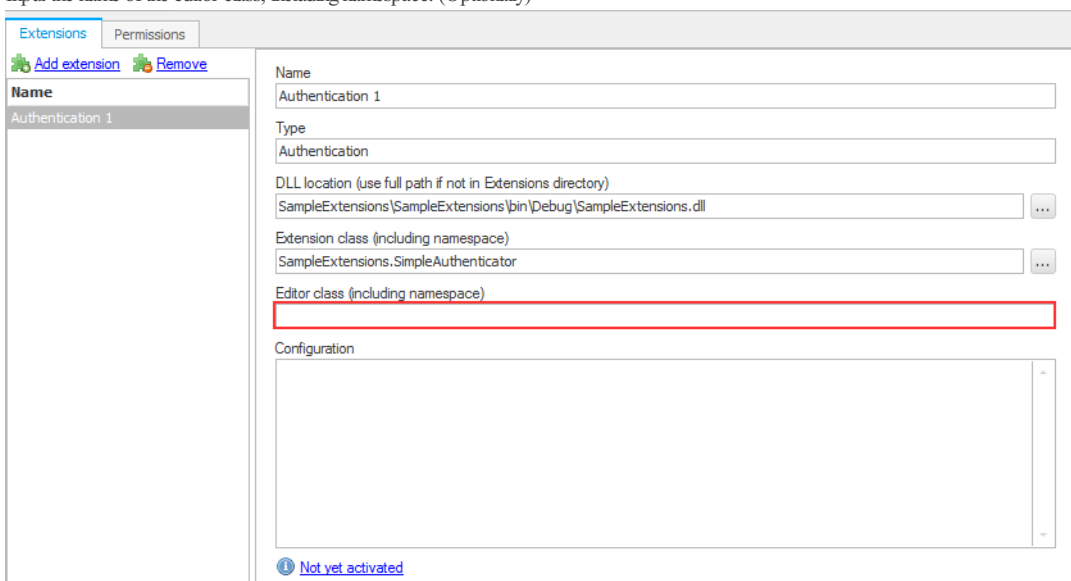
6. Select .NET Assembly file in the .NET Assembly Dialog. Note that the assembly is read by the System user, so this user must have access to the file.



7. Select the class of the extension. (A DLL file can contain more than one class, so the user has to select a class).



8. Input the name of the editor class, including namespace. (Optionally)



9. Some extension types require you to perform an additional registration step, such as adding a new folder-type, or granting permissions to site-commands.
10. Click the Apply button.

EXTENSIONS

Extensions

Permissions

Add extension

Remove

Name

Authentication 1

Type

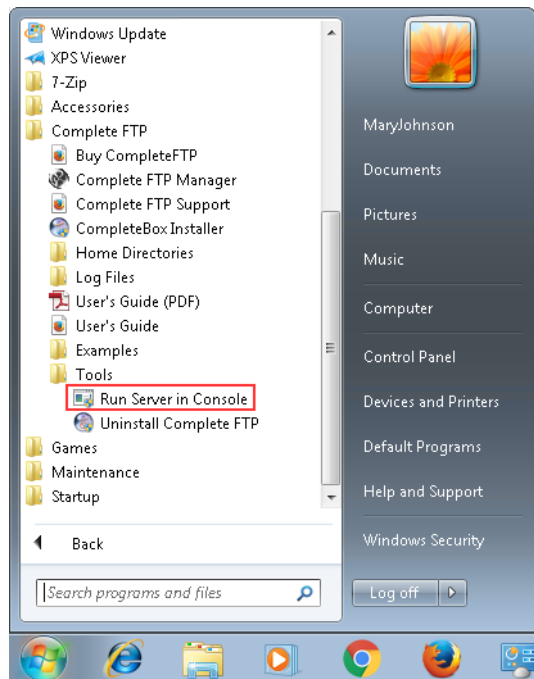
Authentication

APPLY CHANGES

11. Test your .NET Extension by using an FTP client.

## Reloading a .NET extension in CompleteFTP

.NET assemblies cannot be reloaded if they are altered. If you need to make a change to the code of the extension, then you will need to restart the CompleteFTP service to force it to reload the DLL. While developing, it is sometimes helpful to run CompleteFTP in the console mode. This may be done by closing the CompleteFTP service and then selecting **Run server in console** from the Windows start menu.



## .NET Framework Versions

The .NET assembly file must be built with a version of the .NET Framework runtime that is the same as, or older than, the .NET Framework version on which CompleteFTP is running. This can be controlled by editing the `supportedRuntime` element in the file, `CompleteFTPService.exe.config` in the Complete FTP installation folder. E.g. if you want your CompleteFTP's extensions to run on .NET Framework 4.0, you need to make sure that 4.0 appears before 2.0 inside the `supportedRuntime` element and that .NET 4.0 is installed on your machine.

# .NET Authentication Extensions

Use this type of extension to implement any authentication scheme. .NET Authentication Extensions are known as *authenticators*. The server calls methods in your class to find out whether or not a particular user-name/password combination or user/public key combination is valid. Your class can do whatever it needs to do to work out whether or not it is valid, such as calling a webservice or invoking an RPC.

## Creating a .NET Authentication Extension

General instructions on building CompleteFTP .Net extensions may be found [here](#).

.NET Authentication Extension classes must extend `EnterpriseDT.Net.FtpServer.Core.Authenticator`.

CompleteFTP includes a number of predefined authenticators, such as the [external database authenticator](#), the [gateway authenticator](#) (Enterprise MFT only), the [automatic Windows users \(AWU\) authenticator](#), and the [SAML single sign-on authenticator](#).

Note that the order of precedence for authentication is as follows:

1. users explicitly defined within CompleteFTP (both non-Windows and Windows).
2. [external database users](#), [gateway users](#), [SAML single sign-on users](#), and [custom authentication extensions](#)
3. [automatic Windows users \(AWU\)](#).

So if a user is explicitly defined in CompleteFTP and also in an extension, then the extension will never be used to authenticate the user.

When a .NET Authentication Extension is used, the *defaultExtension* user (which may be found in Users panel and by selecting "Show system users/folders/sites" in the main form's Options menu) is used as a template for user details once login has been successful (shown below).

There are two ways to implement a .NET Authentication Extension. The [simple .NET Authentication Extension](#) is the most straightforward, and should be attempted first. If more flexibility is required, consider writing an [advanced .NET Authentication Extension](#).

Note that once an authenticator has been written, it must be [registered in CompleteFTP](#) with an Extension Type of authenticator. This is done via the CompleteFTP manager. It must then be enabled by selecting the Enabled check box in the *General User Settings* dialog box, accessible from the Users tab in the manager.

Add user

Remove

General user settings

Enabled	Username	Type	Home Folder	Groups
<input checked="" type="checkbox"/>	myadmin	Non-Windows		admins
<input checked="" type="checkbox"/>	u1	Non-Windows	/Home/u1	users
<input checked="" type="checkbox"/>	u2	Non-Windows	/Home/u2	users

Select

Groups

Two-factor auth status

**AUTHENTICATORS**

For site: Default Site

Authenticator	Enabled	Log in as	
Authentication 1	<input checked="" type="checkbox"/>	defaultExtension	
Database	<input type="checkbox"/>	defaultDatabase	<a href="#">Configure...</a>
Gateway	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Single sign-on / SAML	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Windows/AD Group	<input type="checkbox"/>	defaultWindows	<a href="#">Configure...</a>

# Simple .NET Authentication Extensions

In many (or even most) cases, authenticators can be written as described below. If this is not flexible enough for a particular application, an [advanced .NET Authentication Extension](#) may be required.

The only method that needs to be implemented is `LoadedUserInfo LoadUserInfo(IUserInfo suppliedUserInfo)`. When the server calls this method in your authenticator, it supplies user information that enables your extension to look up that user's authentication details from an external source. You then return a `LoadedUserInfo` object from this method, which contains the user's home directory, password hash and/or public keys. The authenticator caches this loaded user object, and uses it to authenticate the user.

See the [class reference](#) for more details of the classes and interfaces involved.

[Example 1](#) shows sample source code for implementing password authentication. See [Example 2](#) for an example of public key authentication and password authentication.

Optionally, the `void SetPassword(IUserInfo suppliedUserInfo, string oldPassword, string newPassword)` can also be implemented. This can be used to change the user's password in the external user data source. Ultimately, this method is called via the FTP SITE command, CPWD, or via the SSH/SFTP password changing mechanism.

SSH/SFTP also permits servers to force a password change on clients. This can be implemented by setting the `LoadedUserInfo.MustChangePassword` property on the `LoadedUserInfo` object that is returned from `LoadUserInfo`. FTP does not permit the server to force a user to change their password.

See [Example 3](#) for how to force and implement password changes in SSH.

## Example 1

Below is the source code for a sample authenticator which supports password authentication. For simplicity, the password is hardcoded, whereas in production code, it would be retrieved from a data source. The password is not stored - instead, the `PasswordHash` field is populated when the `Password` field is set. The `PasswordHash` field can be set directly - normally only password hashes would be stored.

```
using EnterpriseDT.Net.FtpServer.Core;

namespace EnterpriseDT.Net.FtpServer.Extensions.Test
{
    public class SimplePasswordAuth : Authenticator
    {
        public override LoadedUserInfo LoadUserInfo(IUserInfo suppliedUserInfo)
        {
            if (suppliedUserInfo.UserName == "fred")
            {
                LoadedUserInfo info = new LoadedUserInfo();
                info.Password = "fredspassword";
                return info;
            }
            else
            {
                return null;
            }
        }
    }
}
```

By Using that example code to create a CompleteFTP .NET extension, you can then build the project. Note that the CompleteFTP has to be stopped completely.

Add .NET authentication extensions.

Extensions	Permissions
Add extension  Remove	
<b>Name</b>	
Authentication 1	Name: Authentication 1
	Type: Authentication
	DLL location (use full path if not in Extensions directory): F:\FTP\Extension\SampleExtensions\bin\Debug\SampleExtensions.dll
	Extension class (including namespace): SampleExtensions.SimplePasswordAuth
	Editor class (including namespace):
	Configuration:
	Activated successfully

Enable the Authentication in the User panel.

<b>AUTHENTICATORS</b>			
For site: Default Site			
Authenticator	Enabled	Log in as	
Authentication 1	<input checked="" type="checkbox"/>	defaultExtension	
Database	<input type="checkbox"/>	defaultDatabase	<a href="#">Configure...</a>
Gateway	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Single sign-on / SAML	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>

Login FileZilla by username = "fred" and password = "fredpassword".

fred@localhost - FileZilla

File Edit View Transfer Server Bookmarks Help New version available!

Host: localhost
Username: fred
Password: .....
Port:
Quickconnect

Status: Connecting to [::1]:21...  
Status: Connection attempt failed with "ECONNREFUSED - Connection refused by server", trying next address.  
Status: Connecting to 127.0.0.1:21...  
Status: Connection established, waiting for welcome message...  
Status: Initializing TLS...  
Status: Verifying certificate...  
Status: TLS connection established.  
Status: Logged in  
Status: Retrieving directory listing...  
Status: Directory listing of "/"ExtensionUser" successful

## Example 2

Below is the source-code for a sample authenticator which allows a user to login via SFTP using public key authentication or by password authentication. For simplicity, the public keys are hard coded, whereas in production code they would be created by [this guide](#).

```

using System.Text;
using EnterpriseDT.Net.FtpServer.Core;

namespace EnterpriseDT.Net.FtpServer.Extensions.Test
{
    public class PublicKeyTest : Authenticator
    {
        public override LoadedUserInfo LoadUserInfo(IUserInfo suppliedUserInfo)
        {
            LoadedUserInfo info = new LoadedUserInfo();
            info.RSAPublicKeys.Add(RSAPublicKey);
            info.DSAPublicKeys.Add(DSAPublicKey);
            info.Password = "mypassword";
            return info;
        }

        private byte[] RSAPublicKey
        {
            get
            {
                string key = "---- BEGIN SSH2 PUBLIC KEY ----\r\n" +
                    "Comment: \"rsa - key - 20200305\"\r\n" +
                    "AAAAB3NzaC1yc2EAAAQEAAsb6uRtZEwCYuV7o1VQ5H3L1WPxelyjc1r7LL\r\n" +
                    "mLGbZN0muXXe5Vvqb+p0AIjqkCYBuL+wj2WPxIUWU1gG0hrs0poS1LbEbQTizrh/\r\n" +
                    "rkwl3GPr+ptc28h0GBVzDUvGip37rh2fRa5SiZT8UP04kCGB1c5x52pFnPCh9ZQP\r\n" +
                    "p073IvCaRFepuDhW0TBxbdJgAqP+KY5FOV4AcTj/60bfoeKRYMawwhGxbOCcycS6\r\n" +
                    "tW2yQnQFfz4x5ThSIU146GR/ViWnrFYeQ4+CNHkJ/JuBAhe4SlywmZg1mNE390SU\r\n" +
                    "HTzPEew+tiTam2qUQ+jnHMF67y8ga2wSkLpdRWzg8kGRWBf2SQ==\r\n" +
                    "---- END SSH2 PUBLIC KEY ----";
                return Encoding.ASCII.GetBytes(key);
            }
        }

        private byte[] DSAPublicKey
        {
            get
            {
                string key = "---- BEGIN SSH2 PUBLIC KEY ----\r\n" +
                    "Comment: \"imported-openssh-key\"\r\n" +
                    "AAAAB3NzaC1kc3MAAACBAPos9tWoXLcd//d0GbaA+1TC09vEi0jQOQM85j34E4Ua\r\n" +
                    "Sza5yjs3vI9K9XchJirbNYrRQNmgM2yn3fUDdTPU5eES+mZRY9K9qpAesk4Ghpwu\r\n" +
                    "btWc3e0APkQTUAoRHL8yiW1tHrRdV6yrowgKDPrIccnL90wYAZFHmUmwIeiESjTB\r\n" +
                    "AAAAFQDhvm9w83LDeixC3oPW+F0Kk673dQAAAIBoBehA6t+eRtNTocY1sb7Dly00\r\n" +
                    "ReeRWo+mHEyUts78ayAN7YFNzTd8UXmUgw8gyGFt0/tXrkeLG46vMhL/0402ek9Z\r\n" +
                    "jNcDq2vF1InYIaOxcceURqg99VGQUqrjEWchIG5egDgtOKRAUtUyK7I52CXG3wN9\r\n" +
                    "20q+W0oUztJCSwgmmwAAAIeAu4G5CHifmoTsBVcObaRkW8UqrTCmz7C84W6AaXA5\r\n" +
                    "uBlwtTIBlAUkZfqStpC76rucJ6i3R9Nk+gHrDb4v6uA6at2UZDlHZlwPCg88fk7\r\n" +
                    "Nbi5umH9B/QSfm+GQOd+ttD54F0cR+lwmrJ+f1mzSX9v9ZrVi+xJJ6Jp+5NDa7g\r\n" +
                    "KtM=\r\n" +
                    "---- END SSH2 PUBLIC KEY ----";
                return Encoding.ASCII.GetBytes(key);
            }
        }
    }
}

```

Firstly, go to CompleteFTP, generate the key pair for the user (public key and private key). Use the public key by adding it into the code.

Manage Public Keys

Public keys: (any of these may be used by the user for authentication)

Type	Fingerprint	Encoded
DSA	a2:e5:eb:f7:ac:d6:b7:ca:ed:6f:1e:45:c4:6...	AAAAB3NzaC1kc3MAAACBAIAAALMAAAUAAA...
RSA	46:7f:be:9a:71:bd:b5:e0:ee:d9:ee:a9:30:...	AAAAB3NzaC1yc2EAAAADAQABAAQDLSw...

[Import a public key and add it to the list](#)  
[Remove currently selected key from the list](#)  
[Export the selected key in OpenSSH format](#)  
[Export the selected key in SECSH format](#)  
[Generate a new keypair for this user and add it to the list](#)

OK Cancel

Build the project again.

Make sure the setting in Default Sites/SFTP/SCP/AdvanceSFTP/authentication method = publickey and password.

Site

Default Site

Site name	Default Site
Site label	CompleteFTP
Site enabled	<input checked="" type="checkbox"/>
<b>FTP / FTPS</b>	
FTP enabled	<input checked="" type="checkbox"/>
FTPS explicit enabled (standard FTPS)	<input checked="" type="checkbox"/>
FTPS implicit enabled (legacy FTPS)	<input checked="" type="checkbox"/>
<b>Anonymous Logins</b>	
<b>Advanced FTP/FTPS Settings</b>	
<b>SFTP / SCP</b>	
SFTP enabled	<input checked="" type="checkbox"/>
SCP enabled	<input checked="" type="checkbox"/>
SSH terminal access enabled	<input type="checkbox"/>
SSH forwarding enabled	<input type="checkbox"/>
<b>Advanced SFTP/SCP Settings</b>	
Port for SFTP/SCP/SSH	22
Listening IP addresses	All available
Authentication methods	All
RSA Key	RSA key: 2048 bits
DSA Key	DSA key: 1024 bits
ECDSA Keys	nistp256
<b>Algorithms</b>	
<b>HTTP / HTTPS</b>	
HTTP enabled	<input checked="" type="checkbox"/>
HTTPS enabled	<input checked="" type="checkbox"/>

Login with WinSCP using SFTP with password = "mypassword" and add private key for this login (Advanced/SSH/Authentication/Private key file).

Session

File protocol: SFTP

Host name: localhost Port number: 22

User name: Password: .....

Save Advanced...

Environment

- Directories
- Recycle bin
- Encryption
- SFTP
- Shell
- Connection
- Proxy
- Tunnel
- SSH
- Key exchange
- Authentication
- Bugs
- Note

☐ Bypass authentication entirely

Authentication options

- ☒ Attempt authentication using Pageant
- ☒ Attempt 'keyboard-interactive' authentication
- ☒ Respond with password to the first prompt
- ☐ Attempt TIS or CryptoCard authentication (SSH-1)

Authentication parameters

- ☐ Allow agent forwarding
- Private key file: C:\Users\Natalie\Desktop\Key\Natprivatekey1.ppk
- Display Public Key Tools

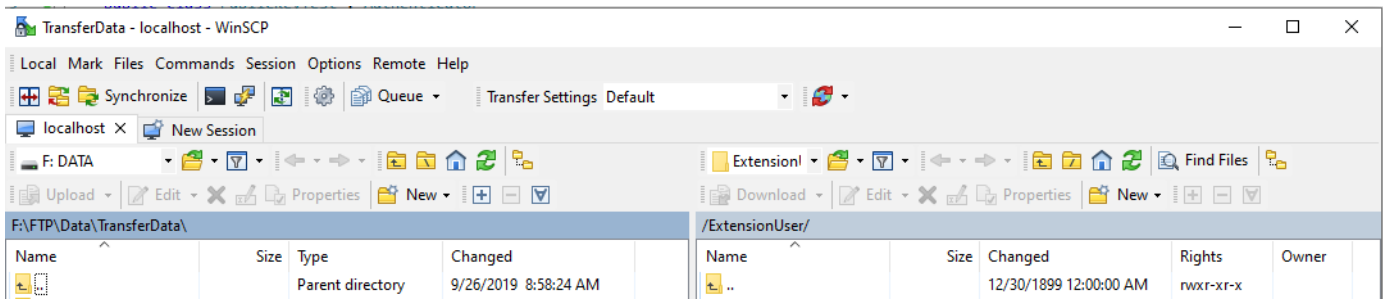
GSSAPI

- ☒ Attempt GSSAPI authentication
- ☐ Allow GSSAPI credential delegation

Login Close Help

Login should be successful





### Example 3

Below is the source-code for a sample authenticator which allows a user to login with SFTP by password authentication, but forces the user to change their password. This is an SSH-specific mechanism that many SSH clients support.

There are three requirements for doing this. Firstly, the `LoadedUserInfo` object must have the `MustChangePassword` property set to true. The `void SetPassword(IUserInfo suppliedUserInfo, string oldPassword, string newPassword)` method must be overridden to allow the new password to be set - and to throw an `InvalidPasswordException` if the new password does not pass the required checks. And finally, from the *Password policies* dialog accessed from the link in the *General User Settings* dialog box (accessible from the Users tab in the manager), *Permit password changes* must be selected.

```
using EnterpriseDT.Net.FtpServer.Core;

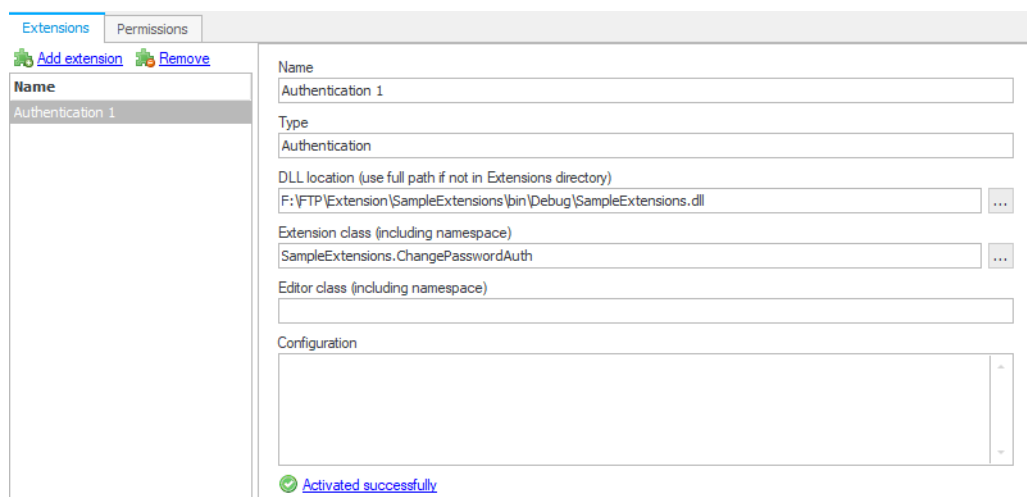
namespace EnterpriseDT.Net.FtpServer.Extensions.Test
{
    public class ChangePasswordAuth : Authenticator
    {
        private const string PASSWORD = "javaftp";
        private string password = PASSWORD;

        public override LoadedUserInfo LoadUserInfo(IUserInfo suppliedUserInfo)
        {
            LoadedUserInfo info = new LoadedUserInfo();
            info.Password = password;
            info.MustChangePassword = (password == PASSWORD);
            return info;
        }

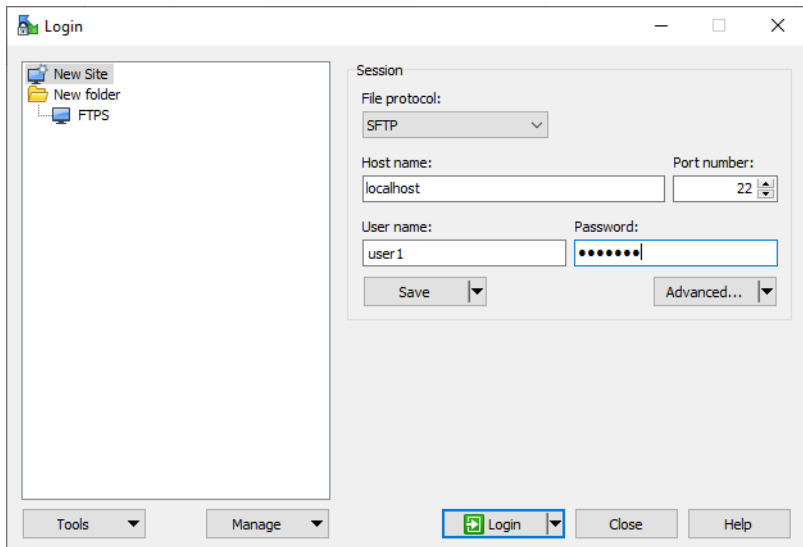
        public override void SetPassword(IUserInfo suppliedUserInfo, string oldPassword, string newPassword)
        {
            if (oldPassword == newPassword)
                throw new InvalidPasswordException("New password must be different");
            password = newPassword;
        }
    }
}
```

Add the above code into project then build it.

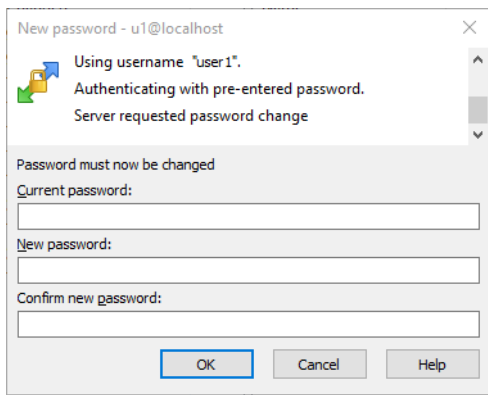
Add .NET authentication extensions. Note to enable the Authentication 1 in the Users panel.



Login with WinSCP using SFTP by user = user1 (which does not exist in CFTP) and password = "javaftp" (which is provided in the code).



It forces them to change the password after clicking on the login button.



Login should be successful after changing the password.

# Advanced .NET Authentication Extensions

Simple .NET Authentication Extensions can usually be used to accomplish what extension developers require. However at other times more complicated scenarios may require more flexibility.

As with simple extensions, advanced .NET Authentication Extension classes must extend `EnterpriseDT.Net.FtpServer.Core.Authenticator`.

However instead of overriding the `LoadUserInfo` method, they override the `CheckUserName` and `Authenticate` methods. In the `EnterpriseDT.Net.FtpServer.Core.Authenticator` base class implementation, `LoadUserInfo` is called by `CheckUserName`. But `LoadUserInfo` need not be overridden - instead `CheckUserName` can be overridden and `LoadUserInfo` need not be called at all.

Similarly, the `Authenticate` method is implemented in the `Authenticator` class, and if it is provided in `CheckUserName` or indirectly via `LoadUserInfo`, the `LoadedUserInfo` class containing the user's data will automatically be used to authenticate the user. However `Authenticate` can be overridden to authenticate in whatever way the extension writer prefers.

The primary requirement when overriding `CheckUserName` is to ensure that the `IsValidUserName` property of the supplied `IUserInfo` argument is set to true for a valid user. When overriding `Authenticate`, ensure that the `IsCorrectPassword` property of the supplied `IAuthenticationInfo` argument is set to true for password authentication, or `IsValidKey` is set to true for public key authentication. `IAuthenticationInfo.AuthenticationMethod` can be inspected to determine if password or public key authentication is being used. `IsValidRSAKey` and `IsValidDSAKey` can be used to determine if a key blob is the correct public key matching that which the user has supplied as part of their public key authentication process.

See the [class reference](#) for more details of the classes and interfaces involved.

## Example 1

Below is the source-code for a sample authenticator, which allows a user to log in with the username, `myusername`, and the password, `mypassword`. The user's home directory will be `C:\Temp` if the defaultExtension's home-directory is set to `%ExternalHomeFolder%`.

```
using EnterpriseDT.Net.FtpServer.Core;

namespace MyTestAuthenticator
{
    public class PasswordAuthenticator : Authenticator
    {
        public override void CheckUserName(IUserInfo userInfo)
        {
            if (userInfo.UserName == "myusername")
                userInfo.IsValidUserName = true;
        }

        public override void Authenticate(IAuthenticationInfo authInfo)
        {
            if (authInfo.AuthenticationMethod == AuthenticationMethod.Password && authInfo.Password == "mypassword")
            {
                authInfo.IsCorrectPassword = true;
                authInfo.HomeDirectory = "C:\\Temp";
            }
        }
    }
}
```

Add the code above into the project and build the project.

Add .NET authentication extensions.

Enable the Authentication in the User panel.

**AUTHENTICATORS**

For site: Default Site

Authenticator	Enabled	Log in as	
Authentication 1	<input checked="" type="checkbox"/>	defaultExtension	
Database	<input type="checkbox"/>	defaultDatabase	<a href="#">Configure...</a>
Gateway	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Single sign-on / SAML	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>

Login with WinSCP via SFTP with username = "myusername" and password = "mypassword".

Login successful.

## Example 2

Below is the source-code for a sample authenticator which allows a user to log in with the username, myusername via public key authentication. For simplicity, the public keys of the user are hard coded, whereas in production code they would be created as shown [here](#). The user's home directory will be C:\Temp if the defaultExtension's home-directory is set to %ExternalHomeFolder%.

```
using System.Text;
using EnterpriseDT.Net.FtpServer.Core;

namespace MvTestAuthenticator
```

```

{
public class PublicKeyAuthenticator : Authenticator
{
public override void CheckUserName(IUserInfo userInfo)
{
if (userInfo.UserName == "myusername")
userInfo.IsValidUserName = true;
}

public override void Authenticate(IAuthenticationInfo authInfo)
{
if (authInfo.AuthenticationMethod == AuthenticationMethod.PublicKey)
{
if (authInfo.KeyAlgorithm == PublicKeyAlgorithm.DSA)
{
authInfo.IsValidKey = IsValidDSAKey(DSAPublicKey, authInfo);
}
else
{
authInfo.IsValidKey = IsValidRSAKey(RSAPublicKey, authInfo);
}
if (authInfo.IsValidKey)
authInfo.HomeDirectory = "C:\\Temp";
}
}

private byte[] RSAPublicKey
{
get
{
string key = "---- BEGIN SSH2 PUBLIC KEY ----\r\n" +
"Comment: \"imported-openssh-key\"\r\n" +
"AAAAB3NzaC1yc2EAAAABIwAAAIEA44J6LBloMMWVvh0jMHZPnmGJWw+UWB19nFEWa\r\n" +
"62IFDrJDg6+kJ2DQD8v0TsQmNjk8803v+r0r/rr+QrotuLrdjrXBvrRrQNMfEbMo\r\n" +
"LhSmUVEFR/Yy3HjVRT6DHhJYPpr1xaXE6++fo5b2ax1zw+d1fPsh531bAhrCHV9b\r\n" +
"NIOimDk=\r\n" +
"---- END SSH2 PUBLIC KEY ----";
return Encoding.ASCII.GetBytes(key);
}
}

private byte[] DSAPublicKey
{
get
{
string key = "---- BEGIN SSH2 PUBLIC KEY ----\r\n" +
"Comment: \"imported-openssh-key\"\r\n" +
"AAAAB3NzaC1kc3MAAACBAPos9tWoXLcd//dOGbaA+1TC09vEi0jQ0QM85j34E4Ua\r\n" +
"Sza5yjS3vI9K9XchJirbNYrRQNmGM2yn3fUDdTPU5eES+mZRy9K9qpAesk4Ghpwu\r\n" +
"btWc3e0APkQTUAoRHL8yiWtHrRdV6yrowgKDPrIccnL90wYAZFHmUmwIeiESjTB\r\n" +
"AAAAFQDhvm9w83LDeixC3oPW+F0Kk673dQAAAIBoBehA6t+eRtNTocY1sb7Dly00\r\n" +
"ReeRwo+mHEyUts78ayAN7YFNzTd8UXmUgw8gyGFt0/tXrkeLG46vMhL/0402ek9Z\r\n" +
"jNcDq2vF1InYIaOxceuRqg99VGQUqrjEWchIG5egDgtOKRAUtUyK7I52CXG3wN9\r\n" +
"20q+W0oUztJCSwgmwwAAAIEAu4G5CHifmoTsBVcObaRkW8UqrTCmz7C84W6AaXA5\r\n" +
"uBlwtTIB1AUnKzfStpC76rucJ6i3R9Nk+gHrDb4v6uA6at2UZD1HZ1wPCg88fk7\r\n" +
"NbiSumH9B/QSfm+GQOd+ttD54FOcR+lwmerJ+f1mzSX9v9ZrVi+xxJJ6Jp+5NDa7g\r\n" +
"KtM=\r\n" +
"---- END SSH2 PUBLIC KEY ----";
return Encoding.ASCII.GetBytes(key);
}
}
}
}

```

Firstly, go to CompleteFTP, generate the key pair for the user (public key and private key). **Add the public key to the code.**

Manage Public Keys

Public keys: (any of these may be used by the user for authentication)

Type	Fingerprint	Encoded
DSA	a2:e5:eb:f7:ac:d6:b7:ca:ed:6f:1e:45:c4:6...	AAAAB3NzaC1kc3MAAACBAIAAALMAAAUAAA...
RSA	46:7f:be:9a:71:bd:b5:e0:ee:d9:ee:a9:30:...	AAAAB3NzaC1yc2EAAAADAQABAAQDLSw...

[Import a public key and add it to the list](#)  
[Remove currently selected key from the list](#)  
[Export the selected key in OpenSSH format](#)  
[Export the selected key in SECSH format](#)  
[Generate a new keypair for this user and add it to the list](#)

OK Cancel

Build the project again.

Go to the setting in Default Sites/SFTP/SCP/AdvanceSFTP/authentication method = publickey and password.

Site

Default Site

Site name	Default Site
Site label	CompleteFTP
Site enabled	<input checked="" type="checkbox"/>
<b>FTP / FTPS</b>	
FTP enabled	<input checked="" type="checkbox"/>
FTPS explicit enabled (standard FTPS)	<input checked="" type="checkbox"/>
FTPS implicit enabled (legacy FTPS)	<input checked="" type="checkbox"/>
<b>Anonymous Logins</b>	
<b>Advanced FTP/FTPS Settings</b>	
<b>SFTP / SCP</b>	
SFTP enabled	<input checked="" type="checkbox"/>
SCP enabled	<input checked="" type="checkbox"/>
SSH terminal access enabled	<input type="checkbox"/>
SSH forwarding enabled	<input type="checkbox"/>
<b>Advanced SFTP/SCP Settings</b>	
Port for SFTP/SCP/SSH	22
Listening IP addresses	All available
Authentication methods	All
RSA Key	RSA key: 2048 bits
DSA Key	DSA key: 1024 bits
ECDSA Keys	nistp256
<b>Algorithms</b>	
<b>HTTP / HTTPS</b>	
HTTP enabled	<input checked="" type="checkbox"/>
HTTPS enabled	<input checked="" type="checkbox"/>

Log with WinSCP via SFTP with username = "myusername" and add private key for this login (Advanced/SSH/Authentication/Private key file).

Login

Session

File protocol: SFTP

Host name: localhost Port number: 22

User name: myusername Password:

Save Advanced...

Tools Manage Login Close Help

Advanced Site Settings

Environment

- Directories
- Recycle bin
- Encryption
- SFTP
- Shell

Connection

- Proxy
- Tunnel

SSH

- Key exchange
- Authentication
- Bugs

Note

Bypass authentication entirely

Authentication options

- ☒ Attempt authentication using Pageant
- ☒ Attempt 'keyboard-interactive' authentication
- ☒ Respond with password to the first prompt
- ☐ Attempt TIS or CryptoCard authentication (SSH-1)

Authentication parameters

- ☐ Allow agent forwarding

Private key file:

\\.\Users\Natalie\Desktop\Key\U3RSAprivatekey1.ppk

Display Public Key Tools

GSSAPI

- ☒ Attempt GSSAPI authentication
- ☐ Allow GSSAPI credential delegation

Login successful.

TransferData - localhost - WinSCP

LocalMarkFilesCommandsSessionOptionsRemoteHelp

Synchronize

Queue

Transfer Settings: Default

localhost XNew Session

F: DATA

Extension!

Find Files

Upload

Edit

Properties

New

Download

Edit

Properties

New

F:\FTP\Data\TransferData\

/ExtensionUser/

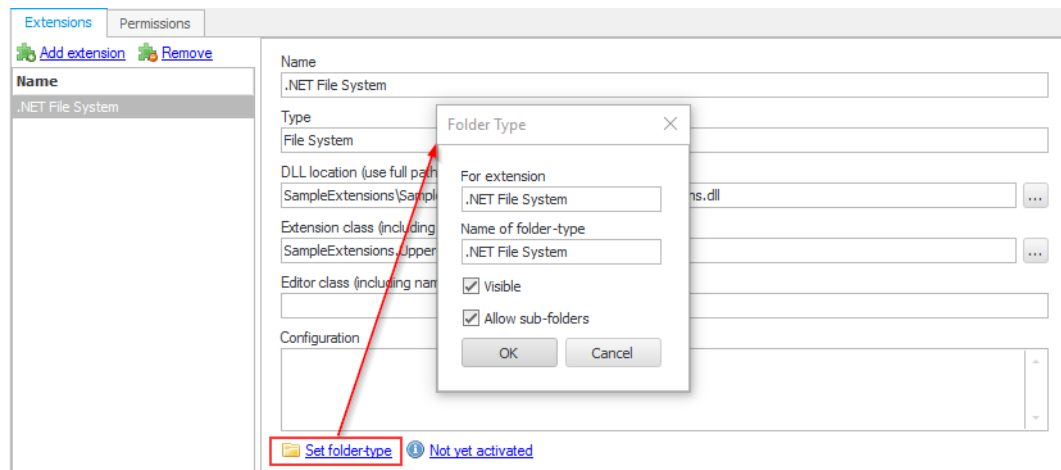
Name	Size	Type	Changed	Name	Size	Changed	Rights	Owner
Parent directory			9/26/2019 8:58:24 AM	..		12/30/1899 12:00:00 AM	rwxx-rx-x	

# .NET File-System Extensions

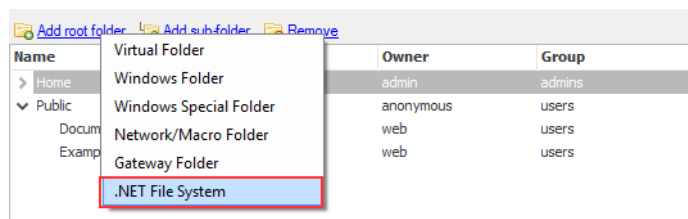
A .NET file-system extension is a .NET assembly (usually a DLL) that contains a class that extends the class `EnterpriseDT.Net.FtpServer.FileSystem.FileSystemAdapter`. The assembly can be developed in C#, VB.NET or any other .NET language. If you don't already have Visual Studio then you can use one of Microsoft's free Visual Studio Express or Community products.

## Creating a .NET file-system extension

General instructions on building CompleteFTP .NET extensions may be found [here](#). In addition to the registration steps described in the general instructions, a .NET file-system extension must have a folder-type defined. The folder-type determines which extension should be used for a specific folder-type.



Once a new folder-type is added, it will be displayed in the list of folder-types that is displayed when a user adds a new folder to the virtual file-system.



File-system extension classes must extend `EnterpriseDT.Net.FtpServer.FileSystem.FileSystemAdapter` or one of its subclasses, e.g. `WindowsAdapter`, `FolderAdapter` or `SpecialFolderAdapter`. The base-class, `FileSystemAdapter`, is a concrete class but it disallows all operations, so some of its methods must therefore be overridden to make it functional. For example, if directory-listing and downloading are to be supported then `GetFileInfos` and `GetReadStream` must be implemented at a minimum. Many FTP clients do things like checking sizes and modification dates before downloading, so methods like `GetFileLength` and `GetModifiedTime` may also need to be implemented.

## Example

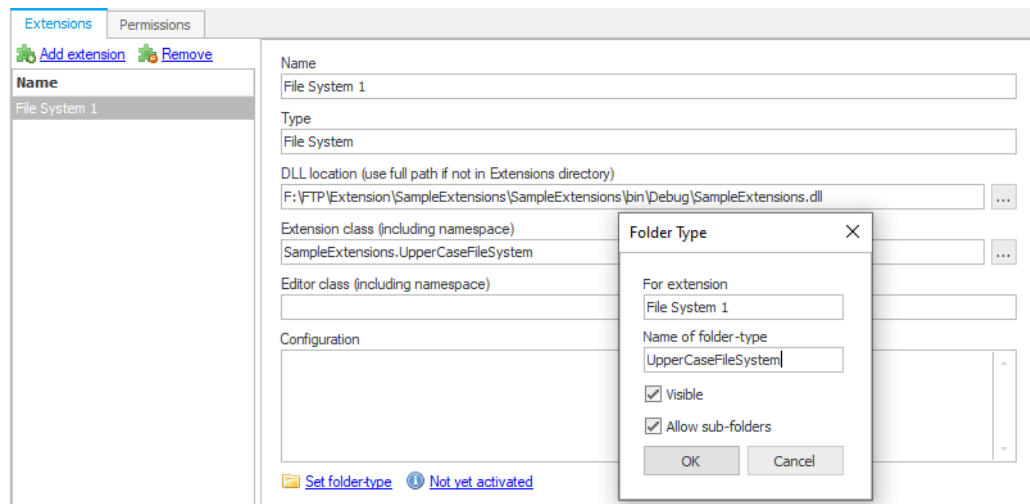
Below is the source-code for a sample .NET file-system extension whose only function is to display file-names in upper case.

```
using System;
using System.Windows.Forms;
using EnterpriseDT.Net.FtpServer.Core;
using EnterpriseDT.Net.FtpServer.FileSystem;
using EnterpriseDT.Net.FtpServer.FileSystem.Windows;

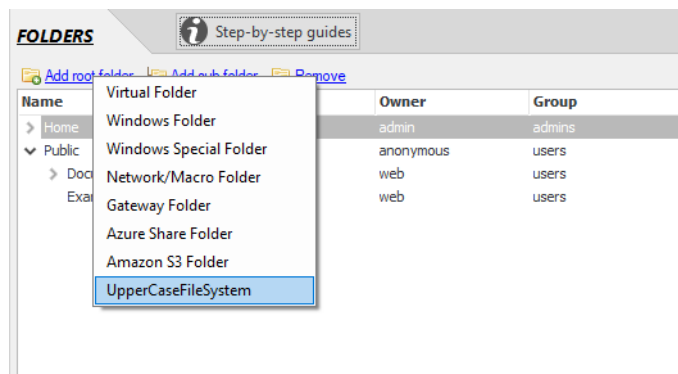
namespace SampleExtensions
{
    public class UpperCaseFileSystem : FolderAdapter
    {
        public override VirtualFileInfo[] GetFileInfos(ISession session, IFileSystemNode node, string path, string pattern)
        {
            VirtualFileInfo[] files = base.GetFileInfos(session, node, path, pattern);
            foreach (VirtualFileInfo file in files)
            {
                file.Name = file.Name.ToUpper();
            }
            return files;
        }
    }
}
```



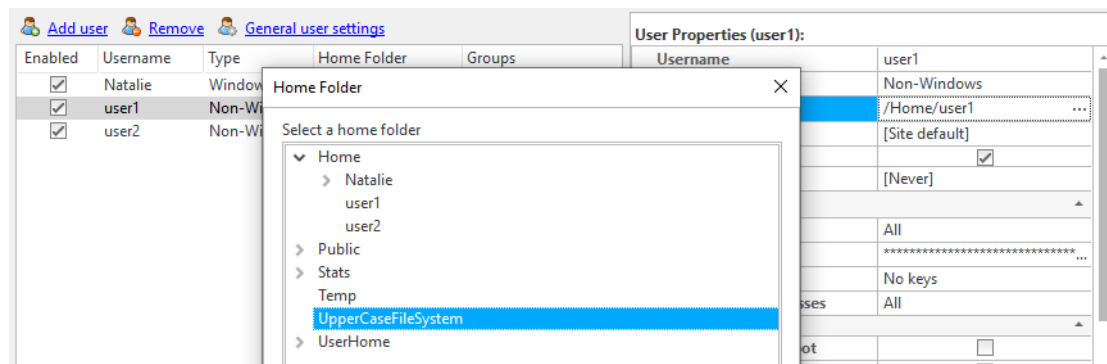
Adding UpperCaseFileSystem class into Extensions panel.



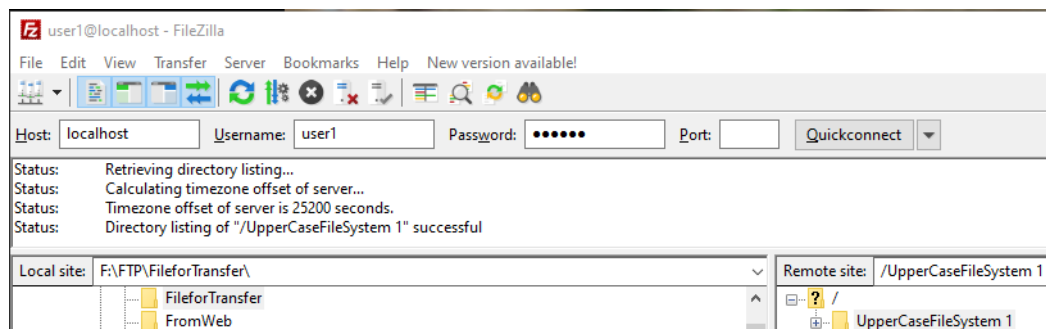
In the Folder panel, create a new folder with the new type which has been added above.



In the User panel, set the Home folder of the user to the newly created folder.



Login with File Zilla by using the user (which is set to the new type of folder). The login should be successful.



Then transfer a file which has a name which is not uppercase.

Filename	Filesize	Filetype	Last modified	Filename	Filesize	Filetype	Last modified	Permissions	C
..				..					
test-image1		JPG File	8/22/2019 3:03:06 ...	Empty directory listing					
test-image2		JPG File	5/4/2020 10:06:45 ...						
test-image3		JPG File	5/4/2020 10:07:01 ...						
test1.txt		Text Document	4/23/2020 3:15:24 ...						
test2.txt		Text Document	3/20/2020 5:12:34 ...						
test3.txt		Text Document	4/16/2020 1:16:32 ...						

Check to see if the transfer was successful and that the name of the transferred file was changed to uppercase.

Filename	Filesize	Filetype	Last modified	Filename	Filesize	Filetype	Last modified	Permissions	O
..				..					
test-image1.jpg	47,095	JPG File	8/22/2019 3:03:06 ...	TEST-IMAGE1.JPG	47,095	JPG File	5/5/2020 1:22:0...	-rwxrwx---	ai
test-image2.jpg	28,672	JPG File	5/4/2020 10:06:45 ...						
test-image3.jpg	7,580	JPG File	5/4/2020 10:07:01 ...						
test1.txt	1,272	Text Document	4/23/2020 3:15:24 ...						
test2.txt	2,471	Text Document	3/20/2020 5:12:34 ...						
test3.txt	816	Text Document	4/16/2020 1:16:32 ...						

### Creating an editor for a .NET file-system extension

A .NET file-system extension can optionally have a visual editor associated with it. This editor simply needs to pop up a form when a particular method is called. Extension editors must extend the class `EnterpriseDT.Net.FtpServer.FileSystem.FileSystemAdapterEditor`. They should be placed in the same assembly as the `FileSystemAdapter` extension itself. At a minimum they must override the `IsUserConfigurable` property and the `ShowConfigDialog` method. The `IsUserConfigurable` method should return `true`. The `ShowConfigDialog` takes an argument of type `IFileSystemNode`, which represents a folder in the virtual file-system. The job of the `ShowConfigDialog` method is to change the `IFileSystemNode`'s `Configuration` property and return `true` if it has changed it.

Every method of `FileSystemAdapter` takes an argument of the same type, i.e. `IFileSystemNode`. Thus every method will have access to the user-configurable `Configuration` property. This is how, for example, each Windows folder in the virtual file-system is mapped to a specific Windows folder on the hard-disk.

#### Example

The following example opens a `FolderBrowserDialog` and allows the user to select a Windows path. Notice that the `node.Configuration` property is only set if the OK-button was pressed by the user.

```

using System;
using System.Windows.Forms;
using EnterpriseDT.Net.FtpServer.Core;
using EnterpriseDT.Net.FtpServer.FileSystem;
using EnterpriseDT.Net.FtpServer.FileSystem.Windows;

namespace SampleExtensions
{
    public class UpperCaseFileSystem : FolderAdapter
    {
        public override VirtualFileInfo[] GetFileInfos(ISession session, IFileSystemNode node, string path, string pattern)
        {
            VirtualFileInfo[] files = base.GetFileInfos(session, node, path, pattern);
            foreach (VirtualFileInfo file in files)
            {
                file.Name = file.Name.ToUpper();
            }
            return files;
        }
    }

    public class UpperCaseFileSystemEditor : FileSystemAdapterEditor
    {
        public virtual bool IsUserConfigurable
        {
            get { return true; }
        }
        public override bool ShowConfigDialog(IWin32Window parent, IFileSystemNode node)
        {
            FolderBrowserDialog browser = new FolderBrowserDialog();
            if (node.Configuration != null)
            {
                browser.SelectedPath = node.Configuration;
            }
            if (browser.ShowDialog(parent) == DialogResult.OK)
            {
                node.Configuration = browser.SelectedPath;
                return true;
            }
            else
            {
                return false;
            }
        }
    }
}

```

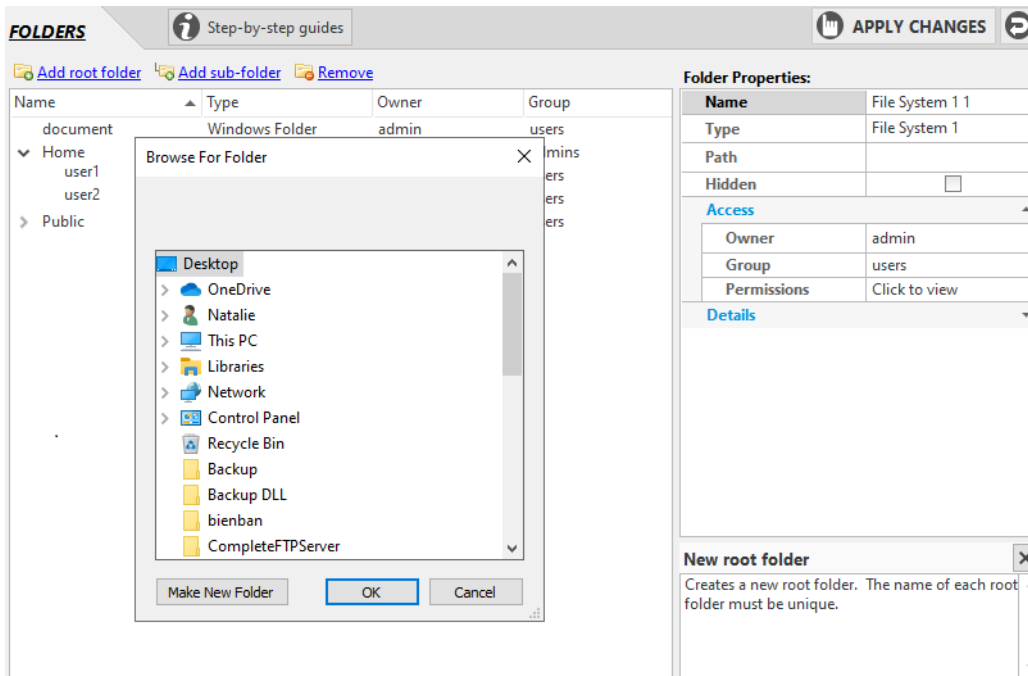
Note that after adding this extension into CompleteFTP, the user has to set the `Editor` class (including namespace) property of the extension in the CompleteFTP Manager to `SampleExtensions.UpperCaseFileSystemEditor`.

The screenshot shows the 'Extensions' tab in the CompleteFTP Manager. On the left, a list of extensions includes 'File System 1'. On the right, the configuration details for 'File System 1' are displayed. The 'Editor class (including namespace)' field is highlighted with a red rectangle and contains the value 'SampleExtensions.UpperCaseFileSystemEditor'. Below the configuration fields, there are two status messages: 'Set folder-type' with a folder icon and 'Activated successfully' with a green checkmark icon.

Extensions	
Name	File System 1
Type	File System
DLL location (use full path if not in Extensions directory)	F:\FTP\Extension\SampleExtensions\bin\Debug\SampleExtensions.dll
Extension class (including namespace)	SampleExtensions.UpperCaseFileSystem
Editor class (including namespace)	SampleExtensions.UpperCaseFileSystemEditor
Configuration	

Set folder-type 
 Activated successfully

Create a new folder with the new type which has been added above, it will show the `FolderBrowserDialog` and allows the user to select a Windows path.



All of the remaining steps are the same as above.

## Overview of File-System Adapter Classes

All file-system adapters must extend directly or indirectly from `FileSystemAdapter`. If you want your adapter to access the Windows file-system, then it's probably much easier to extend from one of its subclasses, such as `FolderAdapter`. For example, if you only want to implement some special mapping between the virtual file-system and the Windows file-system then you can simply override the `GetWindowsPath` method, which is responsible for mapping virtual paths into real paths.

### FileSystemAdapter

**namespace:** `EnterpriseDT.Net.FtpServer.FileSystem`

**editor:** None

Every file or directory operation that the user initiates is done by means of a `FileSystemAdapter`. This includes uploading, downloading, renaming and deleting files, as well as creating, listing and deleting directories. Most of these operations are implemented by a single method. The class itself can be instantiated, but does nothing of value. It yields empty listings and allows no uploads, downloads, deletes, etc. Any method or all methods may be overridden in subclasses to add functionality. As stated previously, it's often desirable to extend `FolderAdapter`, since it already implements all methods required for interfacing with the Windows file-system, but may be modified as required.

Most methods of `FileSystemAdapter` take at least the following three arguments: `session`, `node` and `path`. The `session` contains information about the user's current session; the `node` has information about the folder in the virtual file system that the user is accessing; and the `path` is the path relative to the last virtual folder on the path. So if, for example, the user is accessing the file `/a/b/c/d/e.txt`, which is in a subfolder of the virtual directory, `/a/b/c`, then the `path` argument will be `d/e.txt`. The value of the property `node.Configuration` is the value set by the adapter editor (see [above](#)), when the virtual folder was created by the user.

### WindowsAdapter extends FileSystemAdapter (abstract)

**namespace:** `EnterpriseDT.Net.FtpServer.FileSystem.Windows`

**editor:** None

Provides all the functionality necessary for CompleteFTP to access the Windows file-system **except** for the mapping of paths in the virtual file-system into real Windows paths. When it needs to map a path in the virtual file-system to a Windows path, it calls its own abstract method, `GetWindowsPath`. The main job of the subclasses of `WindowsAdapter` is to implement this method.

### FolderAdapter extends WindowsArchiveAdapter

**namespace:** `EnterpriseDT.Net.FtpServer.FileSystem.Windows`

**editor:** `FolderAdapterEditor`

This is the main adapter for the Windows file-system. The editor sets the `node.Configuration` property to an absolute Windows path. The `GetWindowsPath` concatenates this absolute Windows path, and the path passed to the `GetWindowsPath` method, to return the full Windows path of the folder in the Windows file-system.

### SpecialFolderAdapter extends WindowsArchiveAdapter

**namespace:** `EnterpriseDT.Net.FtpServer.FileSystem.Windows`

**editor:** `SpecialFolderAdapterEditor`

Adapter for a special Windows folder, such as the My Documents folder. The editor sets the `node.Configuration` property to a tag indicating which special folder is required. The `GetWindowsPath` returns the real Windows path of this folder, for the user whose session is being accessed.

## Overview of Other Classes

### ISession

**namespace:** EnterpriseDT.Net.FtpServer.Core

Provides information about the current user-session, including the name of the user and the IP address of the client.

### IFileSystemNode

**namespace:** EnterpriseDT.Net.FtpServer.FileSystem

A file-system node is a folder in the virtual file-system. Every folder that you see in the virtual file-system view has a single file-system node associated with it. The most important property of `IFileSystemNode` is `Configuration`, which stores the path or other configuration that the visual editor sets when the user configured the folder in the CompleteFTP Manager.

### VirtualFileInfo

**namespace:** EnterpriseDT.Net.FtpServer.FileSystem

Used to pass directory listing information from the `FileSystemAdapter` to CompleteFTP.

### VirtualFileMode

**namespace:** EnterpriseDT.Net.FtpServer.FileSystem

## Logging

The class `EnterpriseDT.Util.Debug.Logger` may be used to write messages to the CompleteFTP log. This can be very useful when debugging your adapter. To use this class, create a static instance of it in your adapter class and call it's `Error`, `Warn` or `Debug` method to write something to the CompleteFTP log file. For example:

```
using EnterpriseDT.Util.Debug;

public class UpperCaseFileNameAdapter : FolderAdapter
{
    private Logger log = Logger.GetLogger("UpperCaseFileNameAdapter");

    public override VirtualFileInfo[] GetFileInfos(ISession session, IFileSystemNode node, string path, string pattern)
    {
        log.Debug("Listing requested");
        ...
    }
}
```

## Class Reference

### Class: FileSystemAdapter

```
bool DirectoryExists(ISession session, IFileSystemNode node, string path)
    Returns true if a directory exists and false otherwise.

void CreateDirectory(ISession session, IFileSystemNode node, string path)
    Create a directory at the given path.

VirtualFileInfo[] GetFileInfos(ISession session, IFileSystemNode node, string path, string pattern)
    Return an array of VirtualFileInfo objects corresponding to the files in the given directory, that match the given pattern.

void DeleteDirectory(ISession session, IFileSystemNode node, string path)
    Delete the given directory.

bool FileExists(ISession session, IFileSystemNode node, string path)
    Returns true if a file exists and false otherwise.

void DeleteFile(ISession session, IFileSystemNode node, string path)
    Delete the given file.

bool IsFileLengthKnown(ISession session, IFileSystemNode node, string path)
    Returns true if the length of the given file is known and false otherwise.

long GetFileLength(ISession session, IFileSystemNode node, string path)
    Returns the length of the given file in bytes.

DateTime GetModifiedTime(ISession session, IFileSystemNode node, string path)
    Returns the time when the given file was last modified.

void SetModifiedTime(ISession session, IFileSystemNode node, string path, DateTime dateTime)
    Sets the last-modified time of the given file.

Stream GetReadStream(ISession session, IFileSystemNode node, string path)
```

Returns a stream that may be used for reading the given file (used for downloads).

```
void OnReadStreamClosed(ISession session, IFileSystemNode node, string path, Stream stream)
```

Called when the read-stream (created by `GetReadStream`) is closed.

```
Stream GetWriteStream(ISession session, IFileSystemNode node, string path, VirtualFileMode mode)
```

Returns a stream that may be used for writing to the given file (used for downloads).

```
void OnWriteStreamClosed(ISession session, IFileSystemNode node, string path, Stream stream)
```

Called when the write-stream (created by `GetWriteStream`) is closed.

```
bool MoveFile(ISession session, IFileSystemNode fromNode, string fromPath, FileSystemAdapter toAdapter, IFileSystemNode toNode, string toPath)
```

Move the given file from one location in the virtual file-system (defined by `fromNode` and `fromPath`) to another location (defined by `toAdapter`, `toNode` and `toPath`). For simple file renaming, the `fromNode` will be the same as the `toNode` and the `toAdapter` will be the 'this' adapter.

```
string GetAdapterFilePath(ISession session, IFileSystemNode node, string path)
```

```
void Initialize(IPlugInInfo info)
```

Called when the extension is first created.

```
void Dispose()
```

Called when the extension is disposed of.

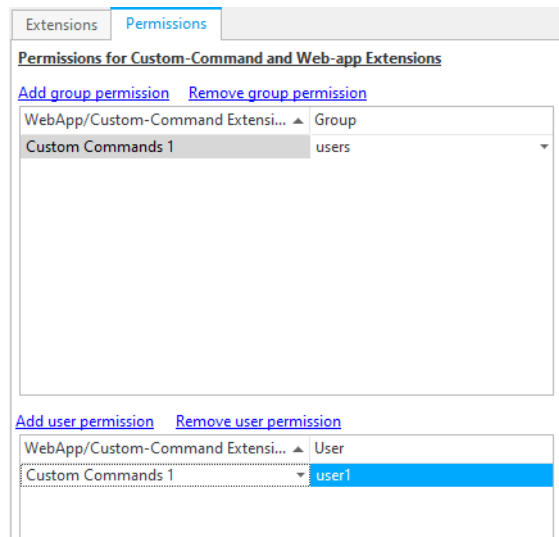
# .NET custom command extensions

A .NET custom command extension is a .NET assembly (usually a DLL) that contains a class that extends the class `EnterpriseDT.Net.FtpServer.Core.CustomCommands`. The assembly can be developed in C#, VB.NET or any other .NET language. If you don't already have Visual Studio then you can use one of Microsoft's free Visual Studio Express or Visual Studio Community products.

## Creating a .NET custom command extension

General instructions on building CompleteFTP extensions may be found [here](#). In addition to the registration steps described in the general instructions, permissions must also be granted for users and/or groups to access a .NET custom command extension, otherwise no users will be able to invoke the commands.

Permissions for .NET custom command extensions are managed in the Permissions tab of the Extensions panel in the CompleteFTP Manager, as shown below. The commands in a .NET custom command extension may be executed by a user that's been directly given, or is a member of a group that has been given permission to do so.



Commands are added to a .NET custom command extension simply by adding methods. By default the name of the method becomes the name of the command. If a `CustomCommandName` attribute is defined, then that becomes the name of the command. All names are case-insensitive. The arguments of the method become the arguments of the command. All the arguments must be strings. Custom commands may return one or more lines of text.

## Example

The following example implements the ubiquitous 'Hello World' example as a Custom command:

```
using EnterpriseDT.Net.FtpServer.Core;

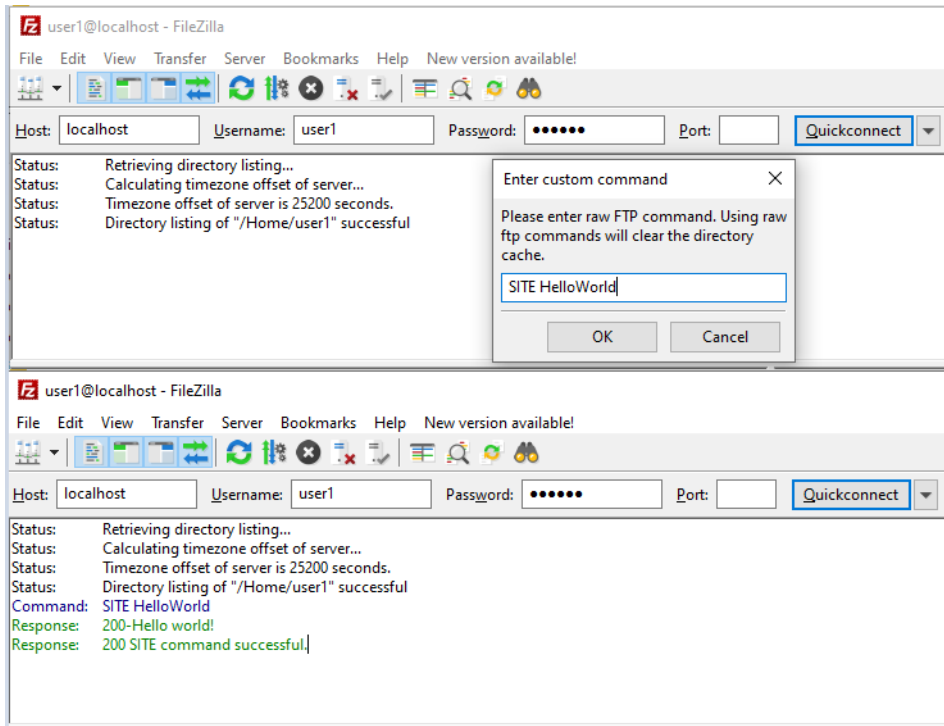
namespace SampleExtensions
{
    public class HelloWorldSiteCommands : SiteCommands
    {
        public string HelloWorld()
        {
            return "Hello world!";
        }

        [SiteCommandName("HelloWorld2")]
        public string HelloWorldWithArgs(string argument1, string argument2)
        {
            return "Hello world!\n" + argument1 + "\n" + argument2;
        }
    }
}
```

Log in using the FTP client (FileZilla for example) with the user who has permission set as shown below:

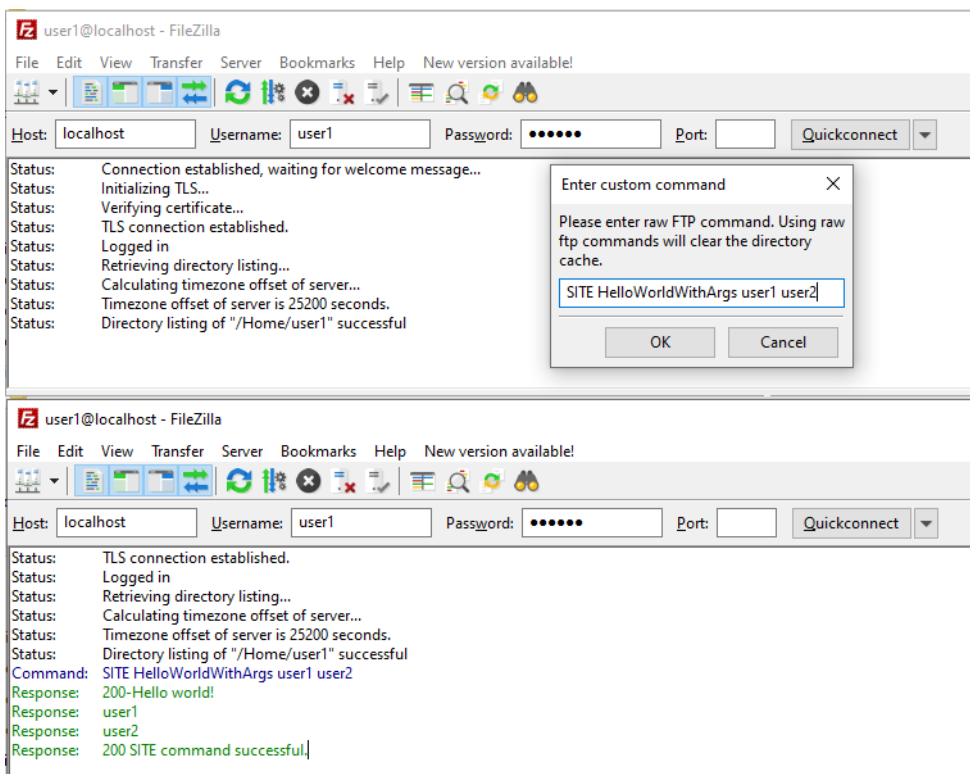


The way in which a command is executed by a client depends on the protocol. With FTP custom commands, they must be preceded by SITE. For example, the first method in the example above can be executed as follows:



The first method has no arguments and defines no CustomCommandName attribute. This means that it must be called by its full name, "HelloWorld" (case is ignored).

The second method has two arguments, so two arguments must be entered when the command is executed by the client. This method also defines a command-name, so that name, "HelloWorldWithArgs", must be used when the command is executed by the client. In addition, it illustrates how multiple lines may be included in a reply.



Note that if you're using Microsoft ftp.exe then you need to enter "quote site mypassword".



# .NET Event Extensions

A .NET Event Extension is a .NET assembly (usually a DLL) that contains a class that extends the class `EnterpriseDT.Net.FtpServer.Trigger.Trigger`. The assembly can be developed in C#, VB.NET or any other .NET language. If you don't already have Visual Studio then you can use one of Microsoft's free Visual Studio Express or Community products.

## Creating a .NET Event Extension

General instructions on building CompleteFTP extensions may be found [here](#).

.NET Event Extension classes **must** extend `EnterpriseDT.Net.FtpServer.Trigger.Trigger` and **implement** the `OnTriggerEvent` method. This method is called each time one of the following events occur: `LogIn`, `LogOut`, `ChangeDirectory`, `UploadFile`, `DownloadFile`, `MoveFile`, `DeleteFile`, `CreateFolder`, `DeleteFolder` and `MoveFolder`.

The `OnTriggerEvent` is passed a single argument of type `TriggerEvent`. This class contains various information about the event that occurred and the user-session in which it occurred.

## Example

The following example implements a minimal auditing .NET Event Extension, which simply logs all events to a CSV file in the user's temporary directory. This is based on the value of environment variables "TEMP" of the system. The default path for this folder is `C:\Users\{username}\AppData\Local\Temp` or `C:\Windows\Temp`.

```
using System;
using System.IO;
using EnterpriseDT.Net.FtpServer.Trigger;

public class AuditingEventHandler : Trigger
{
    public override void OnTriggerEvent(TriggerEvent eventInfo)
    {
        string auditFilePath = Path.Combine(Path.GetTempPath(), "audit.csv");
        using (StreamWriter auditFile = File.AppendText(auditFilePath))
        {
            auditFile.WriteLine("{0:yyyy-MM-dd hh:mm:ss},{1},{2},{3},{4}",
                DateTime.Now, eventInfo.EventType, eventInfo.Session.UserName,
                eventInfo.Path, eventInfo.OldPath);
        }
    }
}
```

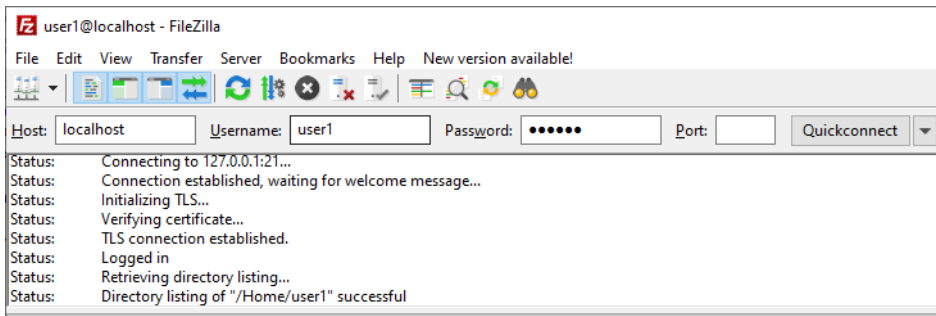
Adding **AuditingEventHandler** class into the CompleteFTP manager/Extension panels.

The screenshot shows the 'Extensions' tab of the CompleteFTP manager. On the left, a list of extensions includes 'Event 1'. The right pane displays the configuration for 'Event 1'.

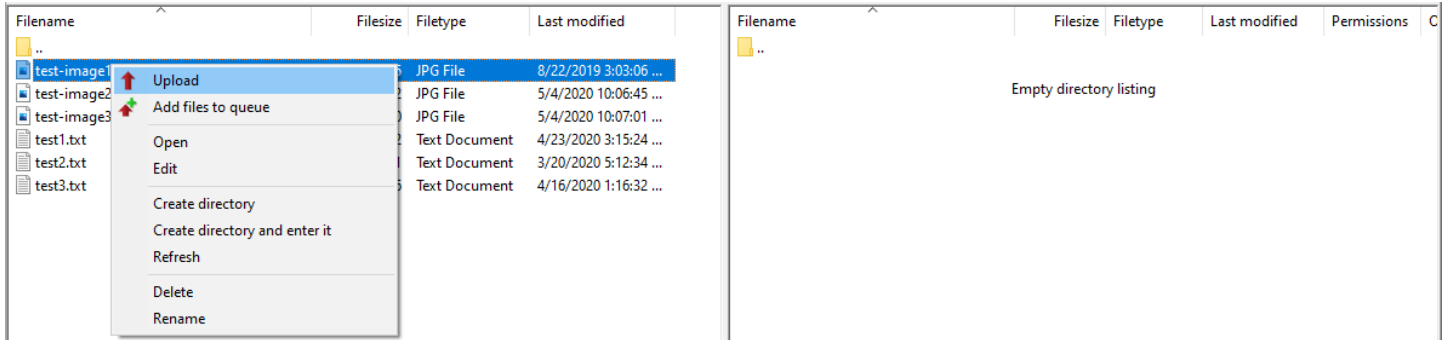
Name	
Event 1	
Type	
Event	
DLL location (use full path if not in Extensions directory)	
F:\FTP\Extension\SampleExtensions\SampleExtensions\bin\Debug\SampleExtensions.dll	
Extension class (including namespace)	
SampleExtensions.AuditingEventHandler	
Editor class (including namespace)	
Configuration	

Not yet activated

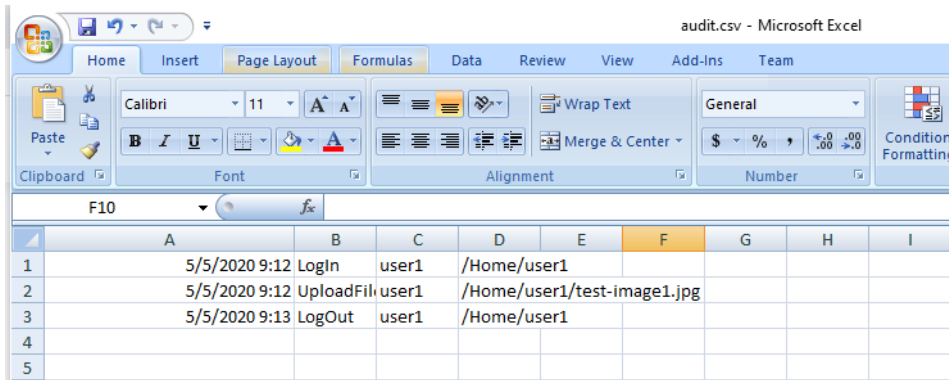
Login to your **FTP client** (FileZilla for example).



Then upload a file.



Check in C:\Users\{username}\AppData\Local\Temp or C:\Windows\Temp to see if the file `audit.csv` is there and that it has written the Login and Upload events.



## Class Reference

### Class: Trigger

All .NET Event Extensions must extend this class. Only the `OnTriggerEvent` method needs to be implemented. It is called every time an event occurs. If any initialization needs to be done then that may be placed in the `Initialize` method, and any required clean-up code may be placed in the `Dispose` method.

```
void OnTriggerEvent(TriggerEvent eventInfo)
```

Called every time an event occurs.

```
void Initialize(IPlugInInfo info)
```

Called when the Trigger is created. Perform any initialization tasks here. `info.PlugInConfiguration` contains the configuration entered by the user.

```
void Dispose()
```

Called when the Trigger is disposed of. Perform any clean-up tasks here

### Class: TriggerEvent

Contains information describing a particular event

```
TriggerEventType EventType
```

Type of event. Possible values are `LogIn`, `LogOut`, `ChangeDirectory`, `UploadFile`, `DownloadFile`, `MoveFile`, `DeleteFile`, `CreateFolder`, `DeleteFolder` and `MoveFolder`.

```
ISession Session
```

Provides information about the current user-session, including the name of the user and the IP address of the client.

```
string Path
```

Path of file (if any)

```
string OldPath
```

Previous path of file (valid for `MoveFile` and `MoveFolder` events only).

# .NET IP Filter Extensions

Use this type of extension to implement a custom IP filtering scheme. The server calls a method in your class to find out whether or not a particular IP address should be allowed or denied.

## Creating a .NET IP Filter Extension

General instructions on building CompleteFTP .NET extensions may be found [here](#).

.NET IP Filter Extension classes must implement the `EnterpriseDT.Net.FtpServer.Core.IPFilter` interface, which has one method:

```
IPFilterAction GetFilterAction(IPAddress address)
```

The `IPAddress` argument is the IP address of the client attempting to make a connection. The return-value can have any of three values: 1) `IPFilterAction.Deny` will deny the connection, 2) `IPFilterAction.Allow` will allow the connection, unless it has been auto-banned due to excessive authentication failures, and 3) `IPFilterAction.AllowAlways` will allow the connection irrespective of the number of authentication failures.

## Example

Below is the source code for an IP Filter which supports denying the IP address if it is not from Australia.

```

using System;
using System.IO;
using System.Net;
using System.Text.RegularExpressions;

using EnterpriseDT.Net.FtpServer.Core;

namespace TestIPFilter
{
    public class TestIPFilter : IIPFilter
    {
        private string reason;

        public IPFilterAction GetFilterAction(IPAddress address)
        {
            var accessKey = "//access key";

            var httpWebRequest = (HttpWebRequest)WebRequest.Create("http://api.ipstack.com/"
                + address + "?access_key=" + accessKey + "&format=1");
            httpWebRequest.ContentType = "application/json";
            httpWebRequest.Method = "GET";

            var httpResponse = (HttpWebResponse)httpWebRequest.GetResponse();
            using (var streamReader = new StreamReader(httpResponse.GetResponseStream()))
            {
                var result = streamReader.ReadToEnd();
                if (GetCountryCode(result) == "AU")
                {
                    reason = null;
                    return IPFilterAction.Allow;
                }
                else
                {
                    reason = "Outside Australia";
                    return IPFilterAction.Deny;
                }
            }
        }

        private string GetCountryCode(string result)
        {
            var match = Regex.Match("\"country_code\":[ \t]*\"([^\"])+\"", result);
            return match.Success ? match.Groups[1].Value : null;
        }

        public string Reason => reason;

        public void Initialize(IPluginInfo info)
        {
        }

        public void Dispose()
        {
        }
    }
}

```

Note: To get the `access key` (which is applied in the example code), please visit this service [ipstack.com](http://ipstack.com).

The following example shows the way to apply the code above:

1. Add the example code 1 into the .NET IP Filter Extension in the Extensions panel of the CompleteFTP Manager.

Extensions

Permissions

Add extension

Remove

Name

IP filter 2

Type

IP filter

DLL location (use full path if not in Extensions directory)

F:\FTP\Extension\SampleExtensions\bin\Debug\SampleExtensions.dll

Extension class (including namespace)

TestIPFilter.TestIPFilter

Editor class (including namespace)

Configuration

Activated successfully

- From the second server, connect to server 1. It will not connect successfully.

FileZilla

File

Edit

View

Transfer

Server

Bookmarks

Help

New version available!

Host: Server 1

Username: user1

Password:

Port:

Quickconnect

Status: Connection established, waiting for welcome message...

Error: Connection closed by server

Error: Could not connect to server

# How to add logging to a .Net extension

The class `EnterpriseDT.Util.Debug.Logger` may be used to write messages to the CompleteFTP log. This can be very useful when debugging your .Net extension. To use this class, create a static instance of it in your extension class and call its `Error`, `Info`, `Warn` or `Debug` method to write something to CompleteFTP's log file. For example:

```
using EnterpriseDT.Util.Debug;

public class UpperCaseFileNameAdapter : FolderAdapter
{
    private static Logger log = Logger.GetLogger("UpperCaseFileNameAdapter");

    public override VirtualFileInfo[] GetFileInfos(ISession session, IFileSystemNode node, string path, string pattern)
    {
        log.Debug("Listing requested");

        ...
    }
}
```

# Scripting Overview

Scripting Guide includes all topics which related to JSS.

1. **JSS (Javascript Server-Side)**: provides JSS API Guide, JSS API Reference and JSS Remote Procedure Calls (RPC).
2. **Web-Apps**: summarizes the basic concepts of developing JSS web-apps.
3. **Events-How to write JSS process triggers**: provides process trigger events written in JSS (Javascript Server-Side).
4. **Extentions-JSS Extensions**: provides extensions written in JSS (Javascript Server-Side).
5. **Sample scripts**: provides sample scripts written in JSS (Javascript Server-Side).

# JSS API Reference

CompleteFTP is highly customizable allowing users to extend its inbuilt functionality using JSS, as well as .NET assemblies, batch scripts, Powershell scripts and executables.

JSS (Javascript Server-Side) is the name given to CompleteFTP's scripting environment. It includes a Javascript language interpreter and an API for integrating with CompleteFTP as well as performing common tasks.

## Applications of JSS

JSS may be used to implement:

- **Process triggers** that are executed when one or more specified events occur within CompleteFTP, such as a user logging in, a client uploading a file or a scheduled timer ticking over.
- **IP filter extensions** facilitate the addition of custom rules for blocking connections from specified IP addresses (e.g. IP addresses from particular countries).
- **Authentication extensions** allowing CompleteFTP to authenticate via a source other than those that are already built-in (i.e. internal users, Windows users, databases and remote FTP/SFTP servers).
- **File-system extensions** which implement custom file systems that may, for example, store files in storage other than file-systems, e.g. databases.
- **Custom-command extensions** contain user-defined functions that may be invoked by clients via FTP (via the SITE command), SFTP (via SSH\_FXP\_EXTENDED), SSH terminal commands) and HTTP/S (via GET or POST).
- **Web applications** that may be accessed by regular users just like any other web application. EnterpriseDT's own customer portal, [edtConnect](#), is a JSS web-app that's hosted on CompleteFTP. It accepts electronic payments and manages licenses for many thousands of users

## JSS API

The JSS API is the set of functions, services and classes available for integrating with CompleteFTP and for performing common operations, such as HTTP requests, sending e-mails and accessing databases.

## JSS API Services

JSS API services in CompleteFTP are global objects containing properties and functions. Their names start with a lower-case letter, e.g. console, http, mail. Since they are global objects, they may be used without first instantiating an object. For example, the statement, `console.log("Hello")`, logs a message to the CompleteFTP logs. The services are listed in the left side-bar.

## JSS API Classes

JSS API classes provide features such as FTP/SFTP clients, virtual file-system access and database access. The names of JSS API classes begin with an upper-case letter. The classes are listed in the left side-bar.

## JSS API Functions

The JSS API provides two global functions: [include](#) and [publish](#). These allow the developer to reuse code between script files and control which functions are exposed via [RPC](#).

## Integrating with .NET

JSS offers two methods of integrating with .NET:

- **Inline use of .NET libraries** - in JSS process triggers and all types of JSS extensions, .NET classes may be instantiated directly from JSS, for example, `new System.Text.StringBuilder()` will instantiate the .NET class, `StringBuilder`. This object's properties and methods may be accessed in the same way as those of native Javascript objects. The names of classes must be fully qualified. This usage is disallowed in web apps for security reasons.
- **JSS-to-.NET bridge extensions** - any .NET class that has a public constructor may be registered as a JSS-to-.NET extension. Once this has been done, an instance of it will be created when a JSS script is invoked by CompleteFTP. The name of this instance will be the same as the registered name of the extension. So if, for example, a class called `MyNamespace.MyClass` containing a method called `MyMethod` is registered in CompleteFTP Manager with the name `myStuff`, then the method may be called from JSS using the expression `myStuff.MyMethod()`. The location of the assembly containing this class is specified upon registration.

## ECMAScript compliance

JSS's Javascript interpreter is compliant with [ECMAScript 3](#) (ES3), but includes some ES5 features such as inbuilt JSON support, and extra methods on Array, Date and String objects.

## JSS API

### Services

- [console](#)
- [http](#)
- [mail](#)
- [request](#)
- [response](#)
- [system](#)
- [templater](#)



**Classes**

- [Config](#)
- [DatabaseSync](#)
- [File](#)
- [FileNameFilter](#)
- [Folder](#)
- [FolderManager](#)
- [Ftp](#)
- [Group](#)
- [GroupManager](#)
- [IPFilter](#)
- [IPFilterRule](#)
- [JsonDB](#)
- [KeyPair](#)
- [Members](#)
- [PasswordPolicy](#)
- [ServerMapping](#)
- [Site](#)
- [SiteMapping](#)
- [SQLResultSet](#)
- [SQLTransactionSync](#)
- [SSLCertificate](#)
- [User](#)
- [UserManager](#)
- [WindowsFile](#)

**Functions**

- [include](#)
- [publish](#)

# JSS RPC Reference

CompleteFTP makes it very easy to call JSS functions as Remote Procedure Calls (RPC). It does this by automatically generating client-side proxies for calling server-side functions, meaning that adding a remotely invocable server-side function is as easy as adding a client-side function.

## Which functions are available via RPC?

Functions declared in the global scope and those referenced by the `publish` function (without the `serverOnly` flag) are automatically made available via RPC when the URL of the `.jss` file in which they're defined are referenced by allowed `script` tag in HTML. The Javascript code required to call them is automatically generated by CompleteFTP (see [here](#)).

The following example illustrates some usages:

```
function globalFunction() {  
}  
  
(function () {  
    function publishedFunction() {  
    }  
  
    function serverOnlyFunction() {  
    }  
  
    function internalFunction() {  
    }  
  
    publish("publishedFunction", publishedFunction);  
    publish("serverOnlyFunction", serverOnlyFunction, { serverOnly: true });  
})();
```

Both `globalFunction()` and `publishedFunction()` are available for RPC, the former because it's a global function and the latter because it's published using the `publish` function. Neither `internalFunction()` and `serverOnlyFunction()` are not available via RPC, but `serverOnlyFunction()` may be called by any other `.jss` file that includes it using the `include` function because it's published with the `serverSide` flag.

## Calling via RPC

Two forms of RPC are supported: JSON-RPC and form-based RPC.

### JSON-RPC

CompleteFTP uses [JSON-RPC 2.0](#) for client-server communications. JSON-RPC is a minimalist JSON-based RPC protocol. For example, the call to the `hello()` function in the [Introduction to JSS](#) is marshalled as

```
{"jsonrpc": "2.0", "method": "hello", "params": ["world"], "id": 1}
```

which returns the result in the following form:

```
{"jsonrpc": "2.0", "result": "Hello world from the server", "id": 1}
```

Errors are returned as:

```
{"jsonrpc": "2.0", "error": {"code": 0, "message": "An error occurred"}, "id": 1}
```

### Form-based RPC

CompleteFTP also supports form-based RPC via standard HTTP GET and POST. If an element named `method` is included in the form then the Javascript function with that name will be called. The arguments in the call will match the elements in the form - in the same order. Thus the `hello()` function in the [Introduction to JSS](#) may be called by HTTP-POST using the following HTML:

```
<html>  
<body>  
  <form method=post>  
    <input type=hidden name=method value=hello>  
    <input type=text name=firstName>  
    <input type=submit>  
  </form>  
</body>  
</html>
```

Alternatively it may be invoked by HTTP-GET via the following URL:

`http://hostname/directory/script.jss?method=hello&firstName=world`

In both these cases the result is returned in the JSON-RPC format, i.e.

```
{"jsonrpc": "2.0", "result": "Hello world from the server", "id": 1}
```

# JSS Web-Apps

CompleteFTP (Professional and Enterprise MFT) includes everything you need to host your own web-apps written in Javascript (**JSS**).

Creating a custom web-app for CompleteFTP's HTTP server offers numerous benefits to users. A tailored web-app can enhance the user experience when accessing files, reduce administrative tasks and enable integration with external systems. By leveraging the power of JSS and the available APIs, users can create web-apps that cater to their specific needs and workflows, such as advanced file searches, custom reporting, or automated file operations. Furthermore, a well-designed web-app can seamlessly integrate with other systems or services, enabling greater interoperability and expanding the capabilities of CompleteFTP beyond its core file-server functionality. Writing a custom web-app empowers users to take full advantage of CompleteFTP's HTTP server, allowing for a more personalized and effective solution to their file management and server administration needs.

## Examples

- Custom file management interface:** Develop a web-app that offers a bespoke file management interface, tailored to the organization's branding and workflow requirements. This can include customized navigation, file previews, or specific file sorting options. Start with the source-code CompleteFTP's existing file-manager (available to users upon request), or write your own.
- Access control and user management:** Create a web-app to simplify and streamline user management, including permissions and access control. This can include group-based permissions, department-specific restrictions, or time-limited access to certain resources.
- File approval and review process:** Implement a web-app that facilitates a document approval and review workflow, enabling users to submit files for review, track changes, and receive approval notifications.
- Automated file processing:** Develop a web-app that automates specific file processing tasks, such as resizing images, converting file formats, or extracting metadata for indexing and searching purposes.
- Advanced search capabilities:** Implement a web-app that offers advanced file search capabilities, enabling users to locate files based on metadata, content, or custom-defined criteria.
- Integration with third-party services:** Develop a web-app that integrates the file server with other systems or services, such as CRM, ERP, or project management tools, to improve overall organizational efficiency and collaboration. This can be combined with CompleteFTP's custom **JSS process triggers** to create a powerful solution.
- Custom notifications and alerts:** Create a web-app combined with **JSS process triggers** to manage custom notifications and alerts.
- Secure file sharing:** Implement a web-app that enhances CompleteFTP's existing secure file sharing experience.
- Data backup and archiving:** Develop a web-app that automates data backup and archiving processes, ensuring that critical files are securely stored and easily retrievable in case of data loss or system failure.

## JSS Web Services

JSS web-apps can also be utilized as web APIs or services that can be called from other software, providing a versatile way to extend the functionality of CompleteFTP's HTTP server. By creating server-side JavaScript applications that expose custom APIs, organizations can integrate CompleteFTP with other systems and services, facilitating seamless data exchange and improving overall efficiency.

## Examples

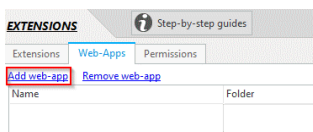
Using JSS web-apps as web APIs or services enables organizations to:

- Automate file operations:** Develop an API that allows external systems to upload, download, or manipulate files on the CompleteFTP server programmatically. For example, an organization's content management system (CMS) can automatically push new assets to the file server or retrieve files for processing.
- Remote user management:** Create a web API that allows external applications to manage user accounts and access permissions, enabling seamless integration with identity management systems or HR tools.
- Custom data validation:** Implement a web service that validates uploaded files based on custom criteria, such as file format, size, or content. This can be useful for ensuring data quality and compliance with organizational policies.
- File metadata extraction:** Develop a web API that extracts metadata from uploaded files and sends it to external systems for indexing, searching, or analysis.
- Secure data exchange:** Implement a web API that facilitates secure data exchange between the file server and external partners or clients. This can include encrypting data, generating secure tokens, or implementing OAuth for authentication.

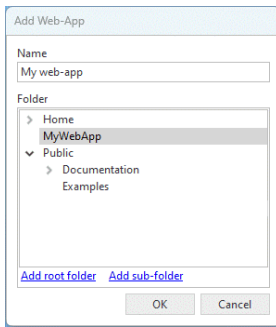
## Instructions:

To add a JSS web-app extension, follow these steps:

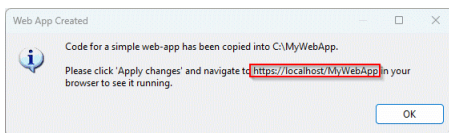
- Open CompleteFTP Manager and navigate to the Extensions panel.
- Select the 'Web-apps' tab and click 'Add web-app'.



- In the dialog, enter the name and select a folder for the web-app.



4. If you need to create a new folder, select the option to do so. The folder must be a Windows folder, so all other folder types are disabled.
5. Click OK to create the web-app.
6. You will be prompted to choose whether the web-app should be accessible to non-authenticated users. Click Yes or No, as needed.
7. After clicking Yes or No, the web-app will be created. If the folder you chose or created does not contain an index.html or index.jss file, a simple web-app with index.jss and template.html files will be created in the folder. Note that the path of a web-app can't be beneath another web-app.
8. A message box displaying the URL of the web-app will appear. Click OK to close the message box.



9. Click Apply changes in the main window.
10. Now you can open the web-app in a browser using the URL.

## Developing JSS Web-Apps

The sample web-app that's generated by CompleteFTP consists of two files `index.jss` and `template.html`

**index.jss:**

```
function processRequest() {
    var data = {
        siteName: system.site.label
    };
    response.writeUsingTemplateFile("template.html", data);
}
```

**template.html:**

```
<html>
<body>
<h1>Welcome to {{siteName}}</h1>
</body>
</html>
```

You can edit these files to start building your own web-apps. To start learning please continue to the [JSS Web-App Basics](#) topic.

## Understanding .jss file execution

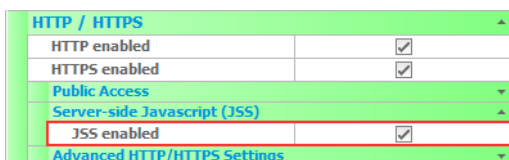
While `.jss` files are executed by default, it is possible to configure CompleteFTP such that they don't get executed. We therefore explain how to control `.jss` file execution in this section.

A file with a `.jss` extension will be run as JSS if

1. JSS is enabled on the site serving up the page, AND
2. JSS is enabled for the either:
  - a. the logged in user, OR
  - b. user who owns the folder that contains the file

If these conditions are not satisfied then the file will be served up the same as an non-JSS file.

JSS is enabled for a site by checking the *JSS enabled* checkbox in the *Site Settings* category, *HTTP/HTTPS -> Server-side Javascript (JSS)*:



JSS is enabled for a user by checking the *JSS (Server-side Javascript)* checkbox in the *Site Properties* category, *Scripting and Shells*:

User Properties (MaryJohnson):	
Username	MaryJohnson
User type	Non-Windows
Home folder	/Home/MaryJohnson
Enabled	<input checked="" type="checkbox"/>
Expires	[Never]
Authentication	▼
File-System	▼
Allowed Protocols	▼
File-Sharing	▼
Quotas and Limits	▼
Scripting and Shells	▲
JSS (Server-Side Javascript)	<input checked="" type="checkbox"/>
SSH terminal shell	UNIX
Personal Details	▼

Note that for JSS web-apps to be available to unauthenticated users, JSS must be enabled for the owner of the `/Public` folder, i.e. `anonymous`, which it isn't by default.

For more information on how to host a web-app in CompleteFTP, please refer to Step-by-step guide: [Host a web-app in CompleteFTP](#) .

# JSS Web-App Basics

This article will summarize the basic concepts of developing JSS web-apps. It starts with the traditional page-request style of programming, in which entire pages are requested from the server; it then moves onto AJAX-style Remote Procedure Calls (RPC's) used in [Single-Page Applications \(SPA's\)](#)

1. [Where to place JSS code in a web-app](#)
2. [Serving a basic request](#)
3. [Handling user input](#)
4. [Building responses using templates](#)
5. [Calling a server-side function via RPC](#)
6. [Combining simple requests and RPC](#)
7. [Asynchronous RPC](#)
8. [Uploading and downloading files](#)
9. [Including other JSS files](#)
10. [Controlling accessibility of JSS functions](#)
11. [Extending JSS using .NET](#)
12. [Exploring Further](#)

For information about the API available to JSS developers, please refer to the [JSS API Guide](#).

## 1. Where to place JSS code in a web-app

Assuming that JSS is enabled (see [here](#)), JSS code will be executed for an HTTP request if

1. It's in a file that's directly referenced by the request, e.g. `http://myhost/mydir/myfile.jss` and the user has permission to read the file or
2. It's in a file named `index.jss` and the request is for any non-existent resource in the same directory or a subdirectory of the file, and the user has permission to read the file and all directories between the file and the requested resource

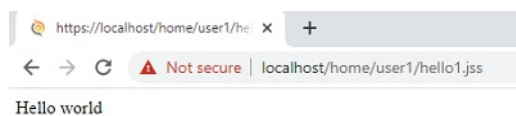
Note that option 2, above, makes it possible to implement [REST API's](#) using JSS. The `request.restPath` property, allows the script to ascertain which resources were requested.

## 2. Serving a basic request

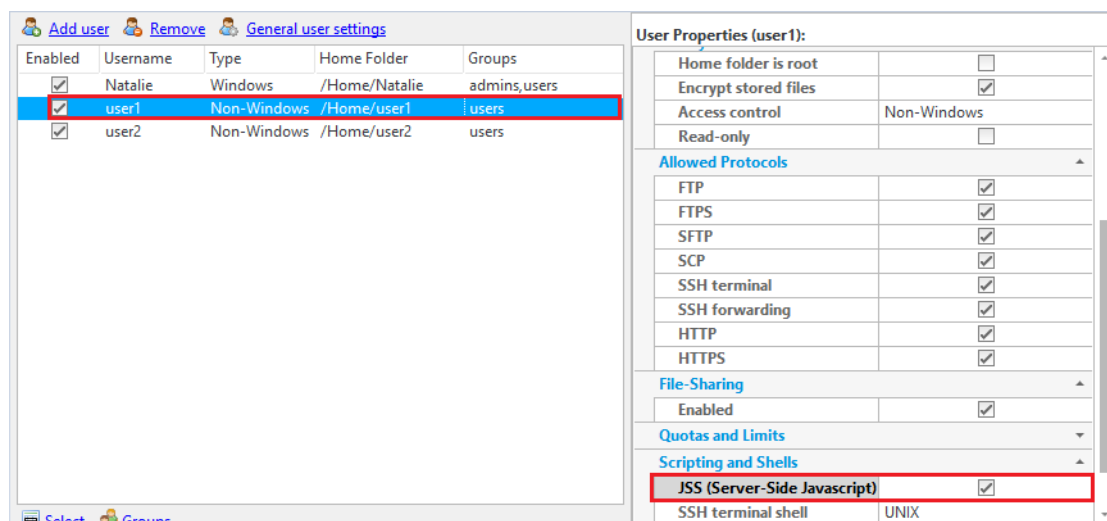
A response to a basic HTTP request may be made, by calling the `write()` method of the `response` object, as follows:

```
response.write("<body><html>Hello world</html></body>");
```

which will respond with an HTML page displaying:



Note: User has to enable the JSS for the login user.

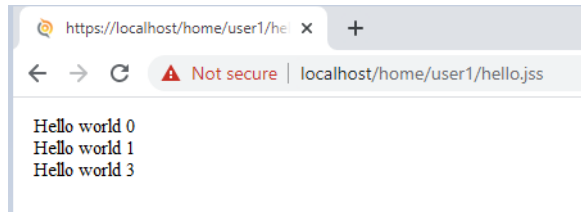


JSS is ECMAScript 3 compliant, so nearly all Javascript syntax which is in common use is supported, for example:

```
function _getMessage(i) {
    return "Hello world " + i;
}

var html = "<html><body>";
for (var i=0; i<3; i++)
    html += _getMessage(i) + <br>
html += "</body></html>";
response.write(html);
```

Here a function is called when building the HTML response. The underscore is prepended to the function to indicate that it's not to be called via RPC. This script will output:

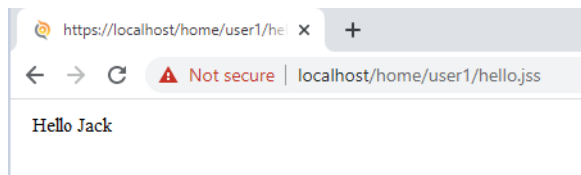


### 3. Handling user input

User-submitted form data and URL arguments may be accessed via the `request` variable. For example, if a form with a text-box called `firstName` is submitted, then the following may be used to access the value of this field:

```
var firstName = request.form.firstName;
response.write("<body><html>Hello " + firstName + "</html></body>");
```

which, if the value of the `firstName` field is `Jack`, will return:



### 4. Building responses using templates

JSS supports **templates**, which facilitates the separation of Javascript and HTML code into separate files.

For example, if a template file called `hello1.html` contains the following:

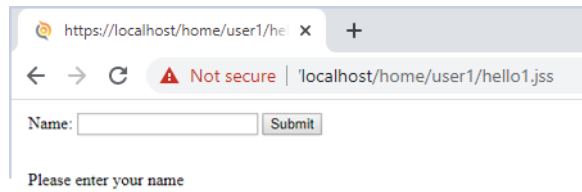
```
<html>
<body>
<form method=post>
  Name: <input type=text name=firstName value={{firstName}}>
  <input type=submit>
</form>
<br>
{{message}}
</body>
</html>
```

and a JSS file called `hello1.jss` contains:

```
var data = { };
if (request.form.firstName != undefined) {
    data.firstName = request.form.firstName;
    data.message = "Hello " + request.form.firstName;
} else {
    data.firstName = "";
    data.message = "Please enter your name";
}
response.writeUsingTemplateFile("hello1.html", data);
```

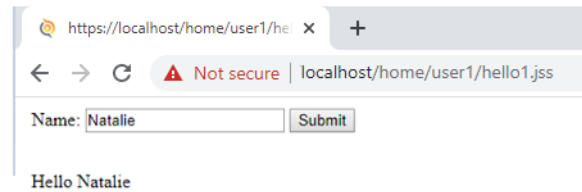


then a request to `hello1.jss` will return:



Notice that the tag `{{message}}` in `hello1.html` was substituted by the field with the same name, `message`, of the Javascript object that was passed to the `writeUsingTemplateFile(file, data)` method.

Once the field has been filled as 'Natalie' and the button clicked, the following will be returned:



If a template is to be provided as a string rather than a file, then the `writeUsingTemplateString(template, data)` may be used.

## 5. Calling a server-side function via RPC

Clients may invoke server-side Javascript functions, simply by calling them (or, to be precise, their automatically generated proxies). For example, say the server has a JSS file called `hello2.jss` with the following contents:

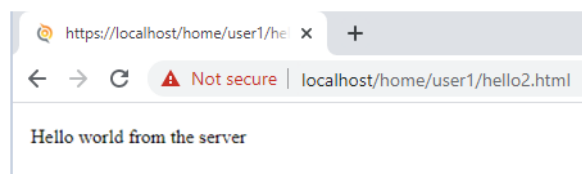
```
function hello(name) {  
    return "Hello " + name + " from the server";  
}
```

This is Javascript code that will be run on the server.

Now, say we have an HTML file called `hello2.html` with the following contents:

```
<html>  
<body>  
<script src="hello2.jss"></script>  
<script>  
    var message = hello("world");  
    document.write(message);  
</script>  
</body>  
</html>
```

When the HTML is rendered in the browser, the result will be:



This message is **generated on the server by the Javascript function** and returned by the call to the `hello()` function. Notice that all code, client-side and server-side, is Javascript and that no AJAX code was written. This is possible because the request to `hello2.jss` returns the Javascript code required to remotely invoke the `hello()` function in `hello2.jss`.

The Javascript code that's returned by the request to `hello2.jss` is compressed for efficiency. The uncompressed code may be viewed by a request to `hello2.jss?proxy=uncompressed`. It contains the auto-generated proxy-code that the browser needs to call the server function. Note that the JSON-RPC protocol is used (see [JSON Remote Procedure Calls \(RPC\)](#)), so there's no problem modifying the auto-generated proxy-code, or even completely replacing it with, for example, [jQuery Ajax](#) code.

Proxies that wrap the functions in a class, may be generated by adding a `class` argument. For example, `hello2.jss?class=Hello` will wrap the function in a class called `Hello`, such that the `hello` method may be called as follows:

```
var helloObject = new Hello();  
helloObject.hello();
```

Proxies that use promises may be generated by adding a `proxy` argument. For example, `hello2.jss?proxy` will allow methods to be called as follows:

```
hello().then(
  function(message) {
    document.write(message);
  },
  function(err) {
    document.write(err.message);
  }
);
```

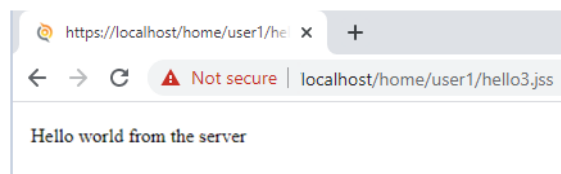
## 6. Combining simple requests and RPC

If a JSS file that contains RPC functions is also to handle basic HTTP requests, then the request-handling code should be placed inside a function called `processRequest()`. This function's job is to write the HTTP response that will be returned to the client. For example, the following JSS script, `hello3.jss`, returns the HTML to the client, which then invokes a function by RPC.

```
function hello(name) {
  return "Hello " + name + " from the server";
}

function processRequest() {
  var html = "<html><body>\n";
  html += "<script src='hello3.jss?proxy'></script>\n";
  html += "<script>\n";
  html += " var message = hello('world');\n";
  html += " document.write(message);\n";
  html += "</script>\n";
  html += "</body></html>";
  response.write(html);
}
```

When the HTML is rendered in the browser, the result will be:



Note that the Javascript code that's inside the quotes in the `processRequest()` function will be run on the client, and will call the code that's in the `hello()` function, which will be run on the server.

It is important to note that this server-side code is called in two different contexts. When `hello3.jss` is first called by the browser (without parameters), the default `processRequest()` is called to write the response. But the script referred to in the HTML generated by `processRequest()` is `hello3.jss?proxy`. The "proxy" parameter indicates that, rather than calling the default `processRequest()`, the caller wants the RPC proxies generated and returned to the browser, so that the `hello` function can be called via RPC.

## 7. Asynchronous RPC

Both synchronous and asynchronous remote procedure calls are supported. If an additional argument is provided, then it's assumed that the function is to be called asynchronously and that the additional argument is a callback function. For example, in the above example, the `hello()` function would be called asynchronously like this:

```
function onHello(message) {
  document.write(message);
}

hello("world", onHello);
```

If a second additional argument is provided, it will be assumed to be an error-callback function, which will be called if an error occurs. For example,

```
function onHelloSuccess(message) {
    document.write(message);
}

function onHelloError(error) {
    document.write("Error: " + error);
}

hello("world", onHelloSuccess, onHelloError);
```

If no error-callback is provided, then a default error-handling function will be used.

Although JSON-RPC is used by default, calls may optionally be made using form-based RPC (see [SS Remote Procedure Calls \(RPC\)](#)). This may be done by including a flag after the error-handler, e.g.

```
hello("world", onHelloSuccess, onHelloError, false);
```

This tells the proxy-function to use forms instead of JSON-RPC.

## 8. Uploading and downloading files

JSS makes it easy to upload and download files. If the desired response to a request is to return a file from the virtual file-system, then this may be achieved by setting the `response.downloadFile` property to the path of the file to be downloaded.

If a request contains one or more files to be uploaded (usually via the `input type=file` element in a form), then CompleteFTP will look for a Javascript function called `getUploadPath(fileName, contentType)` and call it to find out where (in the virtual file-system) the file is located which is to be uploaded. If no such function is found, then the file will be uploaded to the same directory as the script, if possible.

If an error occurs during upload, then a function with the name `errorHandler(errorMessage, errorType)` will be called, if available. Again, an alternative name may be provided by including an element with the name `errorHandler` and with the name of the desired error-handling function as its value.

## 9. Including other JSS files

Other JSS files may be included via the `include(scriptPath)` function. This function evaluates the contents of the included JSS file in the global scope, regardless of what scope it's called from, so that any functions and variables declared in the file will have global scope.

## 10. Controlling access to RPC functions

When developing a complex JSS application, it's important to manage the accessibility of the API. All functions in the global namespace will be available via RPC. If this is not desired, then the `publish` function may be used. For example, if in our hello example a second function was needed, but that function wasn't itself to be accessible via RPC, then the following code would be used:

```
(function() {
    function getSource() {
        return "server";
    }

    function hello(name) {
        return "Hello " + name + " from the " + getSource();
    }

    publish("hello", hello);
})();
```

Since all functions are declared inside a function-block, there are no functions in the global namespace until the call to the `publish` function pushes the `hello` function into it. Note that the name of the function and the function itself, must be provided separately.

Public APIs are vulnerable to denial of service attacks (DDoS), so by default CompleteFTP restricts the number of invocations to 100 per minute. This may be changed by providing an additional argument to the `publish` function, as follows:

```
publish("hello", hello, { maxCallsPerMinute: 10 });
```

Since one JSS file may be included in another JSS file using the `include` function, it's sometimes desirable to allow another JSS file to invoke a function without also exposing it publicly. This may be done by providing a `serverOnly` argument, as follows:

```
publish("getSource", getSource, { serverOnly: true });
```

Functions that are published as `serverOnly` will not be included in proxies and may only be invoked by other JSS scripts.

## 11. Extending JSS using .NET

JSS can call .NET methods that are defined in JSS extensions - see [here](#).

## 12. Exploring Further

Details on the API available in JSS are in the [JSS API Guide](#). Information on the RPC protocol used in JSS, may be found in [JSS Remote Procedure Calls \(RPC\)](#).

# Using Templates

JSS supports templates, which facilitates the separation of Javascript and HTML code into separate files.

JSS templates work by expanding tags in a template using values provided by a hash or object. There are no if statements or any other logic - just tags, which are shown by double curly braces, e.g. `{{name}}`.

This tag expansion happens on the server-side - in the server-side JSS file, Javascript objects are supplied to a render method that takes the HTML file containing the tags and produces the expanded HTML file. See [this example](#).

## Variables

Variables are the most basic tag. A `{{name}}` tag in a basic template will try to find the name key in the current context. If there is no name key, an empty string is substituted.

All variables are HTML escaped by default. If you want to return unescaped HTML, use `{{name}}`. You can also use `&` to unescape a variable, e.g. `{{&name}}`.

## Sections

Sections render blocks of text one or more times, depending on the value of the key in the current context. They begin with a pound and end with a slash, e.g. `{{#item}}` and `{{/item}}`.

If the key has a value of false or is an empty list, the HTML between the pound and slash is not returned. If the key exists and is non-false, the HTML between the pound and the slash is rendered one or more times - once for every item in the list.

JSS templates are based on [Mustache templates](#). More detailed examples and extended capabilities such as lambdas are explained [here](#).

# Web-App Examples

Live JSS examples are hosted on our public server and can be found [here](#).

If you are reading this documentation from a machine that is hosting a running instance of CompleteFTP, you may be able to view the JSS examples on your own server, [here](#).

# How to write a JSS event

A JSS event is a [process trigger](#) event written in [JSS](#) (Javascript Server-Side).

## Creating a JSS event

Creation of process triggers events is described in [How to set up process triggers](#). To create a JSS event, select the type "JSS script".

## Tutorials

The tutorials below demonstrate common examples of JSS events:

1. [Tutorial 1 - Upload file to server](#): demonstrates an uploading event, which is triggered when a user uploads a file to server.
2. [Tutorial 2 - Error handling](#): demonstrates error handling using a try-catch statement.
3. [Tutorial 3 - Multiple Uploading](#): demonstrates how to create a JSS event using multiple uploads.
4. [Tutorial 4 - Scheduled events](#): demonstrates how to work with scheduled events.
5. [Tutorial 5 - Record the type of event](#): demonstrates how to record events in a file.
6. [Tutorial 6 - Move a file from this Path to another Path](#): demonstrates how to move a file from this Path to another Path.

**Note:** if you run these tutorials on localhost, note that the script's uploading of the file will trigger the same event again.

Apart from the normal JSS API, you also have access to the new Ftp class. You can find the reference for Ftp class [here](#).

For easier debugging, use the pop-out, real-time log via the View/Show logging window, Monitoring/Open in window, and in Events/Show log.

Now please proceed to [Tutorial 1- Upload file to server](#).

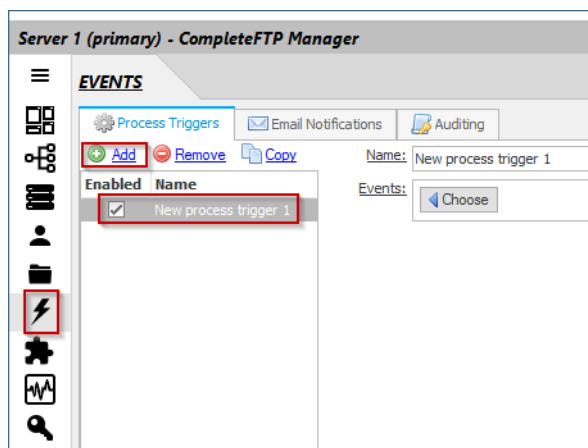
# Tutorial 1 - Upload file to server

Before reading this tutorial, please make sure you have CompleteFTP installed, and two users ('user1' and 'user2') created (see [here](#)).

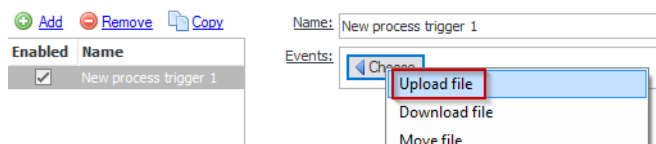
This tutorial will implement a JSS process trigger that automatically transfers a file to a remote server, that has just been uploaded to your server.

## Steps

1. Open CompleteFTP Manager and select the Events tab. In Process Triggers panel click on the 'Add' button to add a new event.



2. Choose the Upload file event.

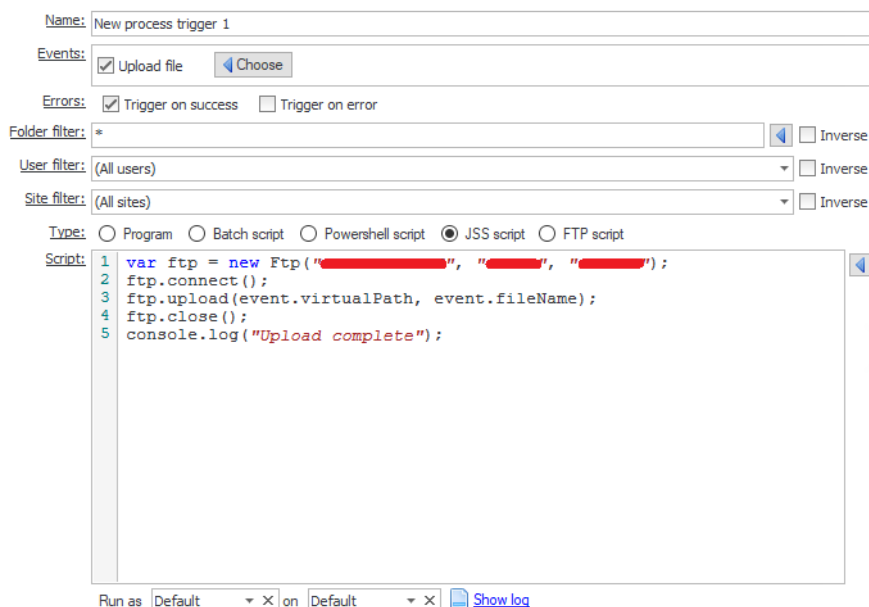


3. Check the "Trigger on success" box, then choose the JSS script and enter the script shown below:

```
var ftp = new Ftp("myftpserver", "myusername", "mypassword");
ftp.connect();
ftp.upload(event.virtualPath, event.fileName);
ftp.close();
console.log("Upload complete");
```

This script uses the `Ftp` class (a part of the JSS API). As you can see, three arguments are passed to the constructor: the IP address of the remote host, the user-name and the password. The `connect()` method is then called to connect to the server. Calling this method is optional, because the client will automatically connect when any other method is called, if it hasn't already connected. Calling it explicitly may be useful for error-handling. The `upload` method is then called to upload a file from the local virtual file-system, to the remote server. After uploading, the `close()` method is called to close the connection (again, this is optional as it happens automatically when the script completes).

The process trigger should now look like this:



Note that, if another FTP server is not available at this point, then for the purposes of this tutorial, the same server (i.e. 'localhost') can be used in the script. If you choose to do this,



however, then you must make sure that this event doesn't trigger again, when the script uploads the file. You can do this by using the user filter to exclude the user, 'user2'. To do this, click the 'User filter' control, select 'user2' from the list and then check the 'Inverse' checkbox to the right.

Name: New process trigger 1

Events: ☒ Upload file

Errors: ☒ Trigger on success ☐ Trigger on error

Folder filter: \*  ☐ Inverse

User filter: user2  ☒ Inverse

Site filter: (All sites)  ☐ Inverse

Type: ☐ Program ☐ Batch script ☐ Powershell script ☒ JSS script ☐ FTP script

Script: 

```
1 var ftp = new Ftp(" ", " ", " ");
2 ftp.connect();
3 ftp.upload(event.virtualPath, event.fileName);
4 ftp.close();
5 console.log("Upload complete");
```

Run as: Default on Default

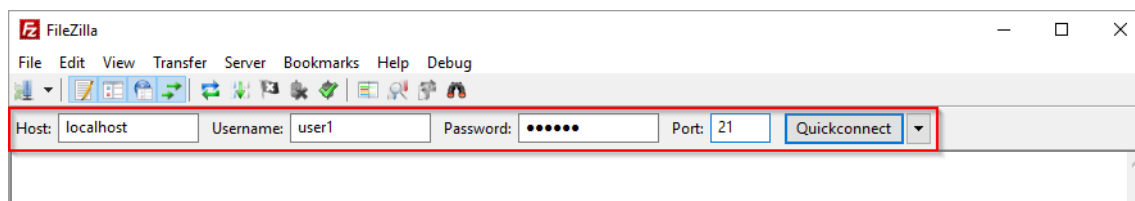
If you don't do this, then the upload operation in the script will trigger the upload event, which will execute the script, ad infinitum.

4. When you've done scripting, click on 'Apply Changes' to save it and you can then start to use this event.

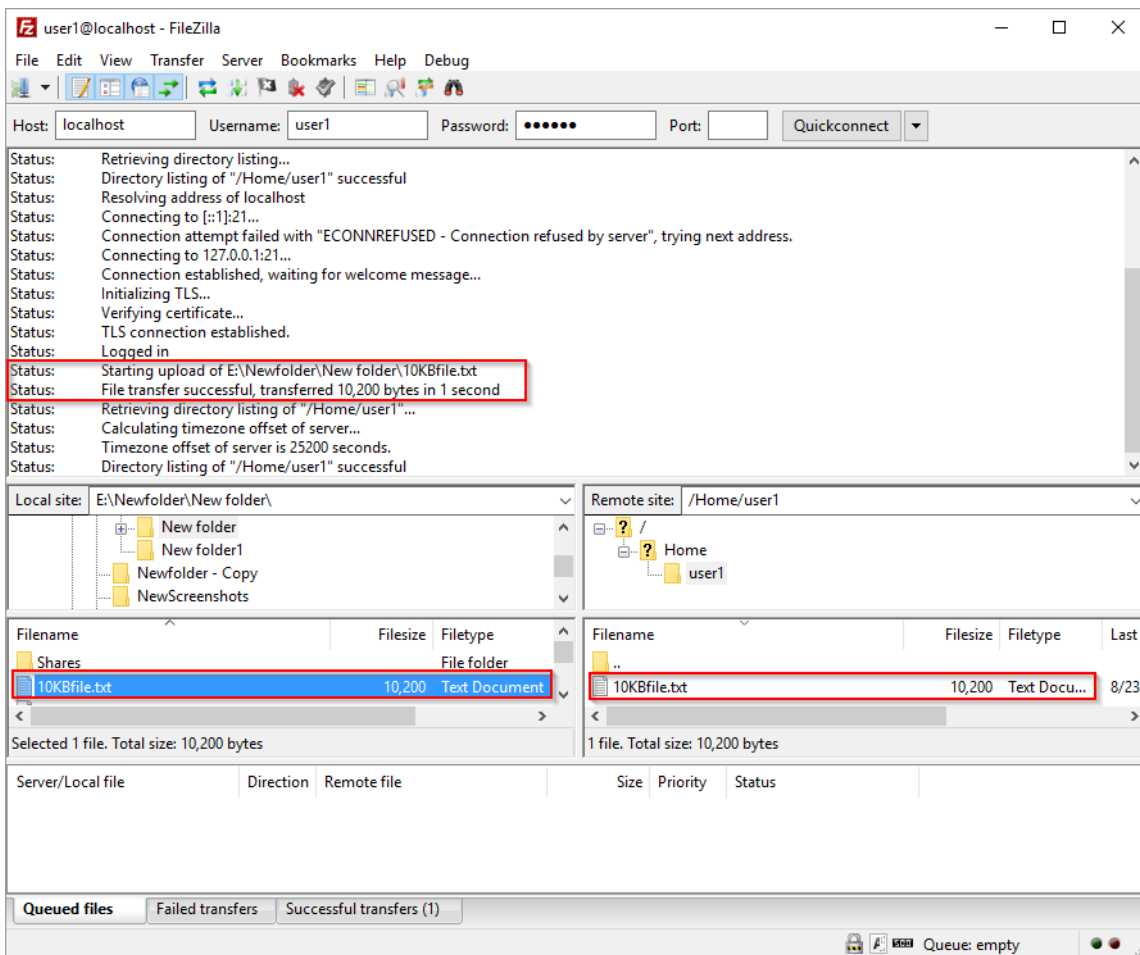
### Testing your process trigger

Now your new JSS process trigger is ready to use. Let's check it using Filezilla. If you're new to Filezilla, please check the how to use Filezilla [here](#).

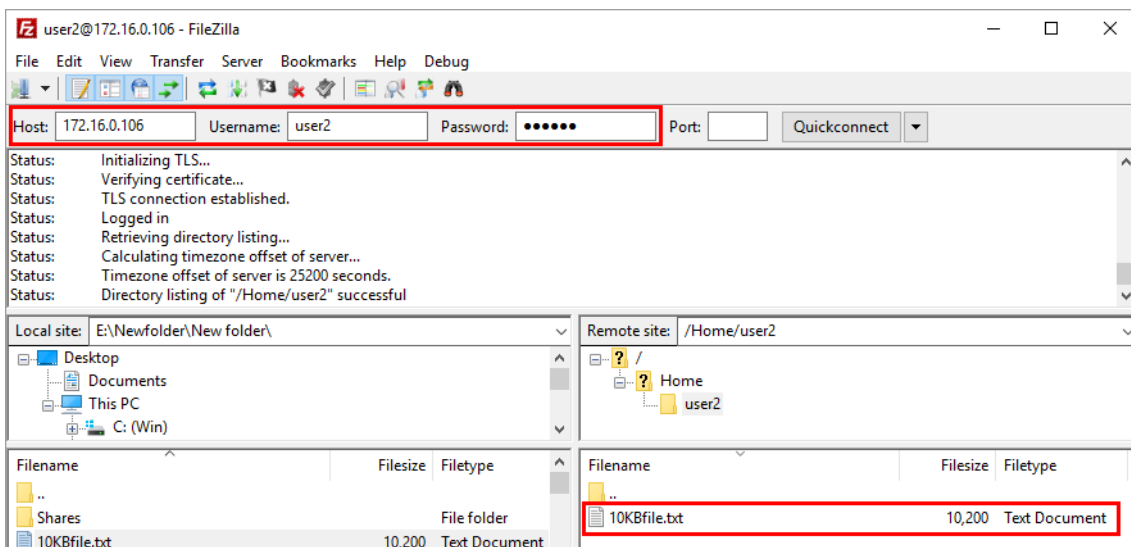
1. Open Filezilla and connect to the server as 'user1'.



2. Upload a file to the server.



3. Now connect to the remote server as 'user2' and you will see the same file has also been uploaded to this account's home folder.



Now please proceed to [Tutorial 2 - Error handling](#).

## Tutorial 2 - Error handling

You have created the first JSS event (see [Tutorial 1 - Upload file to server](#)). Now it's time to add some error handling to your script.

The tutorial below will show you a way to add error handling, using a try-catch statement.

### Steps

1. Open the the event you created in [Tutorial 1](#).

Name: New process trigger 1

Events: ☒ Upload file [Choose](#)

Errors: ☒ Trigger on success ☐ Trigger on error

Folder filter: \* [Inverse](#)

User filter: (All users) [Inverse](#)

Site filter: (All sites) [Inverse](#)

Type: ☐ Program ☐ Batch script ☐ Powershell script ☒ JSS script ☐ FTP script

Script: 

```
1 var ftp = new Ftp("...", "...", "...");
2 ftp.connect();
3 ftp.upload(event.virtualPath, event.fileName);
4 ftp.close();
5 console.log("Upload complete");
```

Run as: Default on Default [Show log](#)

2. Edit it, by putting all the current code into try block and add the catch block. To check the error handling, we change the username to an invalid one (in the example below, the 'user3' doesn't exist in the server).

Name: New process trigger 1

Events: ☒ Upload file [Choose](#)

Errors: ☒ Trigger on success ☐ Trigger on error

Folder filter: \* [Inverse](#)

User filter: (All users) [Inverse](#)

Site filter: (All sites) [Inverse](#)

Type: ☐ Program ☐ Batch script ☐ Powershell script ☒ JSS script ☐ FTP script

Script: 

```
1 try {
2   var ftp = new Ftp("...", "...", "...");
3   ftp.connect();
4   ftp.upload(event.virtualPath, event.fileName);
5   ftp.close();
6   console.log("Upload complete");
7 } catch (error) {
8   console.error("Upload of " + event.fileName + " failed: " + error);
9 }
```

Run as: Default on Default [Show log](#)

### Sample script

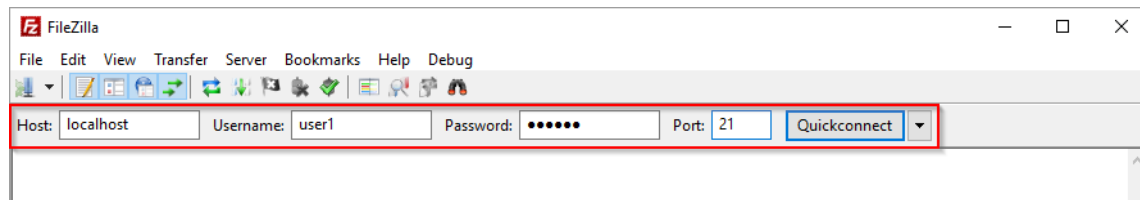
Here is the sample script which is used in this tutorial.

```
try {
  var ftp = new Ftp("myftpserver", "myusername", "mypassword");
  ftp.connect();
  ftp.upload(event.virtualPath, event.fileName);
  ftp.close();
  console.log("Upload to myftpserver complete");
} catch (error) {
  console.error("Upload of " + event.fileName + " failed: " + error);
}
```

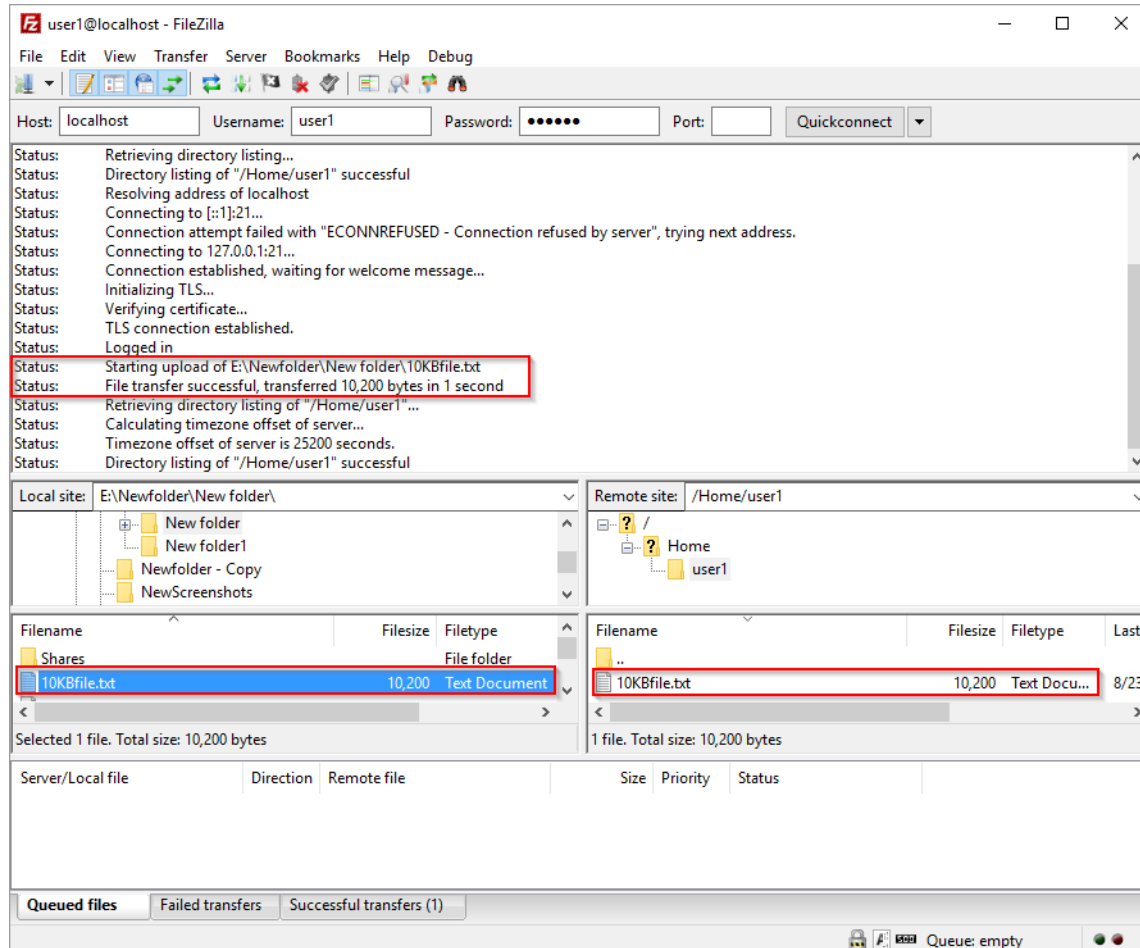
### Checking result

Now your error handling is ready to use, let's check it.

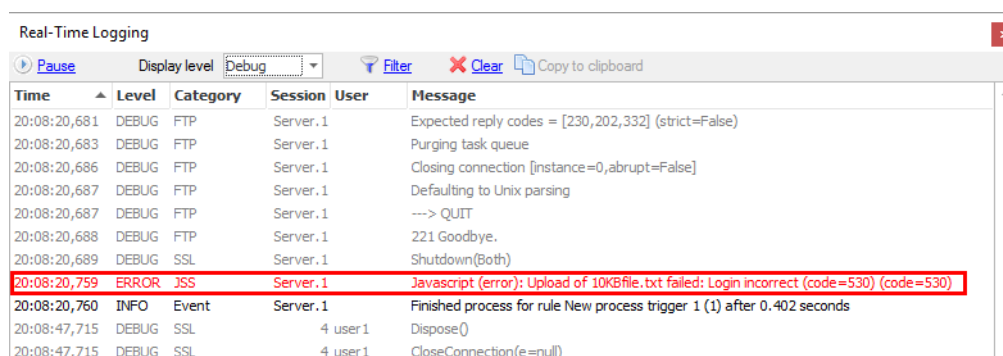
1. Open Filezilla and connect to the server as 'user1'.



2. Upload a file to the server.



3. Now we can check the CompleteFTP logging and get the error log, in the format we set in catch block.



Now please proceed to [Tutorial 3 - Multiple Uploading](#).

# Tutorial 3 - Multiple Uploading

Now it's time to create a more complex JSS event. The tutorial below will show you how to create a JSS event which can work with multiple uploading.

## Steps

1. Open the the event you created in [Tutorial 2 - Error handling](#).

Name: New process trigger 1

Events: ☒ Upload file

Errors: ☒ Trigger on success ☐ Trigger on error

Folder filter: \*  ☐ Inverse

User filter: (All users)  ☐ Inverse

Site filter: (All sites)  ☐ Inverse

Type: ☐ Program ☐ Batch script ☐ Powershell script ☒ JSS script ☐ FTP script

Script:

```
1 try {
2     var ftp = new Ftp(" ", " ", " ");
3     ftp.connect();
4     ftp.upload(event.virtualPath, event.fileName);
5     ftp.close();
6     console.log("Upload complete");
7 } catch (error) {
8     console.error("Upload of " + event.fileName + " failed" + error);
9 }
```

Run as Default on Default

2. Now we can add one more servers to upload the file to (if you have more than one server please feel free to add them in). Then we add a 'for loop', to loop through these servers.

Name: New process trigger 1

Events: ☒ Upload file

Errors: ☒ Trigger on success ☐ Trigger on error

Folder filter: \*  ☐ Inverse

User filter: (All users)  ☐ Inverse

Site filter: (All sites)  ☐ Inverse

Type: ☐ Program ☐ Batch script ☐ Powershell script ☒ JSS script ☐ FTP script

Script:

```
1 var servers = [
2   { host: " ", userName: " ", password: " " },
3   { host: " ", userName: " ", password: " " },
4 ];
5 for (var i in servers)
6   try {
7     var server = servers[i];
8     var ftp = new Ftp(server.host, server.userName, server.password)
9     ftp.connect();
10    ftp.upload(event.virtualPath, event.fileName);
11    ftp.close();
12    console.log("Upload to " + server.host + " complete");
13 } catch (error) {
14     console.error("Upload of " + event.fileName + " failed: " + error);
15 }
```

Run as Default on Default

## Sample script

Here is the sample script which is used in this tutorial.

```

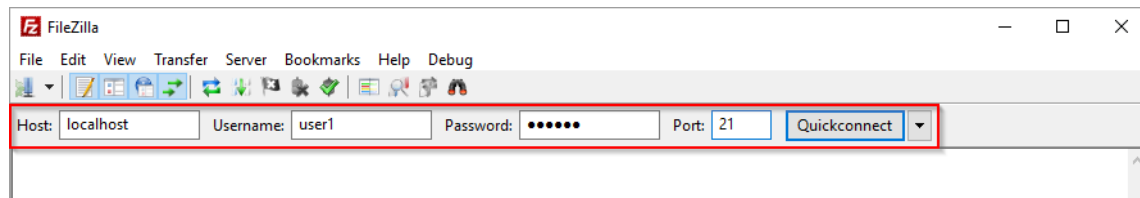
var servers = [
  { host: "myftpserver1", userName: "myusername1", password: "mypassword1" },
  { host: "myftpserver2", userName: "myusername2", password: "mypassword2" },
];
for (var i in servers)
  try {
    var server = servers[i];
    var ftp = new Ftp(server.host, server.userName, server.password)
    ftp.connect();
    ftp.upload(event.virtualPath, event.fileName);
    ftp.close();
    console.log("Upload to " + server.host + " complete");
  } catch (error) {
    console.error("Upload of " + event.fileName + " failed: " + error);
  }
}

```

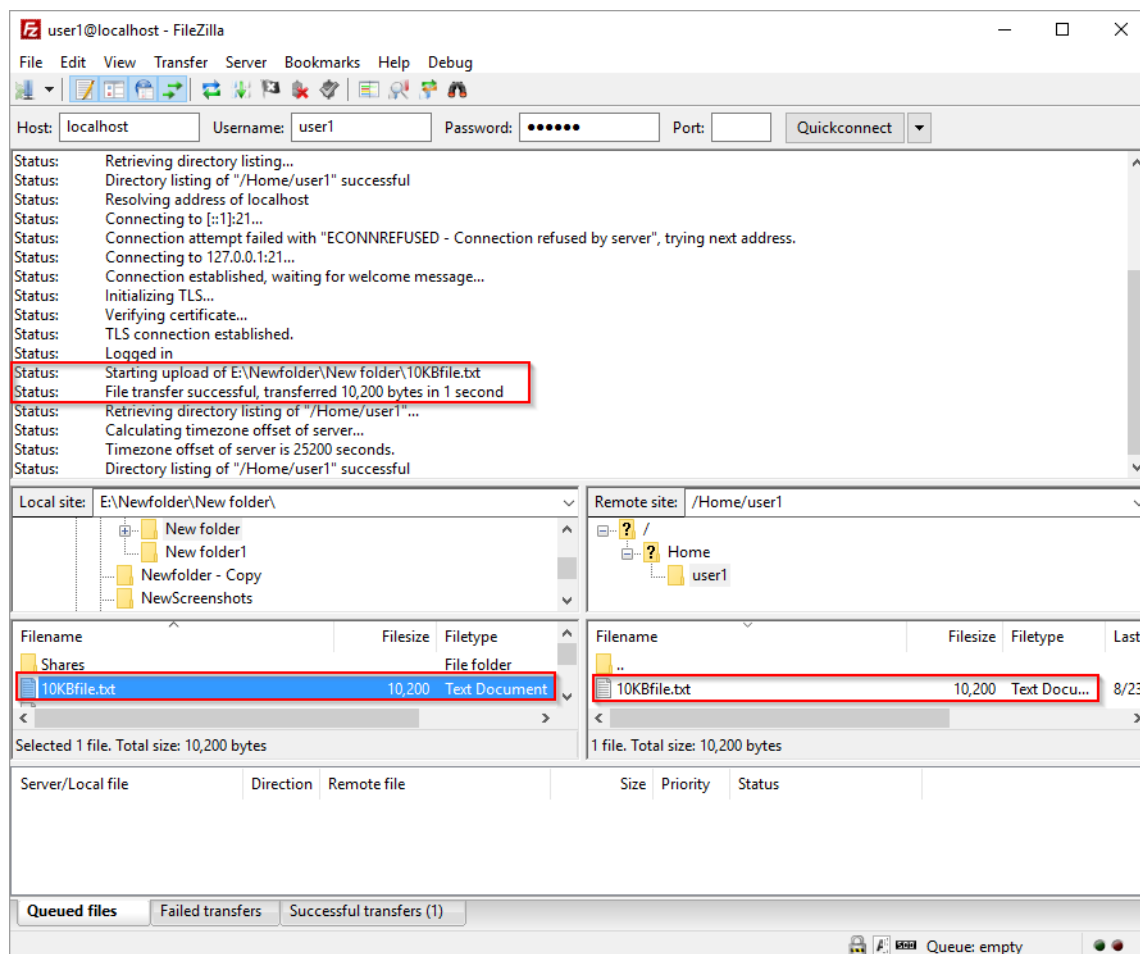
## Checking result

Now your new JSS event is ready to use, let's check it using Filezilla.

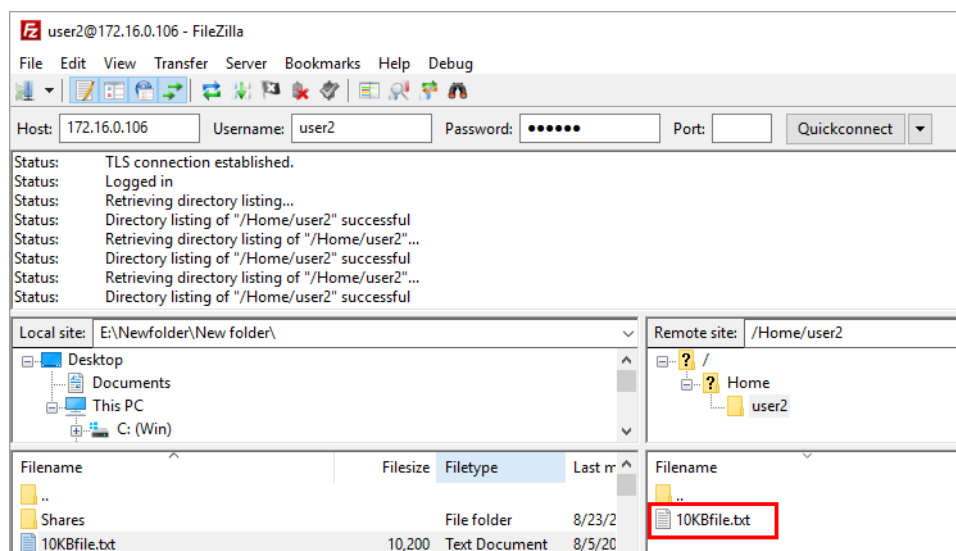
1. Open Filezilla and connect to the server as 'user1'.



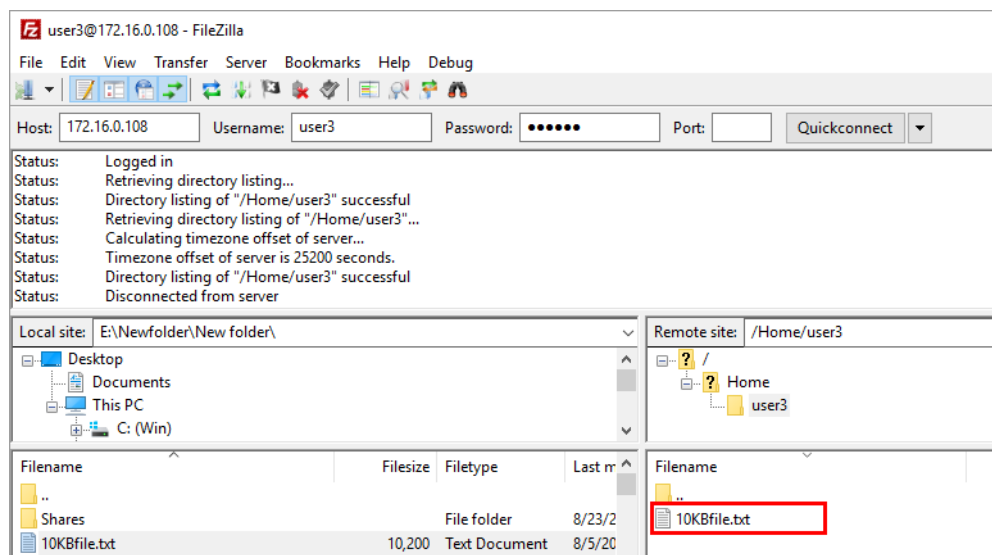
2. Upload a file to the server.



3. Now connect to the first remote server as 'user2' and you will see the same file has also been uploaded to this account's home folder.



4. Now connect to the second remote server as 'user3' and you will see the same file has also been uploaded to this account's home folder.



Now please proceed to [Tutorial4 - Upload event-handler](#).

# Tutorial 4 - Upload event-handler

Automatically uploading a file to a server does not always result in success, there are various reasons for why the upload may fail. In CompleteFTP, we can add one more event, which triggers according to a pre-defined schedule. This will retry the file upload to the server, until it is successfully uploaded. This tutorial will show you how to do this:

## Steps

1. Create a user and folder with the credentials below:

- User-type: Non-Windows user
- User-name: queueOwner
- Type of home folder: Windows folder
- Windows path of home folder: C:\FTP\Queue
- Virtual path of home folder: /Queue.

Then set 'queueOwner' as the owner of the /Queue folder.

Add User

Add a non-Windows user

Step 4: Verify Details

Summary:

Username:  
queueOwner

Full name:  
(none)

Description:  
(none)

E-mail:  
(none)

Home folder:  
/Queue (C:\FTP\Queue)

Previous

Apply

Cancel

2. Clear all group permissions for this folder to prevent other users from accessing it.

Name	Type	Owner	Group
> Home	Windows Folder	admin	admins
> Public	Windows Folder	anonymous	users
Queue	Windows Folder	admin	users

Name

Queue

Type

Windows Folder

Path

C:\FTP\Queue

Hidden

☐

Access (non-Windows users)

Owner

admin

Group

users

Permissions

Click to view

Access (Windows users)

Permissions

Members of group can...

download files

upload files

rename files

delete files

list folder contents

change into folder

create subfolders

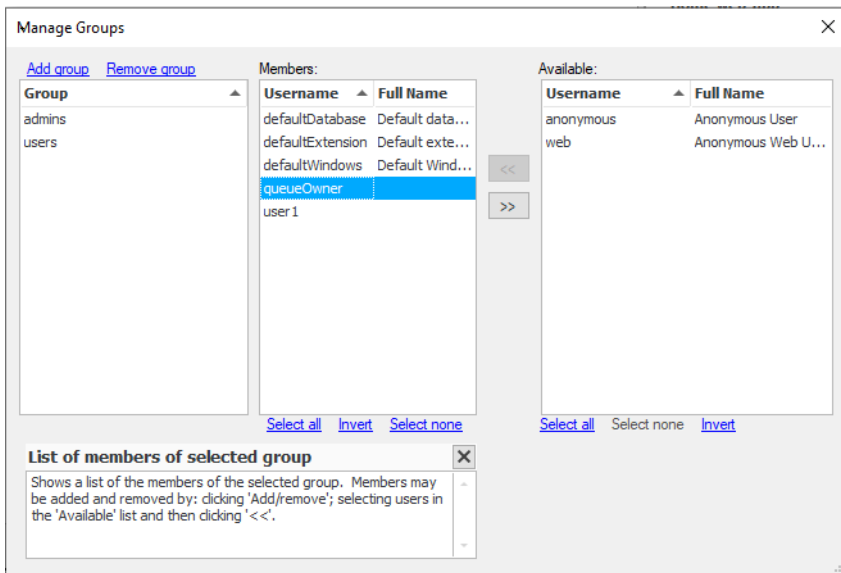
rename subfolders

delete subfolders

Ignore filters

3. Add the 'queueOwner' user to the admins group. This will give 'queueOwner' permission to access any folder in the virtual file-system, which it will need to have, in order to be able to distribute the files that have been uploaded.





4. Create a file called `variables.json` which will be used to store app variables in `C:\FTP\Queue`. Here is the content, in which the `"ftpserver.com"`, `"myusername"`, `"mypassword"`, and `"/Home/myusername/"` are your credentials:

```
{
  "fileCounter": 0,
  "remoteServer": {
    "hostName": "ftpserver.com",
    "userName": "myusername",
    "password": "mypassword",
    "directory": "/Home/myusername/"
  }
}
```

5. Create a process trigger, which adds newly uploaded files to the queue. Notice that the script is configured to run as the `'queueOwner'` user.

```
var db = new JSONDB("/Queue");

// get a number for the uploaded file
var fileNumber;
db.update("/variables", function(variables) {
  variables.fileCounter = variables.fileCounter + 1;
  fileNumber = variables.fileCounter;
});

// write an entry in the queue for the file
db.write("/items/" + fileNumber, {
  path: event.virtualPath,
  uploader: event.loginUserName,
  uploaded: new Date()
});

console.log("File " + event.virtualPath
  + " added to queue as item number " + fileNumber);
```

6. Add another process trigger, which periodically takes items in the queue and uploads the corresponding files. Notice again that the script runs as the `'queueOwner'` user.

```

var db = new JsonDB("/Queue");

// get the remote server details from database and increase the counter
var remoteServer = db.read("/variables.remoteServer");

// prepare the FTP client
var ftp      = new Ftp();
ftp.hostName  = remoteServer.hostName;
ftp.protocol  = remoteServer.protocol;
ftp.userName  = remoteServer.userName;
ftp.password  = remoteServer.password;
ftp.serverDirectory = remoteServer.directory;

// get queued items from database
var items = db.read("/items");
for (var i in items) {
    var item = items[i];
    try {
        ftp.upload(item.path);
        console.log("Queued file uploaded successfully " + item.path);

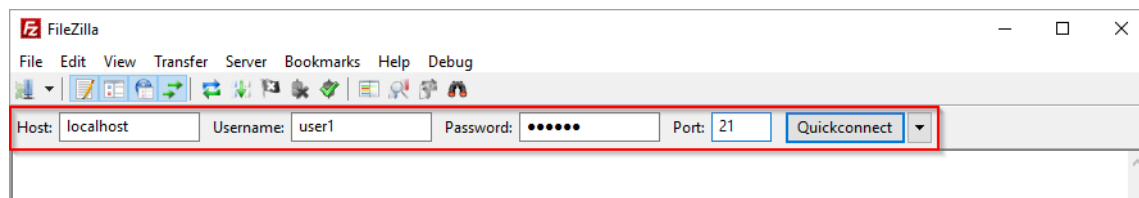
        // remove the item from the queue
        db.remove("/items/" + i);
    } catch (error) {
        console.log("Failed to upload " + item.path + ": " + error);
    }
}

```

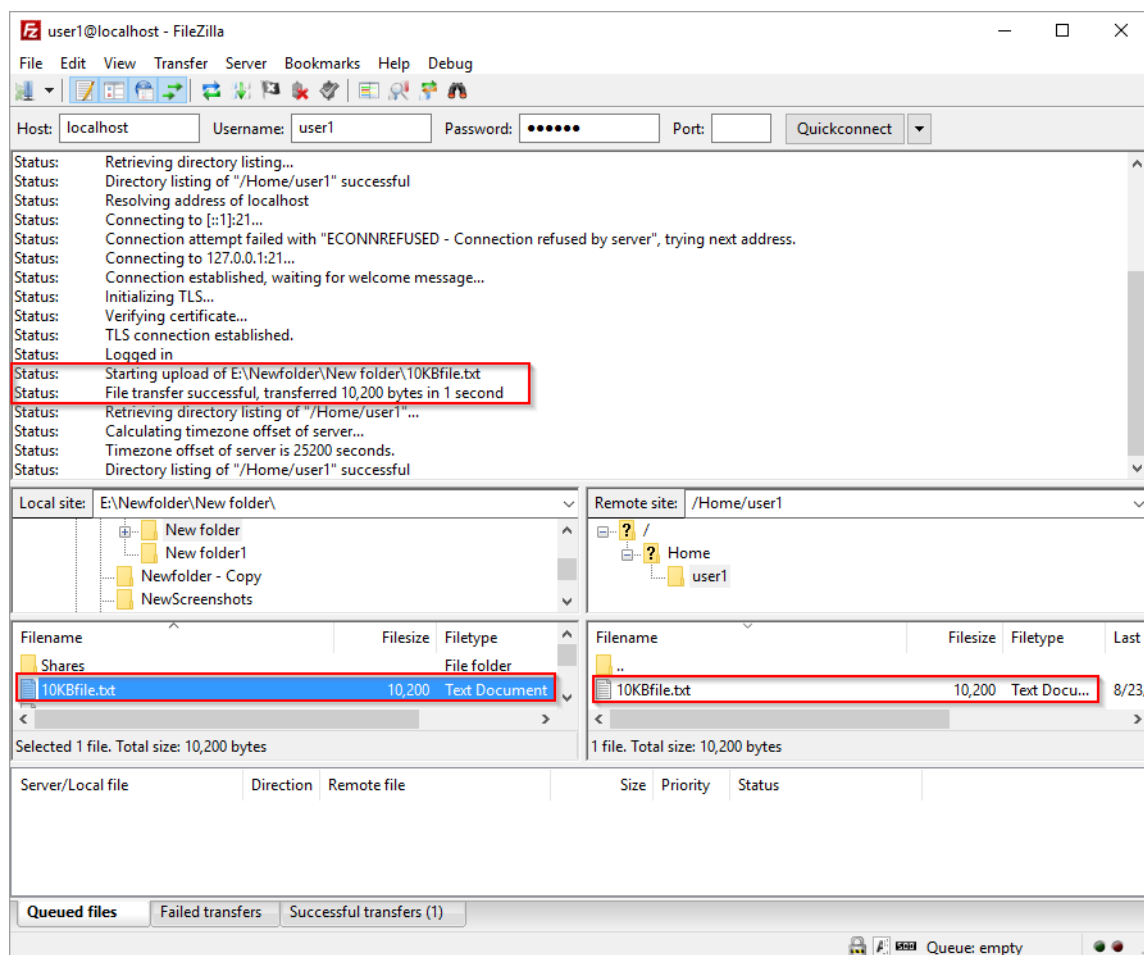
### Checking result

Now your new JSS event is ready to use, and after an upload we can see the file in /Queue/items folder, as well as being able to see the upload retry, after the time interval which we set.

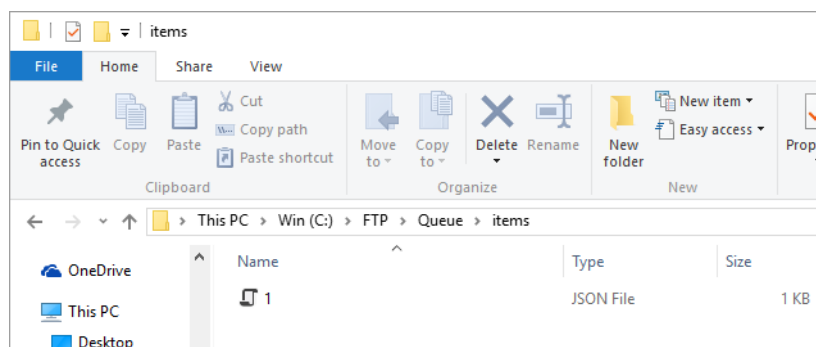
1. Open Filezilla and connect to the server as 'user1'.



2. Upload a file to the server.



3. Now check the Queue folder and you will see the sub folder "items" has been created and it contains a file.



4. Check the log to see the scheduled event triggers. Below is the log after 3 files have uploaded:

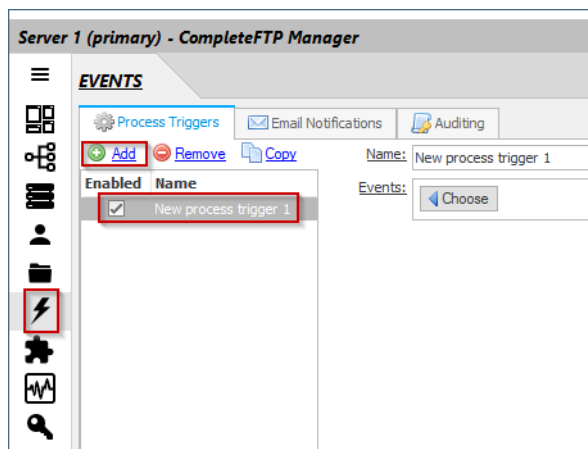
20:17:13,599	INFO	FTP	12 User 1	Closed file /Home/User 1/10kbkey - Copy after writing 469 bytes
20:17:13,599	INFO	Event	Server.1	Starting process for rule Upload event-handler (1)
20:17:13,604	INFO	Core	Server.1	queueOwner logged into Default Site with protocol
20:17:13,608	INFO	JSS	Server.1	Javascript (log): File /Home/User 1/10kbkey - Copy added to queue as item number 3
20:17:13,608	INFO	Event	Server.1	Finished process for rule Upload event-handler (1) after 0.01 seconds
20:17:14,999	INFO	Event	Server.1	Starting process for rule Scheduled event-handler (2)
20:17:15,003	INFO	Core	Server.1	defaultExtension logged into Default Site with protocol
20:17:15,006	INFO	FTP	Server.1	OS: 6.2.9200.0, CLR: 4.0.30319.42000, DLL: 9.1.0.0
20:17:15,006	INFO	FTP	Server.1	Built: @date_time@
20:17:15,007	INFO	FTP	Server.1	SetSSLProtocol=TLS, TLS11, TLS12
20:17:15,009	INFO	FTP	Server.1	Connecting to 172.16.0.119:21 with timeout 120000 ms
20:17:15,010	INFO	FTP	Server.1	Command encoding=System.Text.SBCSCodePageEncoding
20:17:15,028	INFO	FTP	Server.1	Auto FEAT disabled
20:17:15,033	INFO	FTP	Server.1	Connecting to 172.16.0.119:55664 with timeout 120000 ms
20:17:15,037	INFO	JSS	Server.1	Javascript (log): Queued file uploaded successfully /Home/User 1/10kbkey - Copy
20:17:15,038	INFO	FTP	Server.1	Killed control socket
20:17:15,038	INFO	Event	Server.1	Finished process for rule Scheduled event-handler (2) after 0.039 seconds

## Tutorial 5 - Record events in a file

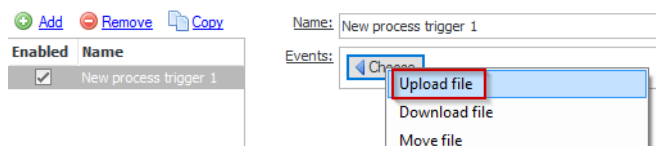
This tutorial will implement a JSS process trigger that automatically writes the type of event which happens with files and folders into a .txt file (file in local).

### Steps

1. Open CompleteFTP Manager and select the Events tab. In the Process Triggers panel, click on the 'Add' button to add a new event.



2. Choose multiple events or all events.

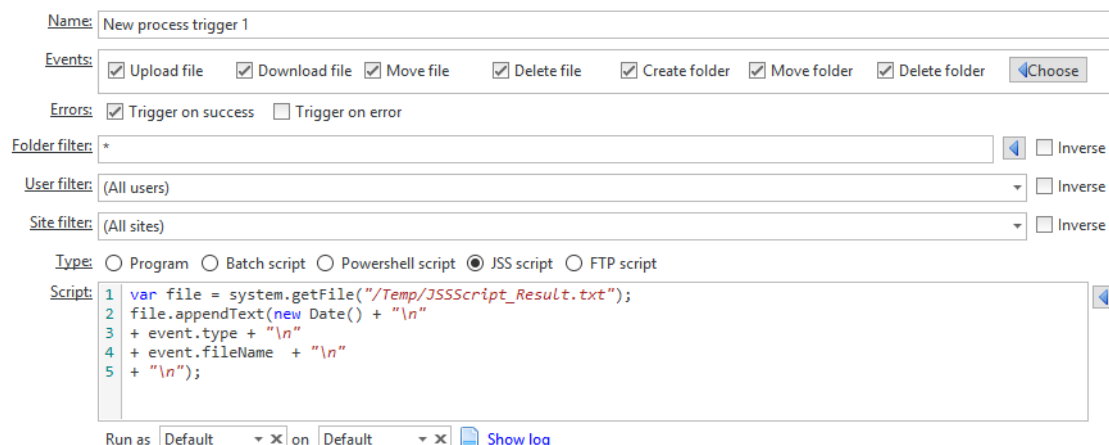


3. Check the "Trigger on success" box, then choose the JSS script and enter the script shown below:

```
var file = system.getFile("/Temp/JSSScript_Result.txt");
file.appendText(new Date() + "\n"
+ event.type + "\n"
+ event.fileName + "\n"
+ "\n");
```

This script will open the file or create the file (if it doesn't exist) then it will append the time stamp, the type of event, and the file name.

The process trigger should now look like this:



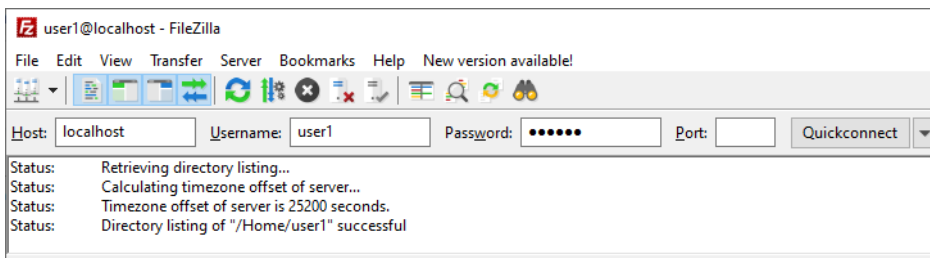
4. When you've done scripting, click on 'Apply Changes' to save and start using this event.

### Testing your process trigger

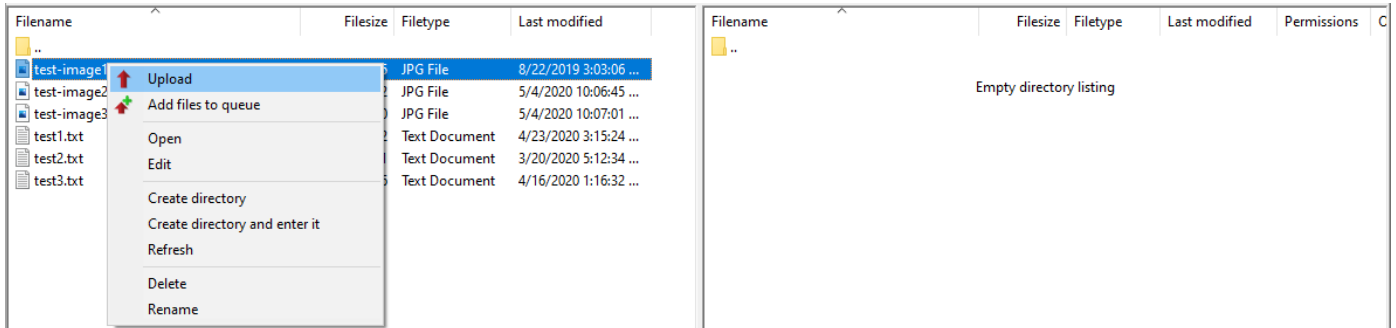
Now your new JSS process trigger is ready to use. Let's check it using Filezilla. If you're new to Filezilla, please check the 'how to use Filezilla' guide [here](#).

Before testing this process trigger, you have to create a Windows folder which maps to C:\Temp.

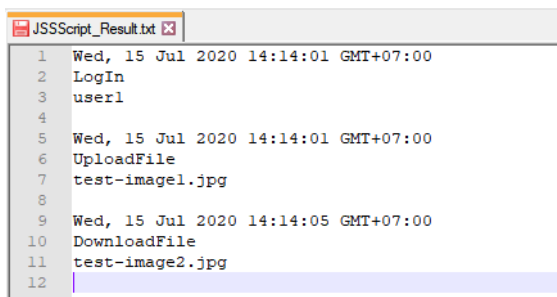
1. Open Filezilla and connect to the server as 'user1'.



- Upload a file ('test-image1.jpg') to the server, rename it (to 'test-image2.jpg') then download it from server.



- Now check in C:\Temp to see if the JSSScript\_Result.txt file has been created and that it has written the content as shown below:

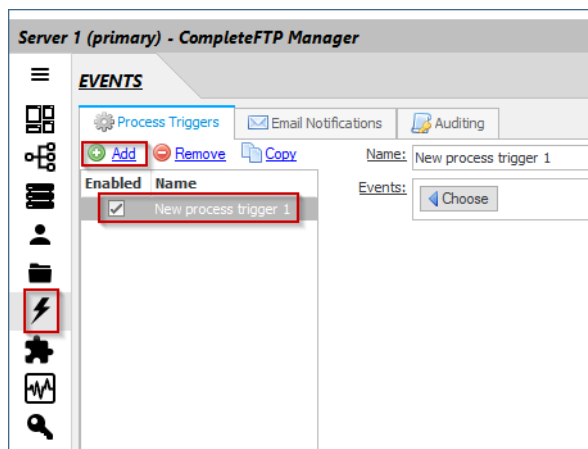


# Tutorial 6 - Move a file from this Path to another Path

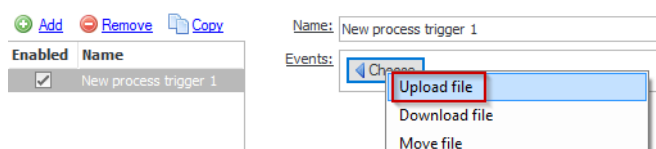
This tutorial will implement a JSS process trigger which automatically moves a file from one Path to another Path.

## Steps

1. Open the CompleteFTP Manager and select the Events tab. In the Process Triggers panel click on the Add button to add a new event.



2. Choose the Upload file event.

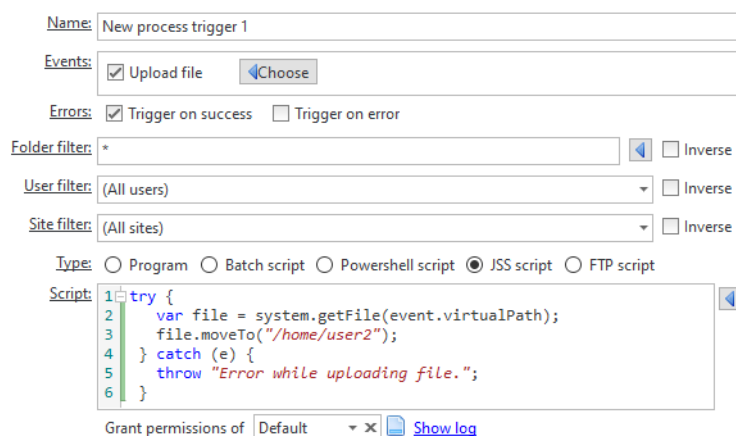


3. Check the "Trigger on success" box, then choose the JSS script and enter the script shown below:

```
try {  
  var file = system.getFile(event.virtualPath);  
  file.moveTo("/home/user2");  
} catch (e) {  
  throw "Error while uploading file.";  
}
```

This script will move myfile.txt from its current path (in this case is /home/user1) to /home/user2.

The process trigger should now look like this:

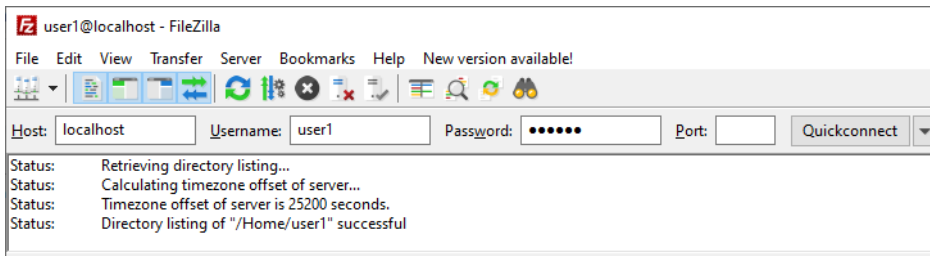


4. When you've done with scripting, click on 'Apply Changes' to save it and you can then start using this event.

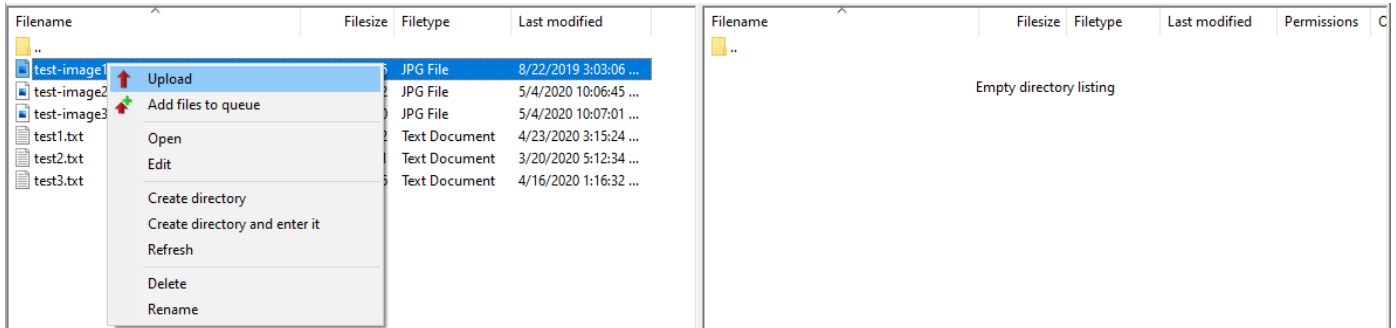
## Testing your process trigger

Now your new JSS process trigger is ready to use. Let's check it using Filezilla. If you're new to Filezilla, please check the 'how to use Filezilla' Guide [here](#).

1. Open Filezilla and connect to the server as 'user1'.



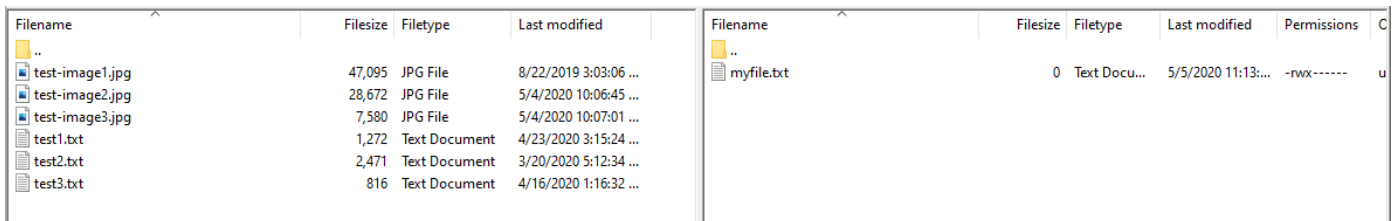
- Upload a file to server, it will be upload to /home/user1, then our process trigger is triggered and move the file to /home/user2



- Now log into the server, with the 'user2' which was used in the second path.



- The file myfile.txt should be moved there:



# How to write a JSS extension

A JSS Extension is an extension written in **JSS** (Javascript Server-Side).

- JSS uses standard Javascript syntax and supports the **JSS API**, which gives access to resources such as the CompleteFTP virtual file-system.
- JSS supports the use of standard .NET classes, which can be instantiated in the same way as in C# (e.g. `new FullDotNetClassName()` ;).
- JSS can call into custom DLLs using **JSS-to-.NET Bridge Extensions**.
- JSS code may be stored in the CompleteFTP configuration or in external files. Changes to JSS files are detected in real time.

Four types of JSS extensions are supported:

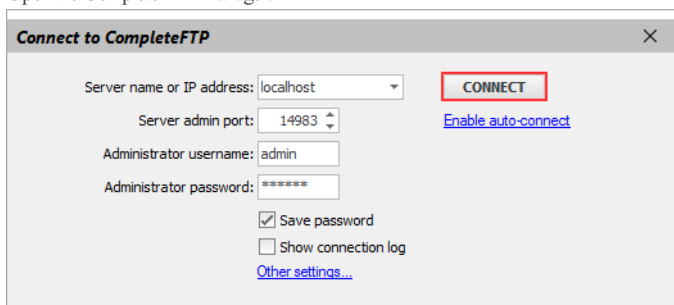
- **Authentication extensions**
- **File-system extensions**
- **Custom-command extensions**
- **IP filter extensions**

Note that this feature is only available in Enterprise MFT.

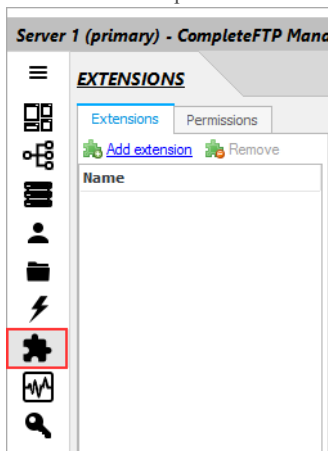
## Creating a JSS extension

To create a CompleteFTP JSS extension:

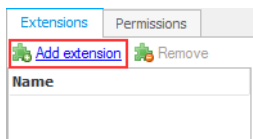
1. Open the CompleteFTP Manager.



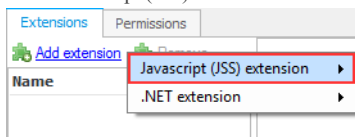
2. Select the Extensions panel then select the Extensions tab.



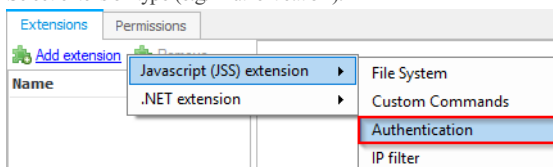
3. Click 'Add extension'.



4. Select 'Javascript (JSS) extension'.

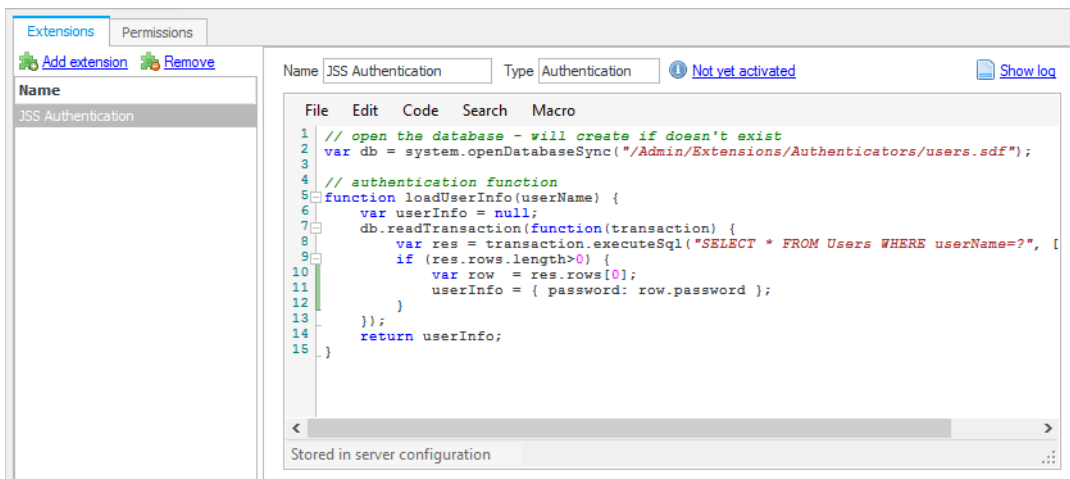


5. Select extension type (e.g. 'Authentication').

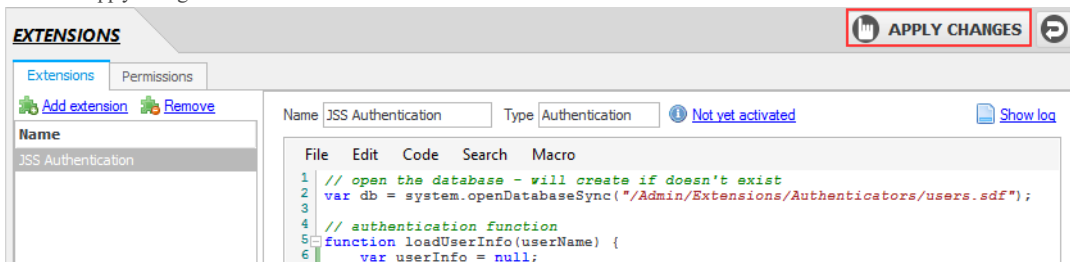


6. Enter the code for your extension. You can write it in the CompleteFTP syntax-highlighting editor or any of your favourite Javascript editors, then add this JSS file to the CompleteFTP editor by 'File/Open from/File on server...!'.





- Some extension types require you to perform an additional registration step, such as adding a new folder-type, or granting permissions to site-commands.
- Click the 'Apply Changes' button.



- Test your JSS Extension by using an FTP client.

Note: Full details of JSS API are provided in the [JSS API Reference](#).

## Registration of JSS extensions

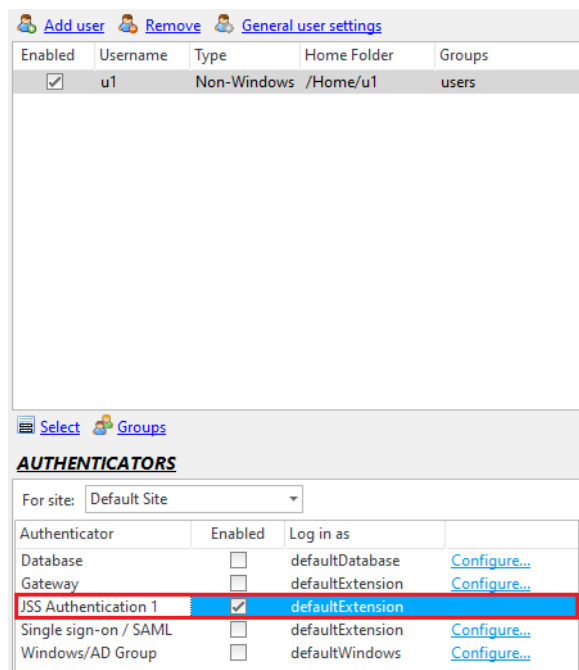
Some JSS extensions must be registered in order for them to take effect. This is done automatically for authentication and file-system extensions when they are created, but it must be done manually for custom-command extensions. Each type of JSS extension has its own method of registration:

### IP filter extensions

IP filter extensions will be used as soon as they're added.

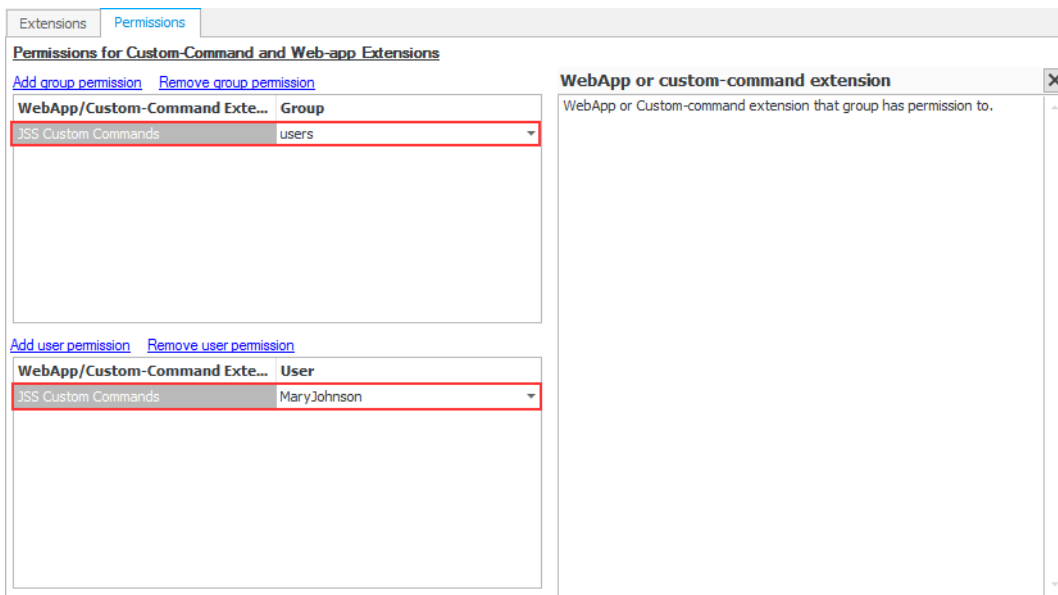
### Authentication extensions

Custom authenticators will only be used if they're enabled in the 'Other authentication methods' list, in General User Settings, in the Users panel.



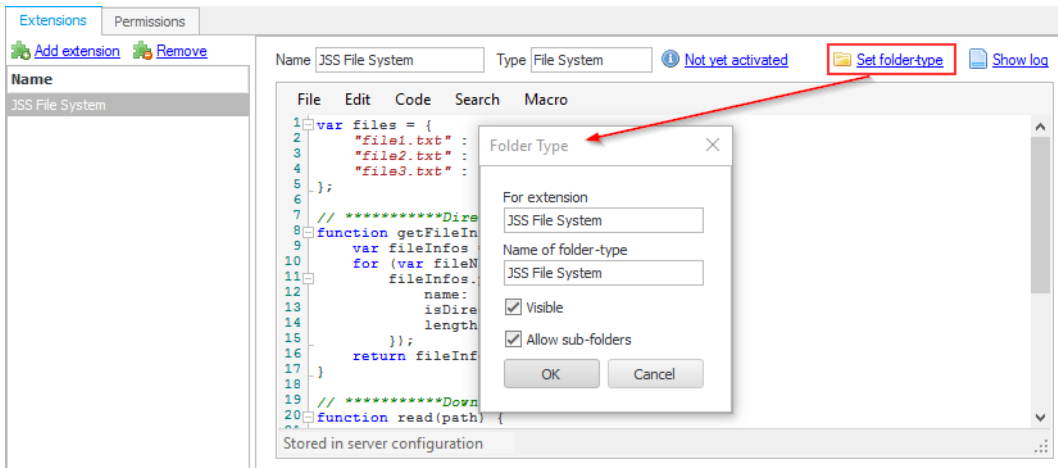
### Custom Command extensions

Only users who have permission for a particular custom command extension may invoke the commands that it defines. Permission may be granted to individual users or groups in the Permissions tab, in the Extensions panel, as shown in the example below:



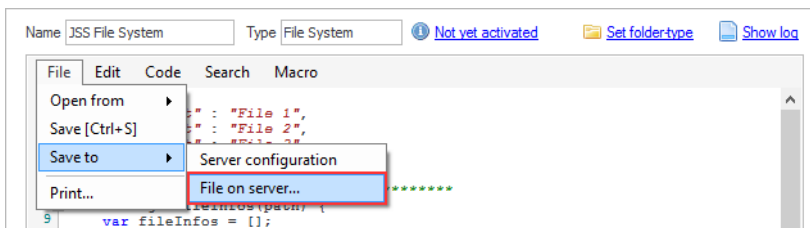
## File-System extensions

Folders that use a custom file-system extension may only be created, if they have a folder-type defined for them. This is done in the 'Folder Types' tab of the Extensions panel.

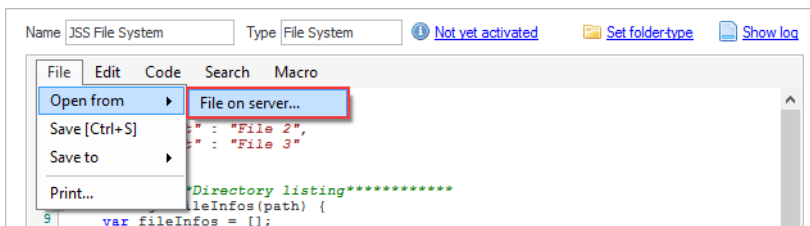


## Saving a JSS extension

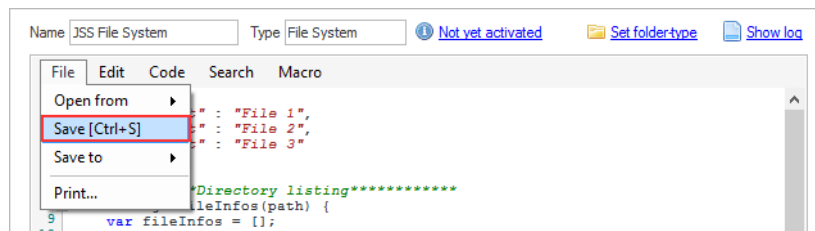
For the JSS code in the CompleteFTP editor, you can save it by choosing 'File/Save to/File on server...' then choose where to save the file and name the file.



After that, the file can be loaded by 'File/Open from/File on server...'.



For a quick save after editing, you can use 'File/Save [Ctrl+S]' or just click on 'Apply Changes'.



# JSS Authentication Extensions

Use this type of extension to implement any user-name/password authentication scheme, or to support public key authentication in SFTP. JSS Authentication Extensions are known as *authenticators*. The server calls functions in your extension to find out whether or not a particular user-name/password combination or user/public key combination is valid. Your class can do whatever it needs to do to work out whether or not it is valid, such as calling a webservice or invoking an RPC.

## Creating a JSS Authentication Extension

General instructions on building CompleteFTP JSS extensions may be found [here](#).

There are two ways to implement a JSS Authentication Extension. The [simple JSS Authentication Extension](#) is the most straightforward, and should be attempted first. If more flexibility is required, consider writing an [advanced JSS Authentication Extension](#).

For step by step instructions on how to create your own authentication method, please refer to Step-by-step guide: [Add my own authentication method](#).

Note that once an authenticator has been written and Apply Changes, it will be automatically registered in CompleteFTP. It can be enabled/disabled by selecting/deselecting the Enable check box in the *General User Settings* dialog box, accessible from the Users tab in the manager, as shown below.

## Precedence

CompleteFTP includes a number of predefined authenticators, such as the [external database authenticator](#), the [gateway authenticator](#) (Enterprise MFT only), the [automatic Windows users \(AWU\) authenticator](#), and the [SAML single sign-on authenticator](#).

Since a user-name can potentially be authenticated by more than one authenticator it's necessary to define the order in which the various authenticators are accessed. This order of precedence is as follows:

1. users explicitly defined within CompleteFTP (both non-Windows and Windows).
2. [external database users](#), [gateway users](#), [SAML single sign-on users](#), and [custom authentication extensions](#).
3. [automatic Windows users \(AWU\)](#).

For example, if a user with a given user-name is explicitly defined in CompleteFTP, but authentication using that user-name is also available via an extension, then the extension will never be used to authenticate the user since it has lower precedence. This can sometimes cause confusing when testing.

When a JSS Authentication Extension is used, the *defaultExtension* user (which may be found in Users panel and by selecting "Show system users/folders/sites" in the main form's Options menu) is used as a template for user details once login has been successful.

The screenshot shows the 'General user settings' dialog box in CompleteFTP. At the top, there are buttons for 'Add user', 'Remove', and 'General user settings'. Below these is a table with columns: 'Enabled', 'Username', 'Type', 'Home Folder', and 'Groups'. The first row shows a user 'u1' with 'Non-Windows' type, home folder '/Home/u1', and group 'users'. Below this table are buttons for 'Select', 'Groups', and 'Two-factor auth status'. The main section is titled 'AUTHENTICATORS'. It has a dropdown menu 'For site:' set to 'Default Site'. Below this is a table with columns: 'Authenticator', 'Enabled', 'Log in as', and a link to 'Configure...'. The rows are: 'Database' (disabled, logs in as 'defaultDatabase'), 'Gateway' (disabled, logs in as 'defaultExtension'), 'JSS Authentication 1' (enabled, logs in as 'defaultExtension', highlighted with a red box), 'Single sign-on / SAML' (disabled, logs in as 'defaultExtension'), and 'Windows/AD Group' (disabled, logs in as 'defaultWindows').

Authenticator	Enabled	Log in as	
Database	<input type="checkbox"/>	defaultDatabase	<a href="#">Configure...</a>
Gateway	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
JSS Authentication 1	<input checked="" type="checkbox"/>	defaultExtension	
Single sign-on / SAML	<input type="checkbox"/>	defaultExtension	<a href="#">Configure...</a>
Windows/AD Group	<input type="checkbox"/>	defaultWindows	<a href="#">Configure...</a>

# Simple JSS Authentication Extension

Use simple authentication when the password, password-hash or public key are known by the authenticator.

The only method that needs to be implemented is:

```
loadUserInfo(userName, userInfo)
```

When the server calls this method in your authenticator, it supplies user information that enables your extension to look up that particular user's authentication details from an external source. This function returns authentication information for the user, with the given user-name.

The first argument is simply a string containing the supplied user-name.

The second argument, userInfo, has the following fields:

- userName - the user-name which is presented by the client.
- siteName - the name of the site that the client is connecting to.
- remoteEndPoint - object with two fields, ip and port.
- session - object containing information about the session.

The function may either return the password-hash, or an object containing any of the following fields:

- password - the password of the user.
- passwordHash - the password hash (either MD5 or SHA).
- passwordSalt - the salt of the password hash.
- homeDirectory - home directory to be used (Windows path).
- groups - list of groups that the user is to be a member of.
- tags - object whose fields are arbitrary name-value pairs, that will be available via the system.user.tags field for the lifetime of the user's session.
- dsaPublicKey - a DSA key (in any common format).
- dsaPublicKeys - an array of DSA keys if there are more than one.
- rsaPublicKey - an RSA key (in any common format).
- rsaPublicKeys - an array of RSA keys, if there are more than one.
- mustChangePassword - force the user to change their password (SSH only).

If the user cannot be authenticated, then null should be returned.

## Examples

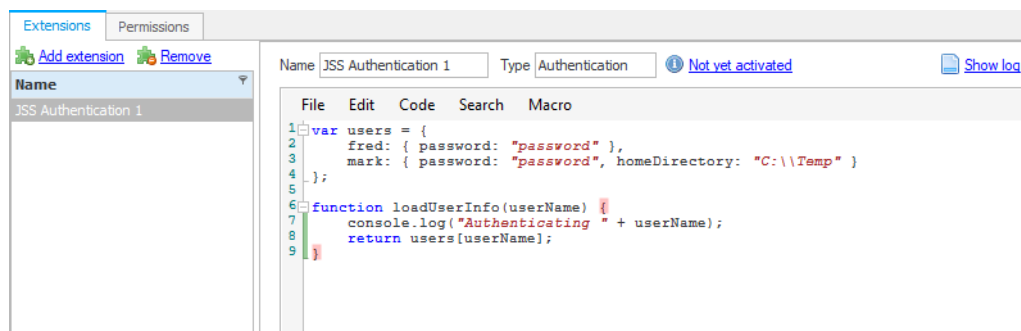
### Example 1

This example illustrates an authenticator, which shows sample source code of using a global associative array (a.k.a. object) to store user account data.

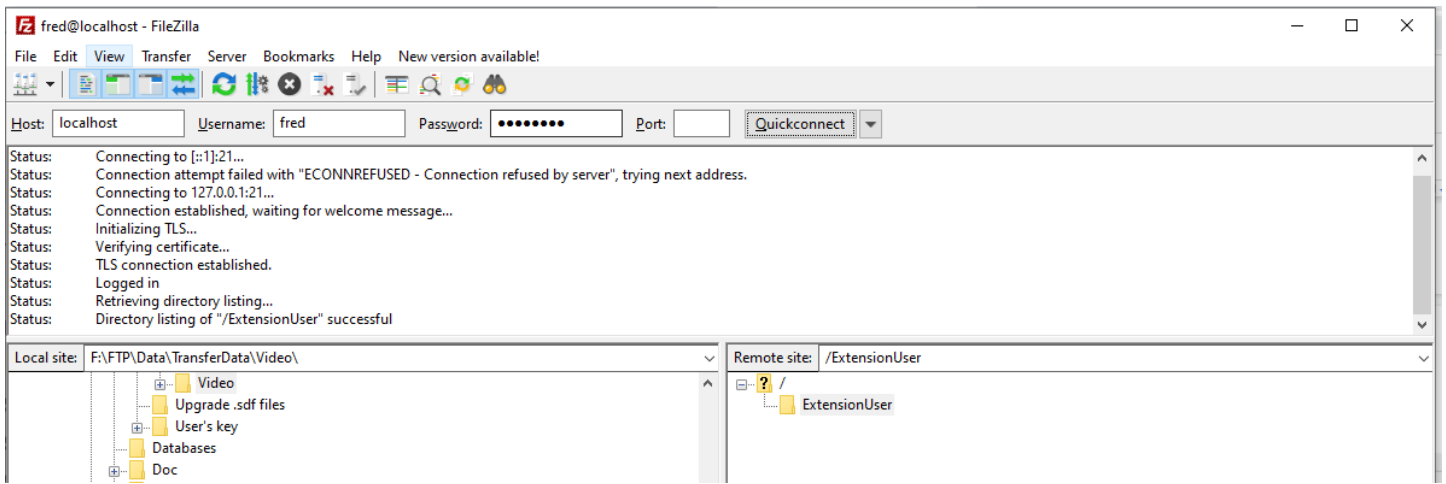
```
var users = {
  fred: { password: "password" },
  mark: { password: "password", homeDirectory: "C:\\Temp" }
};

function loadUserInfo(userName) {
  console.log("Authenticating " + userName);
  return users[userName];
}
```

Add the script above into the CompleteFTP Manager.



Use the username and password in the script to login with FileZilla.



## Example 2

This example illustrates an authenticator, which authenticates by reading user accounts from a JSON file.

```
var users = system.getFile("/Public/examples/users.json").readObject();

function loadUserInfo(userName) {
    return users[userName];
}
```

Before using this extension, the user needs to create a file `users.json` with the content below:

```
{
  "user1" : { "password": "123456", "homeDirectory": "F:\\FTP" }
}
```

Place the file into the directory, `C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Public\examples`.

To check this, use the username and password in the `.json` file to login via FileZilla, it should be successful.

## Example 3

This example illustrates an authenticator which authenticates from an SQL Server Compact database. The database has a single table, `Users`, with two columns, `userName` and `password`.

```
// open the database - will create if doesn't exist
var db = system.openDatabaseSync("/Public/examples/users.sdf");

// authentication function
function loadUserInfo(userName) {
    var userInfo = null;
    db.readTransaction(function(transaction) {
        var res = transaction.executeSql("SELECT * FROM Users WHERE userName=?", [userName]);
        if (res.rows.length>0) {
            var row = res.rows[0];
            userInfo = { password: row.password };
        }
    });
    return userInfo;
}
```

To apply this example code, create a database file with the content as shown below.

SQL Compact Query Analyzer - C:\Users\Natalie\Desktop\users.sdf

File Query Database Help

Database Information

File name: users.sdf  
Date Created: 4/3/2020 3:43:00 PM  
Last Modified: 5/5/2020 9:46:15 AM  
Version: 3.5.8080.0  
Password Protected: False  
File size: 36.00kB

Schema Summary

Schema Information

Tables

Users

ID	username	password	path	groups
1	user1	123456	NULL	NULL
2	Testing	123456	NULL	NULL
NULL	NULL	NULL	NULL	NULL

Copy this file into C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Public\examples.

Enterprise Distributed Technologies > Complete FTP > Public > examples

Search examples

Name	Date modified	Type	Size
users.json	4/3/2020 3:03 PM	JSON File	1 KB
users.sdf	4/3/2020 4:11 PM	SQL Server Comp...	36 KB

Add JSS authentication extensions.

Extensions Permissions

Add extension Remove

Name

JSS Authentication 1

Name: JSS Authentication 1 Type: Authentication Not yet activated Show log

File Edit Code Search Macro

```

1 // open the database - will create if doesn't exist
2 var db = system.openDatabaseSync("/Public/examples/users.sdf");
3
4 // authentication function
5 function loadUserInfo(userName) {
6   var userInfo = null;
7   db.readTransaction(function(transaction) {
8     var res = transaction.executeSql("SELECT * FROM Users WHERE userName=?", [userName]);
9     if (res.rows.length>0) {
10      var row = res.rows[0];
11      userInfo = { password: row.password };
12    }
13  });
14  return userInfo;
15 }

```

Stored in server configuration

Login with the username and password, which are set in the database file. Login should be successful.

Testing@localhost - FileZilla

File Edit View Transfer Server Bookmarks Help New version available!

Host: localhost Username: Testing Password: ..... Port: Quickconnect

Status: Verifying certificate...

Status: TLS connection established.

Status: Logged in

Status: Retrieving directory listing...

Status: Directory listing of "/ExtensionUser" successful

#### Example 4

This example illustrates an authenticator, which authenticates by fetching the user account data via HTTP (JSS supports HTTP GETs).

```

var usersFile = http.get("http://localhost/Examples/users.json").body;
var users      = JSON.parse(usersFile);

function loadUserInfo(userName) {
  return users[userName];
}

```

To execute this example, the user needs to create a users.json file with content.

```

{
  "user1" : { "password": "123456", "homeDirectory": "F:\FTP" }
}

```

Then place it into /Public/Examples/.

Login by using the username and password which are defined in that file, and the login should be successful.

/ExtensionUser

No files in this folder...

Server 1 (primary) - CompleteFTP Manager

EXTENSIONS

Step-by-step guides

Performance warning

APPLY CHANGES

Extensions

Permissions

[Add extension](#) [Remove](#)

Name

JSS Authentication 1

Name JSS Authentication 1

Type Authentication

Activated successfully

[Show log](#)

File Edit Code Search Macro

```
1 var usersFile = http.get("http://localhost/examples/users.json").body;  
2 var users = JSON.parse(usersFile);  
3  
4 function loadUserInfo(userName) {  
5     return users[userName];  
6 }
```



# Advanced JSS Authentication Extension

Use advanced authentication when the password or password hash are unknown by the authenticator; usually when authentication is being handled by an external system.

There are two methods that needs to be implemented are:

```
checkUserName(userName, userInfo)
authenticate(userName, password, authInfo)
```

When the server call these methods in your authenticator, it supplies user information that enables your extension to look up that user's authentication details from an external source.

Optionally, a `setPassword()` method can be defined, which enables users who have logged in via this authenticator to change their passwords:

```
setPassword(userName, newPassword, oldPassword)
```

## **function checkUserName(userName, userInfo)**

Indicates whether or not the given user-name is valid. If true is returned the CompleteFTP won't allow other authenticators to try to authenticate this user.

If it's not possible to determine whether or not a user-name is valid also having a password then this method may return true regardless of the value of `userName`. However, if this is done then no authenticators (with lesser precedence) will be utilized.

The first argument is a string containing the user-name. If that's all that's required then the second argument may be left out.

The second argument, `userInfo`, has the following fields:

- `userName` - the user-name being presented by the client.
- `siteName` - the name of the site that the client is connecting to.
- `remoteEndPoint` - object with two fields, ip and port.
- `session` - object containing information about the session.

## **function authenticate(userName, password, authInfo)**

Authenticates the client. This method will only be called if `checkUserName` return true.

The first and second arguments are strings containing the user-name and the password, respectively. If that's all that's required then the third argument may be left out.

The third argument, `authInfo`, has the following fields.

- `userName` - the user-name being presented by the client.
- `password` - the password being presented by the client.
- `siteName` - the name of the site that the client is connecting to.
- `remoteEndPoint` - object with two fields, IP and port.
- `session` - object containing information about the session.
- `authenticationMethod` - authentication method being used - either 'Password' or 'PublicKey'.
- `keyAlgorithm` - the key algorithm - either 'DSA' or 'RSA'.

This method may return either a boolean or an object containing the following fields:

- `isCorrectPassword` - indicates whether or not the password is correct (only used if `authenticationMethod` is 'Password').
- `isValidKey` - indicates whether or not the public key is correct (only used if `authenticationMethod` is 'PublicKey').
- `homeDirectory` - home directory to be used (Windows path).
- `mustChangePassword` - indicates whether or not the user will be forced to change their password (SFTP/SCP only).
- `groups` - array of the names of groups (strings) of which the user should be a member.
- `tags` - object whose fields are arbitrary name-value pairs that will be available via the `system.user.tags` field for the lifetime of the user's session.

The methods `isValidDSAKey(key)` and `isValidRSAKey(key)` may be used to check keys. The argument must be a public key in a common format such as OpenSSH.

## **function setPassword(userName, newPassword, oldPassword)**

Sets the password of the user to `newPassword`.

# JSS File-System Extensions

A JSS file-system extension is a set of Javascript functions that define the behaviour of the files and folders, within a folder of the CompleteFTP virtual file-system.

CompleteFTP's virtual file-system is available via the `system.getFile(path)` method. This method returns objects of the `File` class. The fact that an object is returned, does not imply that a file exists on that path; this may be checked using the `exists()` method.

## Developing a JSS file-system extension

General instructions on writing CompleteFTP JSS extensions may be found [here](#). Remember that file-system extensions are linked to the file-system via a folder-type, so an entry for the file-system extension being developed must be present in the folder-type tab, before a folder of that type may be added to the virtual file-system. It's created automatically but it's important to understand this linkage. Once a folder-type has been created the file-system extension may be tested by creating a folder of that type in the virtual file-system, and then listing that folder using a client such as FileZilla.

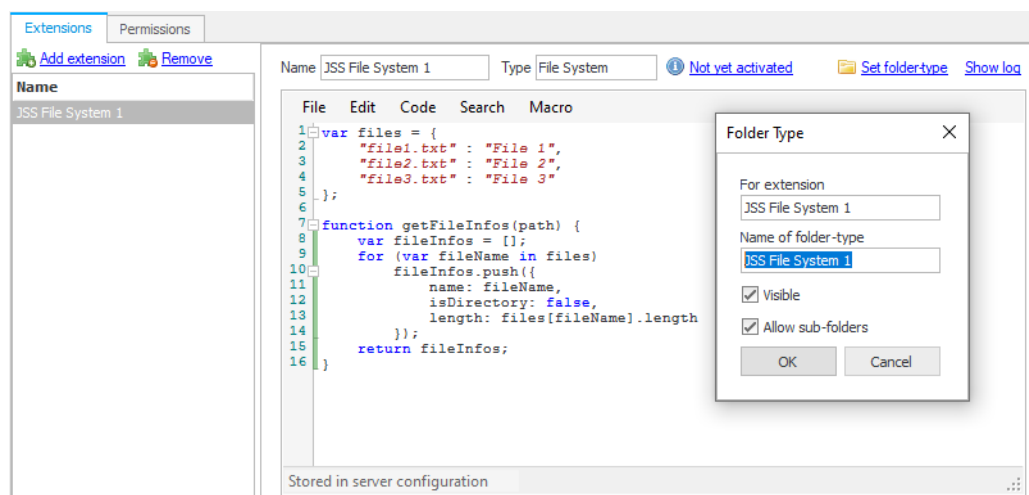
At a minimum, a JSS file-system extension must implement a function called `getFileInfos()`, which returns a file-listing. For example, this code will generate a listing containing three files (`file1.txt`, `file2.txt` and `file3.txt`):

```
var files = {
  "file1.txt" : "File 1",
  "file2.txt" : "File 2",
  "file3.txt" : "File 3"
};

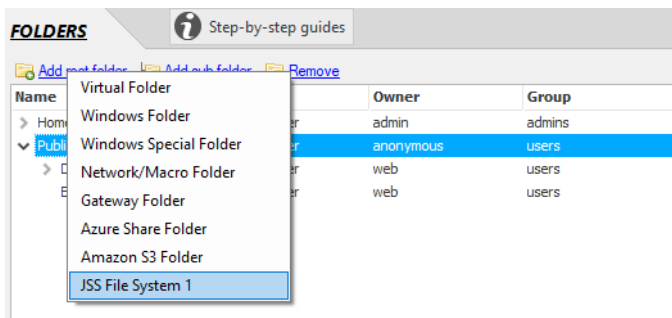
function getFileInfos(path) {
  var fileInfos = [];
  for (var fileName in files)
    fileInfos.push({
      name: fileName,
      isDirectory: false,
      length: files[fileName].length
    });
  return fileInfos;
}
```

`getFileInfos()` returns an array of objects. Each object in the array describes one file: its name (`name`), whether or not it's a directory (`isDirectory`) and its length (`length`). The `path` argument specifies the path of the subdirectory to be listed, relative to the directory that this file-system extension is handling. In this case it may be ignored, since no subdirectories are returned in the listing.

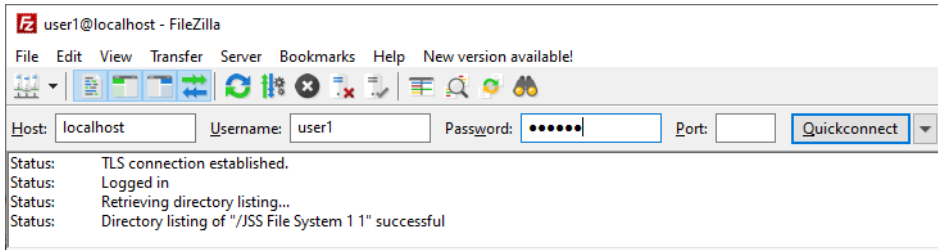
Add the script above into the CompleteFTP Manager.



Then create the folder with the newly created folder and set the login user as owner of that folder.



Then login with FileZilla as that user.



The listing of files will then be generated.

Filename	Filesize	Filetype	Last modified	Filename	Filesize	Filetype	Last modified	Permissions	O
..				..					
test-image1.jpg	47,095	JPG File	8/22/2019 3:03:06 ...	file1.txt	6	Text Docu...	1/1/2000	-rwxrwx---	ai
test-image2.jpg	28,672	JPG File	5/4/2020 10:06:45 ...	file2.txt	6	Text Docu...	1/1/2000	-rwxrwx---	ai
test-image3.jpg	7,580	JPG File	5/4/2020 10:07:01 ...	file3.txt	6	Text Docu...	1/1/2000	-rwxrwx---	ai
test1.txt	1,272	Text Document	4/23/2020 3:15:24 ...						
test2.txt	2,471	Text Document	3/20/2020 5:12:34 ...						
test3.txt	816	Text Document	4/16/2020 1:16:32 ...						

The capability of downloading the 'files' may be added as follows:

```
function read(path) {
  if (files[path])
    return { text: files[path] };
  else
    throw "No such file: " + path;
}
```

Here the `path` argument becomes important since it specifies which file is to be read. The object returned, implicitly defines the type of the contents. Here, the object has a property named `text`. Any of the following may be returned:

Return value:	Used when...
{ file: '/folder/file.dat' }	the file being downloaded is at another location in the virtual file-system.
{ text: 'some text' }	the content of the 'file' being downloaded is text.
{ base64: 'c29tZSB0ZXh0' }	the content of the file is binary. It will be converted to binary before sending to the client.

Uploading may be implemented as follows:

```
function onWriteBegin(session, node, path) {
  return { type: "text" };
}

function write(path, data) {
  files[path] = data;
}
```

The method, `onWriteBegin()`, is called when uploading of the file commences. It instructs CompleteFTP on what to do with the incoming file. In this case, we're expecting text to be uploaded. The `write()` method is called when the upload has completed. Here we simply store it in the `files` object.

`onWriteBegin()` may also return a type of 'file' and 'base64'. If 'file' is returned, then a second property, `path`, must also be returned, which defines the path in the virtual file-system where the incoming file should be stored (e.g. { type: 'file', path: '/mydir/myfile.dat' }). In this case, the content of the file is not buffered in memory, but is streamed straight to the disk so that very large files can be handled. If a binary file is expected, then 'base64' may be returned, in which case a base-64 encode string is passed to the `write()` method.

For File system extension example, please check [here](#).

There are a number of other functions that may be implemented to add other capabilities, such as file deletion and folder creation. The section below is the full list of interfaces.

## Full details of JSS APIs

Full details are provided in the [JSS API Reference](#).

### Folder listing

```
// Return a directory listing as an array of objects. The objects may have the following fields:
//   name - string,
//   isFolder - boolean,
//   length - number (optional),
//   modifiedTime - Date (optional),
//   createdTime - Date (optional),
function getFileInfos(path, session, node)
```

### File reading

```
// Read the file at the given path. Return one of the following:
//   { file: 'path-of-file' },
//   { text: 'text-content' } or
//   { base64: 'base64-encoded-bytes' }
function read(path, session, node)

// Release any used resources at the end of a read operation (optional)
function onReadEnd(path, session, node)
```

### File writing

```
// Instructs CompleteFTP what to do with the file that's being written. Return one of the following:
//   { type: 'file', path: 'path-of-file' },
//   { type: 'text' } or
//   { type: 'base64' }
// If type is 'text' or 'base64' and it is returned, then the content will be passed to the write function once
// the content has been streamed to the server.
function onWriteBegin(path, session, node)

// Called when a file has finished streaming to the server.
// If the onWriteBegin method specified the type as 'text' then the second argument, data, is the
// text and the third argument, length, is the number of characters. If the type is 'base64' then
// data is base64-encoded bytes and length is the number of bytes prior to encoding. If the type
// is 'file' then data is null and length is the size of the file.
function write(path, data, length, session, node)
```

### File attributes

```
// Indicate whether or not the file-length is known (only used for HTTP requests)
function isFileLengthKnown(path, session, node)

// Return the file-length in bytes of the file at the given path. If this method is not provided
// then getFileInfos is called to get the file-length.
function getFileLength(path, session, node)

// Return the time when the file at the given path was last modified. If this method is not
// provided then getFileInfos is called to get the timestamp.
function getModifiedTime(path, session, node)

// Set the time that indicates when the file at the given path was last modified.
function setModifiedTime(path, session, node)

// Return a string that canonically defines the given virtual file-system path from the point of
// view of this adapter (e.g. a Windows adapter returns the Windows path)
function getAdapterFilePath(path, session, node)
```

## Existence/deletion/creation

```
// Indicate whether or not a file exists. Return true or false.
function fileExists(path, session, node)

// Delete the file at the given path. No return value required.
function deleteFile(path, session, node)

// Indicate whether or not a file exists. Return true or false.
function directoryExists(path, session, node)

// Delete the directory at the given path. No return value required.
function deleteDirectory(path, session, node)

// Create a directory at the given path
function createDirectory(path, session, node)
```

## Rename/move/copy

```
// Move/rename the file at fromPath to toPath. No return value required.
function moveFile(fromPath, toPath, session, fromNode, toNode)

// Copy the file at fromPath to toPath. No return value required.
function copyFile(fromPath, toPath, session, fromNode, toNode)
```

## Inheritance

```
// Return the type of file-system adapter to inherit functionality from - "windows" or "default".
function getBaseType()

// Return the Windows path of the file/folder at the given virtual file-system path.
function getWindowsPath(path, session, node)
```

# Example: DB-based file-system extension

## Example

The source-code below implements a JSS file-system extension, where files and folders are stored in a SQL Server CE database.

## Installing and using the extension

To try it, create a new JSS file-system extension, copy the code into it and give it a name such as "DB Folder". Each folder of this type which is created, will create a database in a subfolder of the user's home folder called "fileDBs". The name of the database will be that of the folder that it stores files for, so each DB Folder must have a name that's unique to each user.

## Source code

```

function getFileInfos(path, pattern, session, node) {
    var fileInfos = [];
    getDB(session, node).readTransaction(function(transaction) {
        var res = transaction.executeSql("SELECT name, length, modifiedTime, createdTime FROM Files");
        for (var i=0; i < res.rows.length; i++) {
            var row = res.rows[i];
            fileInfos.push({
                name: row.name,
                isFolder: false,
                length: row.length,
                modifiedTime: row.modifiedTime,
                createdTime: row.createdTime
            });
        }
    });
    return fileInfos;
}

function onWriteBegin() {
    return { type: 'binary' };
}

function write(path, data, length, session, node) {
    getDB(session, node).transaction(function(tx) {
        tx.executeSql("INSERT INTO Files(name, data, length, modifiedTime, createdTime) VALUES (?, ?, ?, ?, ?)",
            [path, data, length, new Date(), new Date()]);
    });
}

function read(path, session, node) {
    var data = null;
    getDB(session, node).readTransaction(function(transaction) {
        var res = transaction.executeSql("SELECT * FROM Files WHERE name=?", [path]);
        if (res.rows.length > 0)
            data = res.rows[0].data;
    });
    return { binary: data };
}

function deleteFile(path, session, node) {
    getDB(session, node).transaction(function(tx) {
        tx.executeSql("DELETE FROM Files WHERE name=?", [path]);
    });
}

function moveFile(fromPath, toPath, session, fromNode, toNode) {
    var success = false;
    getDB(fromNode).transaction(function(tx) {
        var res = tx.executeSql("UPDATE Files SET name=? WHERE name=?", [toPath, fromPath]);
        success = res.rowsAffected > 0;
    });
    return success;
}

var databases = {};
function getDB(session, node) {
    var db = databases[node.path];
    if (!db) {
        var dbFolder = system.getFile(session.fullHomePath + "/fileDBs");
        if (!dbFolder.exists())
            dbFolder.createFolder();
        var folderName = node.path.substring(node.path.lastIndexOf('/') + 1);
        db = databases[node.path] = system.openDatabaseSync(dbFolder.fullPath + "/" + folderName + ".sdf");
    }

    // create the Files table if it doesn't already exist
    if (!tableExists(db, "Files"))
        db.transaction(function(tx) {
            tx.executeSql("CREATE TABLE [Files] ([name] NVARCHAR(1024), [data] IMAGE, [length] INT, [modifiedTime] DATETIME, [createdTime] DATETIME)");
        });
    return db;
}

function tableExists(db, tableName) {
    var rowCount = 0;
    db.readTransaction(function(tx) {
        var res = tx.executeSql("SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = ?", [tableName]);
        rowCount = res.rows.length;
    });
    return rowCount > 0;
}

```

## Description of source code

The code consists of eight functions. The first six, i.e. `getFileInfos`, `onWriteBegin`, `write`, `read`, `deleteFile` and `moveFile`, are called by `CompleteFTP` when directory listings, upload downloads, deletions and moves are required. The other two: `getDB` and `tableExists`, are utility functions used by the first six functions.

## Main methods

Read file information from database.

```
function getFileInfos(path, pattern, session, node) {
  var fileInfos = [];
  getDB(session, node).readTransaction(function(transaction) {
    var res = transaction.executeSql("SELECT name, length, modifiedTime, createdTime FROM Files");
    for (var i=0; i < res.rows.length; i++) {
      var row = res.rows[i];
      fileInfos.push({
        name: row.name,
        isFoldesr: false,
        length: row.length,
        modifiedTime: row.modifiedTime,
        createdTime: row.createdTime
      });
    }
  });
  return fileInfos;
}
```

Write file to database.

```
function write(path, data, length, session, node) {
  getDB(session, node).transaction(function(tx) {
    tx.executeSql("INSERT INTO Files(name, data, length, modifiedTime, createdTime) VALUES (?, ?, ?, ?, ?)",
      [path, data, length, new Date(), new Date()]);
  });
}
```

Read file from database and return data as binary.

```
function read(path, session, node) {
  var data = null;
  getDB(session, node).readTransaction(function(transaction) {
    var res = transaction.executeSql("SELECT * FROM Files WHERE name=?", [path]);
    if (res.rows.length > 0)
      data = res.rows[0].data;
  });
  return { binary: data };
}
```

Delete file out of database.

```
function deleteFile(path, session, node) {
  getDB(session, node).transaction(function(tx) {
    tx.executeSql("DELETE FROM Files WHERE name=?", [path]);
  });
}
```

Move file.

```
function moveFile(fromPath, toPath, session, fromNode, toNode) {
  var success = false;
  getDB(fromNode).transaction(function(tx) {
    var res = tx.executeSql("UPDATE Files SET name=? WHERE name=?", [toPath, fromPath]);
    success = res.rowsAffected > 0;
  });
  return success;
}
```

## Database methods

Returns database object for the given node.



```
function getDB(session, node) {
  var db = databases[node.path];
  if (!db) {
    var dbFolder = system.getFile(session.fullHomePath + "/fileDBs");
    if (!dbFolder.exists())
      dbFolder.createFolder();
    var folderName = node.path.substring(node.path.lastIndexOf('/') + 1);
    db = databases[node.path] = system.openDatabaseSync(dbFolder.fullPath + "/" + folderName + ".sdf");
  }

  // create the Files table if it doesn't already exist
  if (!tableExists(db, "Files"))
    db.transaction(function(tx) {
      tx.executeSql("CREATE TABLE [Files] ([name] NVARCHAR(1024), [data] IMAGE, [length] INT, [modifiedTime] DATETIME, [createdTime] DATETIME);");
    });
  return db;
}
```

Returns true, if a table with the given name exists in the database.

```
function tableExists(db, tableName) {
  var rowCount = 0;
  db.readTransaction(function(tx) {
    var res = tx.executeSql("SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = ?", [tableName]);
    rowCount = res.rows.length;
  });
  return rowCount > 0;
}
```

## Common methods

The following JSS API methods are used in the source-code. Their documentation may also be found in the [JSS API Reference](#).

### 1. readTransaction ( [callback] )

Run the read transaction contained in the given callback function. A `SQLTransactionSync` object is passed as an argument to the callback. This may be used to execute queries. The transaction is committed if the callback returns normally, and rolled back if it throws an exception.

#### Parameters:

- [callback] Function optional  
callback function
  - [transaction] `SQLTransactionSync` optional  
Transaction object which can be used to execute SQL queries.

### 2. transaction ( [callback] )

Run the non-read transaction contained in the given callback function. A `SQLTransactionSync` object is passed as an argument to the callback. This may be used to execute queries. The transaction is committed if the callback returns normally and rolled back if it throws an exception.

#### Parameters:

- [callback] Function optional  
callback function
  - [transaction] `SQLTransactionSync` optional  
Transaction object which can be used to execute SQL queries.

### 3. executeSql ( sqlStatement arguments ) `SQLResultSet`

Execute the given SQL and return the result as an `SQLResultSet` object.

#### Parameters:

- sqlStatement String  
SQL statement.
- arguments Array

Array containing the arguments to substitute into the statement.

**Returns:**

ResultSet

**4. getFile ( path ) File**

Returns a File object for the given path.

The File object may or may not represent an existing file or directory.

**Parameters:**

- path String

Path of file

**Returns:**

File

# JSS Custom Command Extensions

User-defined commands can be implemented in Javascript - each command is implemented by a Javascript function. The commands may be accessed as SITE commands in FTP/FTP! and as regular commands in SSH terminal. They may also be invoked via HTTP/HTTPS (if enabled as a [web feature](#)) using a URL of the format shown below, and the result is returned in [JSON format](#).

`http://hostname/CustomCommand/commandName?arg1=val1`

The following function defines a command called 'Hello' which takes one argument:

```
function Hello(who)
{
    return 'Hello ' + who;
}
```

All arguments are strings.

Commands added in this way are available to all users, who have permission for the 'CustomCommandScript' extension. These permissions may be modified in the Permissions tab of the Extensions panel.

.NET objects may be instantiated using the 'new' keyword, followed by the fully qualified name, e.g.

```
function Hello(who)
{
    var b = new System.Text.StringBuilder();
    b.Append('Hello').Append(' ').Append(who);
    return b.ToString();
}
```

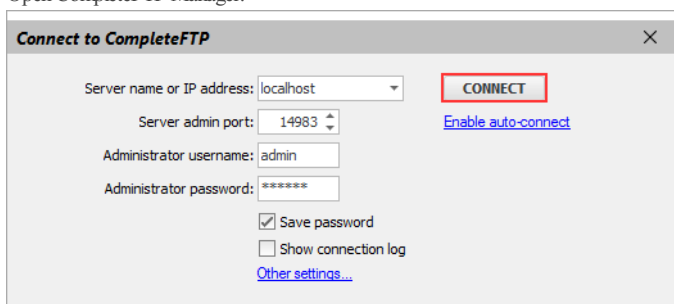
Static methods and properties of .NET objects may be accessed using their fully qualified name, e.g.

```
function HostName()
{
    return System.Environment.MachineName;
}
```

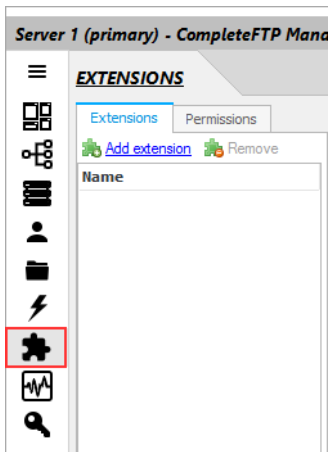
## Creating a JSS custom command extension

To add JSS custom commands extension:

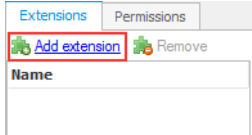
1. Open CompleteFTP Manager.



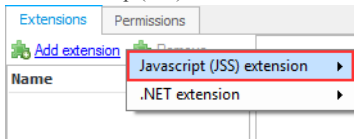
2. Select the Extensions panel and then select the Extensions tab in sub menu.



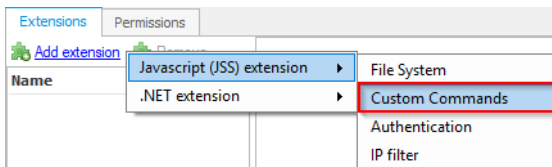
3. Click 'Add extension'.



4. Select 'Javascript(JSS) extension'.



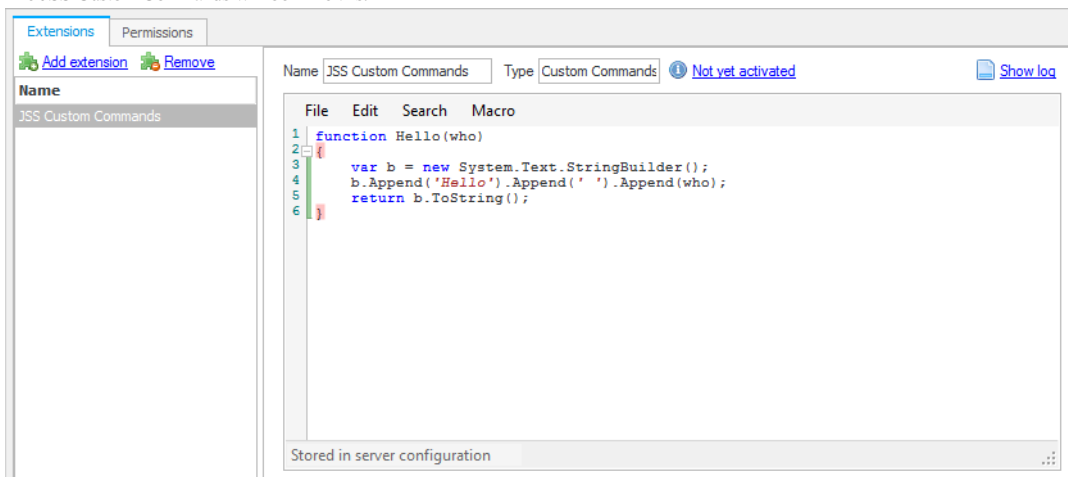
5. Select 'Custom Command'.



6. Write your command.

```
function Hello(who)
{
    var b = new System.Text.StringBuilder();
    b.Append('Hello').Append(' ').Append(who);
    return b.ToString();
}
```

7. The JSS Custom Commands will look like this.



8. Select Permissions tab and grant permission for groups or users to use this extension.

ExtensionsPermissions

Permissions for Custom-Command and Web-app Extensions

[Add group permission](#)
[Remove group permission](#)

WebApp/Custom-Command Ext...

JSS Custom Commands

users

Add user permission

Remove user permission

WebApp/Custom-Command Ext...

User

Add group permission

Give all members of the selected group permission to execute commands in the selected extension.

9. Click the Apply Changes button.

EXTENSIONS

ExtensionsPermissions

Permissions for Custom-Command and Web-app Extensions

[Add group permission](#)
[Remove group permission](#)

WebApp/Custom-Command Ext...

JSS Custom Commands

users

WebApp or custom-command extension

WebApp or Custom-command extension that group has permission to.

APPLY CHANGES

10. Test your JSS Extension by using an FTP client.

user1@localhost - FileZilla

File Edit View Transfer Server Bookmarks Help New version available!

Host: localhost Username: user1 Password: Port: Quickconnect

Status: Retrieving directory listing...

Status: Calculating timezone offset of server...

Status: Timezone offset of server is 25200 seconds.

Status: Directory listing of "/Home/user1" successful

Enter custom command

Please enter raw FTP command. Using raw ftp commands will clear the directory cache.

SITE Hello user1

OK Cancel

user1@localhost - FileZilla

File Edit View Transfer Server Bookmarks Help New version available!

Host: localhost Username: user1 Password: Port: Quickconnect

Status: Connection established, waiting for welcome message...

Status: Initializing TLS...

Status: Verifying certificate...

Status: TLS connection established.

Status: Logged in

Status: Retrieving directory listing...

Status: Directory listing of "/Home/user1" successful

Command: SITE Hello user1

Response: 200-Hellouser1

Response: 200 SITE command successful.

# JSS IP Filter Extensions

Use this type of extension to implement any IP filtering scheme required. IP filtering is used to deny or allow access by specific IP addresses or ranges of IP addresses. CompleteFTP offers two **inbuilt filtering** schemes, auto-banning and rule-based, so a custom IP filter extension need only be implemented if neither of these schemes satisfy requirements. Examples of other types of IP filtering that could be implemented include:

- IP filtering by country using a geo-location web-service.
- IP filtering by blacklisted user-name.

Sample code for both of these are provided below.

## Creating a JSS IP Filter Extension

General instructions on building CompleteFTP JSS extensions may be found [here](#).

An IP filter extension must implement a single Javascript function with the signature:

```
function getFilterAction(ipAddress)
```

The `ipAddress` is a string containing the IP address. The function must return one of the following data-types:

- A boolean where `true` means allow and `false` means deny.
- A string whose value is `allow`, `deny` or `allowAlways`. The last value will cause auto-banning to be turned off for that IP address, meaning that there's no limit to the number of authentication failures allowed.
- An object with two fields: `action` and `reason`. The value of the `action` field may be either of the two types above, and the value of the `reason` field is a text description of why the connection was disallowed. This description will be logged.

### Example: Filtering by country

There are many geo-location web-services available, as well as free ones such as [ipstack.com](#). This web-service will return geo-location data for an IP address in response to a simple HTTP GET request, of the format which is set to **JSON**, then a JSON object is returned which includes a field, `country_code`, which is the two-letter code of the country in which the IP address is located. To filter by country, we just need to check this field. If we wish to only allow access from, say Australia, then we just need to verify that the `country_code` is `AU`. This can be done by the following code:

```
function getFilterAction(ipAddress) {
    var accessKey = "Insert IPstack access key here";
    var geoData = http.json("http://api.ipstack.com/"
        + ipAddress + "?access_key=" + accessKey + "&format=1");
    return geoData.country_code == "AU";
}
```

The `http.json` function makes a HTTP request to a given URL, parses the (JSON) response, and returns it as a Javascript object. Note that connections from `localhost` are always allowed.

It's a good idea to provide a reason, so that the logs show why a connection was denied. This may be done as follows:

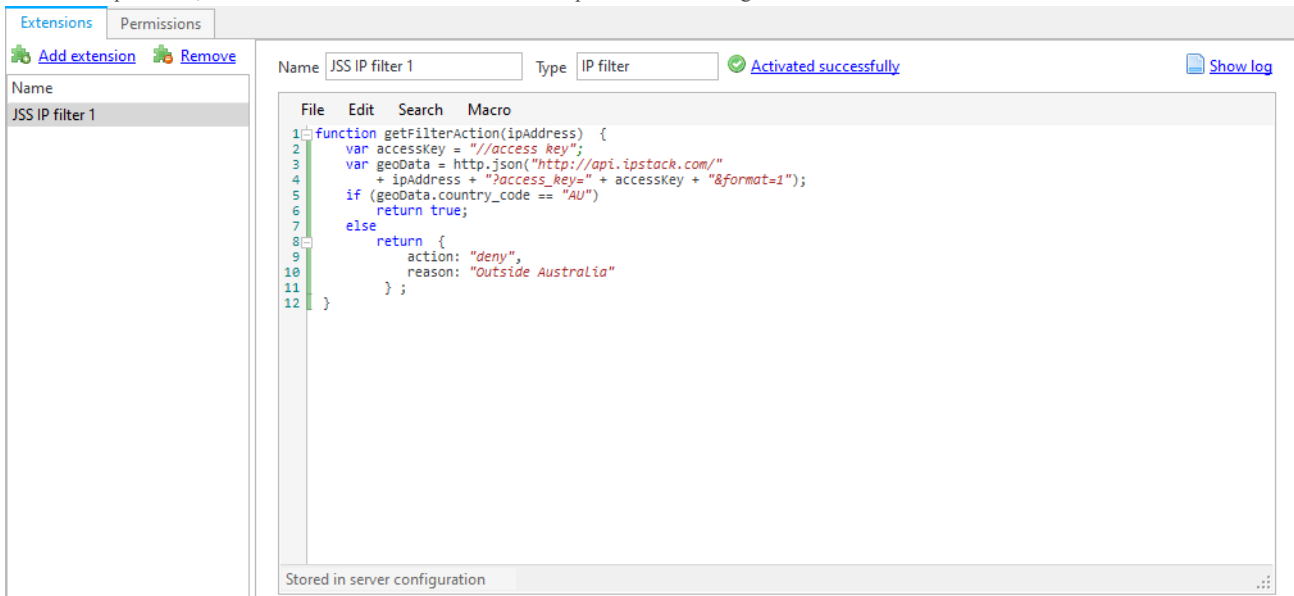
```
function getFilterAction(ipAddress) {
    var accessKey = "Insert IPstack access key here";
    var geoData = http.json("http://api.ipstack.com/"
        + ipAddress + "?access_key=" + accessKey + "&format=1");
    if (geoData.country_code == "AU")
        return true;
    else
        return {
            action: "deny",
            reason: "Outside Australia"
        };
}
```

Note that:

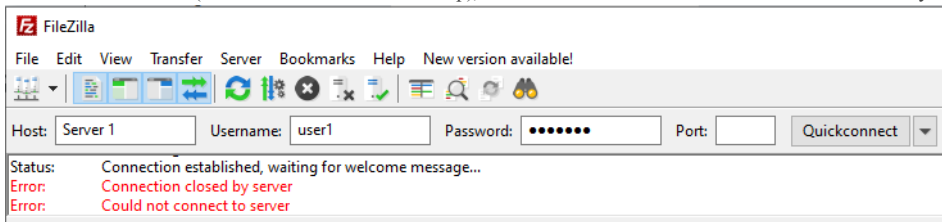
- For the sake of performance, it would be a good idea to cache the geo-location data, so that fewer requests would need to be made to the geo-location web-service. Some sort of storage would be required for this, as shown in the case in the example.
- For the `access key`, it can be found in this web-service [ipstack.com](#).

The following shows you how to apply the code above:

1. Add the example code 1, into the JSS IP filter extension in the Extensions panel of CFTP Manager.



2. From the second server (which has the IP from the first step), connect to the main server. It will not connect successfully.



### Example: Filtering by user-name blacklist

One of the attacks most commonly observed in logs, is the **password attack**. This involves trying to log in using commonly used user-name/password combinations. Some of the most common user-names are `root`, `admin` and `Administrator`, so it makes sense to automatically disallow any IP addresses from which login-attempts employing those user-names have been made.

To implement this in JSS we need storage for our user-name blacklist and also for our list of IP addresses that have used the blacklisted user-names. An SQLite database is an easy option as it's lightweight, stores data in a single file and the required driver is included with CompleteFTP. The database must have two tables:

- `UserNames` where the blacklisted user-names will be stored.
- `IPAddresses` where the IP addresses that have attempted to log in with those user-names, will be stored.

An SQLite database stores its data in a single file, so we must specify the place where this file is stored. In the code below, it's assumed that a folder has been created at `/Databases` in the virtual file-system and that this folder maps to a suitable location in the Windows file-system.

Assuming that the `IPAddresses` table contains all the IP addresses from which clients have tried to authenticate using blacklisted user-names, the following IP filter extension code will block connection attempts from those IP addresses:

```

function getFilterAction(ipAddress) {
    if (isIPAddressDenied(db, ipAddress))
        return {
            action: "deny",
            reason: "previously used blacklisted user-name"
        };
    else
        return true;
}

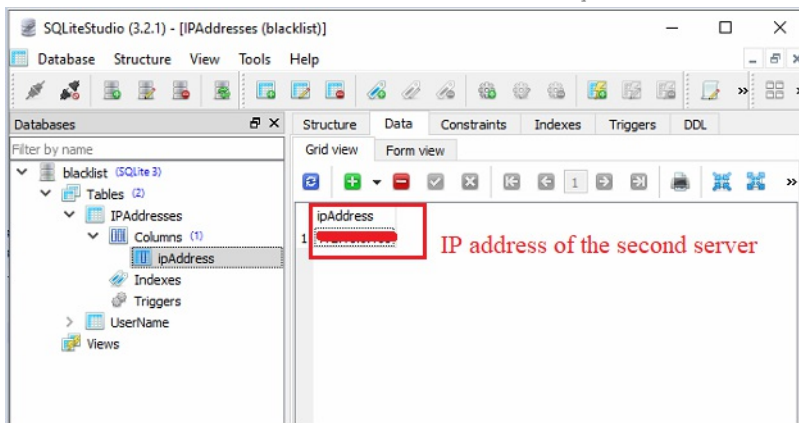
function isIPAddressDenied(db, ipAddress) {
    var dbFile = system.getFile("/Databases/blacklist.sqlite3");
    if (!dbFile.exists())
        return false;
    else {
        db = system.openDatabaseSync(dbFile.fullPath);
        var found = false;
        db.readTransaction(function(tx) {
            var res = tx.executeSql("SELECT * FROM IPAddresses WHERE ipAddress = ?", [ipAddress]);
            found = res.rows.length > 0;
        });
        return found;
    }
}

```

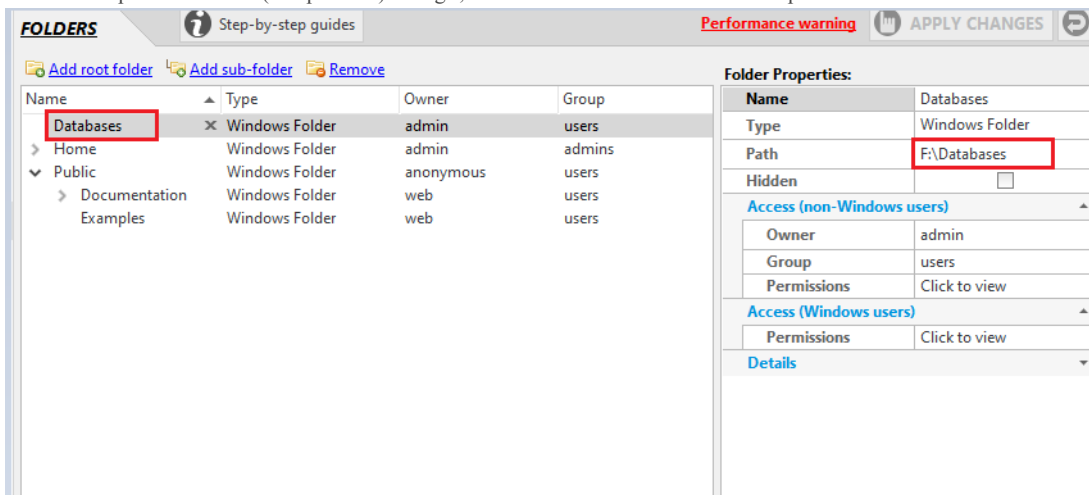
The `isIPAddressDenied` function first checks to see if the database file exists, if it doesn't, then it returns `false` to indicate that the IP address is not banned. If the database file does exist, then it opens it and queries the `IPAddresses` table for the given IP address, and returns `true` if it finds it there.

The following is the way in which to apply the code shown above:

1. Create Database folder which includes database file names `blacklist.sqlite3` with the content as shown below:

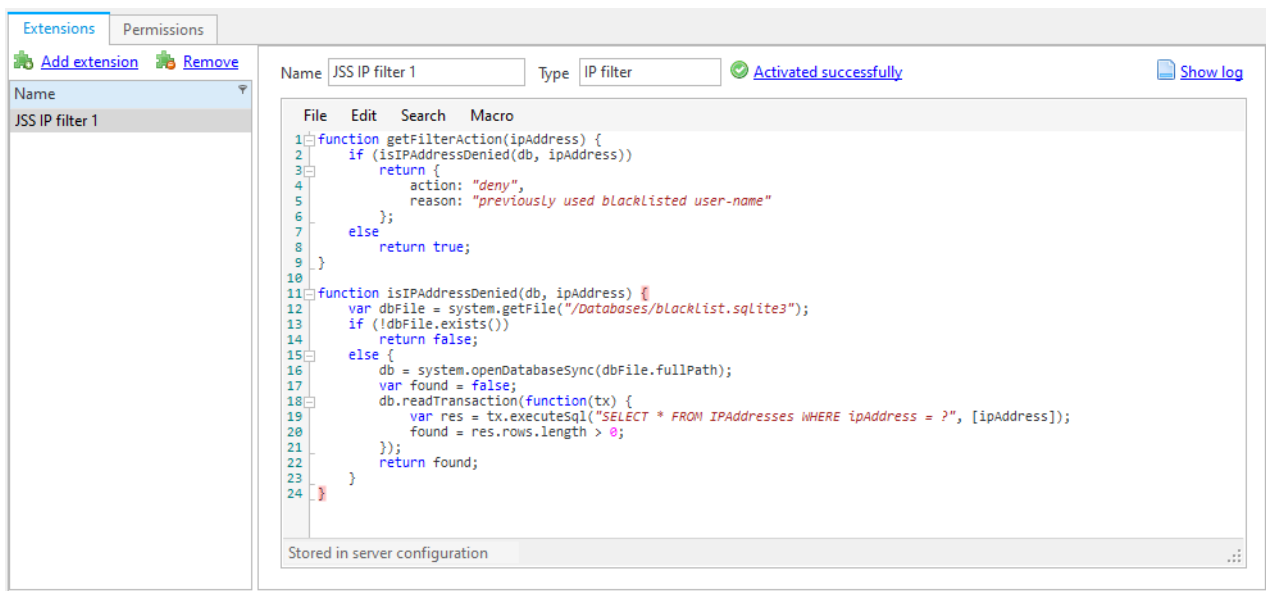


2. Put that Database folder in a directory.
3. In the Folders panel of the CFTP(CompleteFTP) Manager, create a Windows Root Folder which maps to the Database folder shown above.

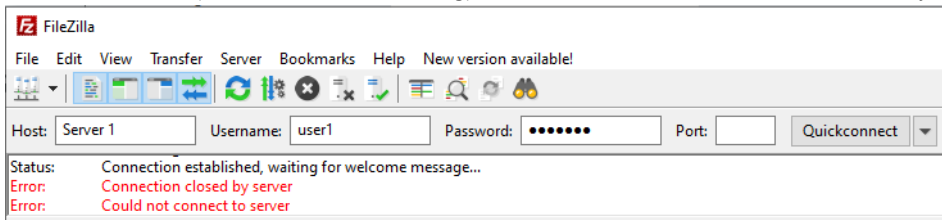


4. Add the example code 2 into the JSS IP filter extension in the Extensions panel of the CFTP(CompleteFTP) Manager.

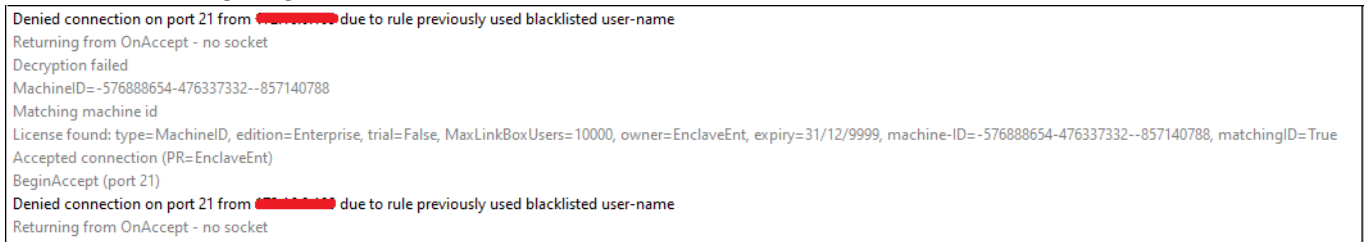




5. From the second server (which has the IP from the first step), connect to the main server. It will not connect successfully.



6. User should see the following message which shows that this IP is in the blacklist of the first server, so it can't connect to the first server.



Since IP filters don't see the user-names associated with incoming connections, we need an **authentication extension** to check the user-name against the blacklist, and add the IP address of the client if the user-name is found there. The following code achieves this:

```

function loadUserInfo(userName, userInfo) {
    console.log(userName);
    var db = getDB();
    if (isUserNameBlacklisted(db, userName))
        addIPAddress(db, userInfo.remoteEndPoint.ip);
    return null;
}

function isUserNameBlacklisted(db, userName) {
    var found = false;
    db.readTransaction(function(tx) {
        var res = tx.executeSql("SELECT * FROM UserNames WHERE userName = ?", [userName]);
        found = res.rows.length > 0;
    });
    console.log(found);
    return found;
}

function addIPAddress(db, ipAddress) {
    db.transaction(function(tx) {
        tx.executeSql("INSERT INTO IPAddresses VALUES (?)", [ipAddress]);
    });
}

function getDB() {
    var dbFile = system.getFile("/Databases/blacklist.sqlite3");
    var db;
    if (!dbFile.exists()) {
        db = system.openDatabaseSync(dbFile.fullPath);
        db.transaction(function(tx) {
            tx.executeSql("CREATE TABLE UserNames (userName TEXT)");
            tx.executeSql("CREATE TABLE IPAddresses (ipAddress TEXT)");
        });
    } else
        db = system.openDatabaseSync(dbFile.fullPath);
    return db;
}

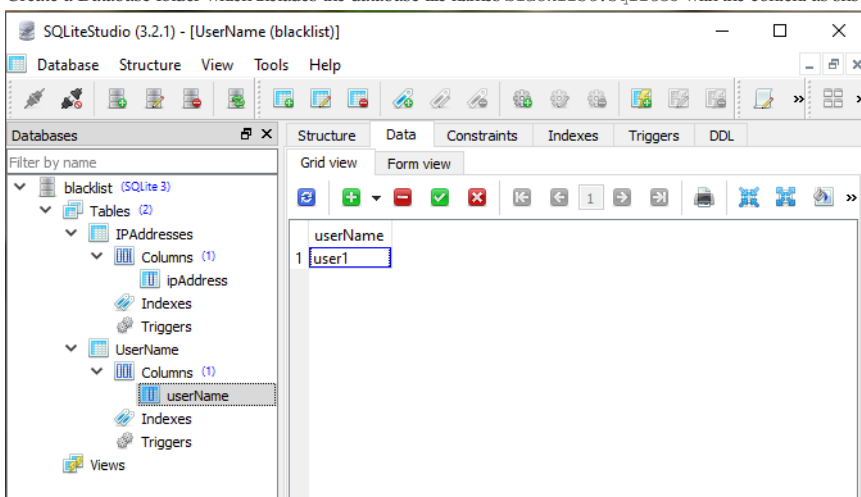
```

The `loadUserInfo` function is called, when authentication of a user is required, as described in the section on [Simple JSS Authentication Extension](#) `isUserNameBlacklisted` queries the database to see if the given user-name is in the blacklist. `addIPAddress` adds the given IP address to the list of banned IP addresses, which is the list that the associated IP filter extension checks (above). The `getDB` function checks to see if the database has already been created. If it has, then it opens it and returns an object that can be used to access it. If it hasn't then it creates it as well as the required tables.

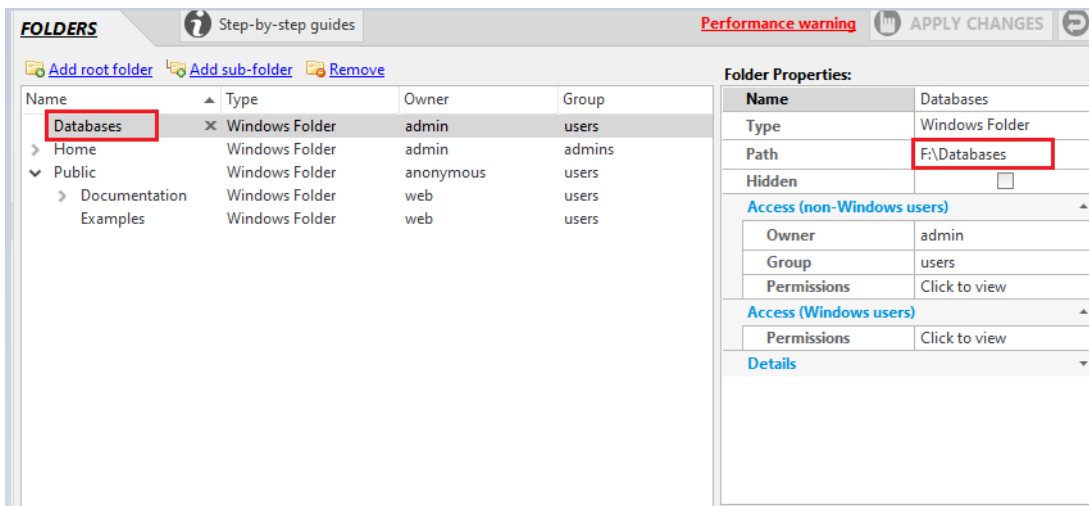
Adding user-names to the blacklist can be done with a SQLite editing tool such as [DB Browser for SQLite](#), or alternatively a [JSS web-app](#) could be developed that provides a way to do this.

To apply this example, a user can follow the steps below:

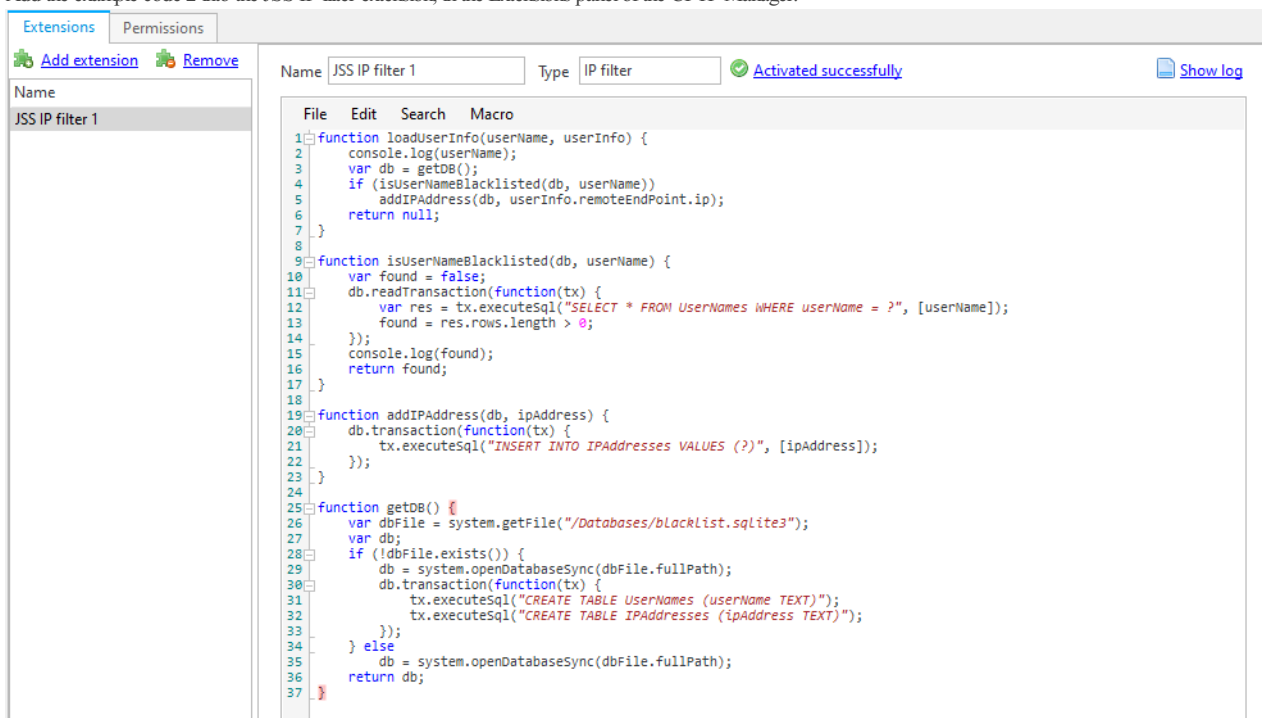
1. Create a Database folder which includes the database file names `blacklist.sqlite3` with the content as shown below:



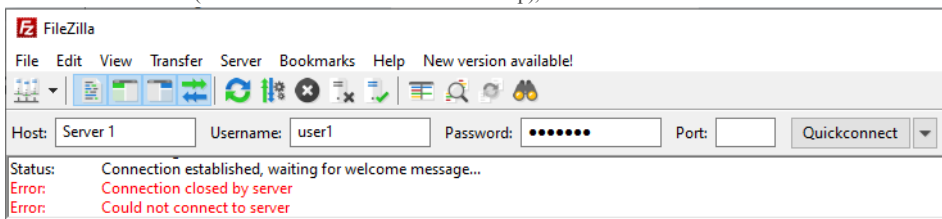
2. Put that Database folder into a directory.
3. In the Folders panel of the CFTP Manager, create a Windows Root Folder which maps to the Database folder mentioned above.



4. Add the example code 2 into the JSS IP filter extension, in the Extensions panel of the CFTP Manager.



5. From the second server (which has the IP mentioned in the first step), connect to the main server. It will not connect successfully.



# Overview of Sample Scripts

Sample Scripts includes seven samples below:

1. **Purge old files**: describes how to delete files that are older than 'N' days in specified directories.
2. **Low drive-space email**: describes how to send an e-mail when the drive-space is slow.
3. **Maintain user stats**: describes how to maintain user stats.
4. **Trigger mobile notifications from CompleteFTP**: describes how to trigger mobile notifications directly from CompleteFTP by using Notify17.
5. **Add user to group from SSH terminal**: describes how to add the user to the group from the SSH terminal.

# How to delete files that are older than 'N' days in specified directories

This JSS script will be for a scheduled process trigger that deletes files that are older than 'N' (Number of) days in specified directories.

Now let us see how to do this. The following steps will configure CompleteFTP to perform this trigger:

1. Define the folders which you want the script to work on. The folders below include some files, which are older than 'N' days.

Name	Date modified	Type
Directory1	4/20/2020 3:21 PM	File folder
Directory2	4/20/2020 3:21 PM	File folder
Directory3	4/20/2020 3:21 PM	File folder

This PC > DATA (F:) > FTP > Test > Directory1

Name	Date modified	Type	Size
test1.txt	5/4/2020 9:43 AM	Text Document	0 KB
test2.txt	5/4/2020 9:43 AM	Text Document	0 KB
test3.txt	5/4/2020 9:44 AM	Text Document	0 KB
test4.txt	5/4/2020 9:44 AM	Text Document	0 KB
test5.txt	5/4/2020 9:44 AM	Text Document	0 KB
test6.txt	4/23/2020 3:15 PM	Text Document	2 KB
test7.txt	3/20/2020 5:12 PM	Text Document	3 KB
test8.txt	4/16/2020 1:16 PM	Text Document	1 KB

2. Go to the CompleteFTP Manager, open the Folder panel, then create 3 Windows Folders which match the 3 folders in the step 1.

**FOLDERS** Step-by-step guides APPLY CHANGES

Add root folder Add sub-folder Remove

Name	Type	Owner	Group
Directory1	Windows Folder	admin	users
Directory2	Windows Folder	admin	users
Directory3	Windows Folder	admin	users
Home	Windows Folder	admin	admins
Public	Windows Folder	anonymous	users
Documentation	Windows Folder	web	users
Examples	Windows Folder	web	users

**Folder Properties:**

Name	Directory1
Type	Windows Folder
Path	F:\FTP\Test\Directory1
Hidden	<input type="checkbox"/>
Access (non-Windows users)	
Owner	admin
Group	users
Permissions	Click to view
Access (Windows users)	
Permissions	Click to view
Details	

3. Go to Events panel, select Schedule event and set it up as Every '1' day (although you can set any time/number of days you wish).

**Set Schedule**

Set schedule as:

- ☐ Specific dates/times
- ☒ Even time intervals
- ☐ Days of the week
- ☐ Cron expression (advanced)

Every 1 days  
Note: Day-counter restarts from one every month

Next 50 events:

Tue	2020-04-21	00:00:00
Wed	2020-04-22	00:00:00
Thu	2020-04-23	00:00:00
Fri	2020-04-24	00:00:00
Sat	2020-04-25	00:00:00
Sun	2020-04-26	00:00:00
Mon	2020-04-27	00:00:00
Tue	2020-04-28	00:00:00
Wed	2020-04-29	00:00:00
Thu	2020-04-30	00:00:00
Fri	2020-05-01	00:00:00
Sat	2020-05-02	00:00:00
Sun	2020-05-03	00:00:00
Mon	2020-05-04	00:00:00
Tue	2020-05-05	00:00:00
Wed	2020-05-06	00:00:00
Thu	2020-05-07	00:00:00
Fri	2020-05-08	00:00:00
Sat	2020-05-09	00:00:00
Sun	2020-05-10	00:00:00
Mon	2020-05-11	00:00:00
Tue	2020-05-12	00:00:00
Wed	2020-05-13	00:00:00
Thu	2020-05-14	00:00:00

OK Cancel

4. Then add the script below in to the JSS type script.

```

var MAX_FILE_AGE_DAYS = 30;

var foldersToPurge = [
  "/Directory1",
  "/Directory2",
  "/Directory3"
];

// returns the difference between two days in days
function subtractDates(date1, date2) {

  // Convert both dates to milliseconds
  var date1ms = date1.getTime();
  var date2ms = date2.getTime();

  // Calculate the difference in milliseconds
  var ONE_DAY = 1000*60*60*24;
  return (date2ms - date1ms)/ONE_DAY;
}

for (var i in foldersToPurge) {

  // get listing of folder
  var folder = system.getFile(foldersToPurge[i]);
  var files = folder.GetFiles();

  for (var j in files) {
    var file = files[j];
    if (file.isFolder)
      continue;

    // Work out the age of the file
    var now = new Date();
    var modTime = file.modifiedTime;
    var age = subtractDates(modTime, now);

    // If it's old then delete it
    if (age > MAX_FILE_AGE_DAYS) {
      console.log(file.fullPath + " is " + Math.floor(age) + " old - deleting");
      file.remove();
    }
  }
}

```

5. The process triggers will look like this:

Name:

Events: ☒ Scheduled (At 0:00, every 30 days - [change](#))

Type: ☐ Program ☐ Batch script ☐ Powershell script ☒ JSS script ☐ FTP script

Script:

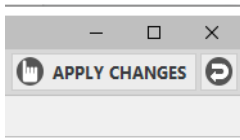
```

1 var MAX_FILE_AGE_DAYS = 30;
2 var foldersToPurge = [
3   "/Directory1",
4   "/Directory2",
5   "/Directory3"
6 ];
7
8 // returns the difference between two days in days
9 function subtractDates(date1, date2) {
10   // Convert both dates to milliseconds
11   var date1ms = date1.getTime();
12   var date2ms = date2.getTime();
13   // Calculate the difference in milliseconds
14   var ONE_DAY = 1000*60*60*24;
15   return (date2ms - date1ms)/ONE_DAY;
16 }
17
18 for (var i in foldersToPurge) {

```

Run as  on

6. Apply the changes.



You should check the relevant folders after the time you set (1 day), the files which are older than 30 days (set in the code) will be deleted as below:

This PC > DATA (F:) > FTP > Test > Directory1				
	Name	Date modified	Type	Size
✦	test1.txt	5/4/2020 9:43 AM	Text Document	0 KB
✦	test2.txt	5/4/2020 9:43 AM	Text Document	0 KB
✦	test3.txt	5/4/2020 9:44 AM	Text Document	0 KB
✦	test4.txt	5/4/2020 9:44 AM	Text Document	0 KB
✦	test5.txt	5/4/2020 9:44 AM	Text Document	0 KB
✦	test6.txt	4/23/2020 3:15 PM	Text Document	2 KB
	test8.txt	4/16/2020 1:16 PM	Text Document	1 KB

If the trigger is checked, it will run continuously and will automatically check these folders every day.

# Send e-mail when drive-space is low

This JSS script will do a scheduled process trigger that emails administrators when the free space on a given drive falls below a certain percentage.

First of all, the script should check the free space of the drive by the code below:

```
function GetFreeDriveSpace(drive) {
    var drives = System.IO.DriveInfo.GetDrives();
    for (var i=0; i<drives.Length; i++) {
        if (drives[i].Name.ToLower().Substring(0, 1)==drive.toLowerCase()[0])
            return drives[i].TotalFreeSpace;
    }
    throw "Drive not found: " + drive;
}
```

Then using a similar script to check the total space of the drive:

```
function GetDriveSpace(drive) {
    var drives = System.IO.DriveInfo.GetDrives();
    for (var i=0; i<drives.Length; i++) {
        if (drives[i].Name.ToLower().Substring(0, 1)==drive.toLowerCase()[0])
            return drives[i].TotalSize;
    }
    throw "Drive not found: " + drive;
}
```

This script focuses on sending an email to administrators after checking the space of the given drive. So, the script below will help you on it:

```
//define the minimum percentage of free space to send email
var minPercentFreeSpace = 5;

//to get free Drive space
function GetFreeDriveSpace(drive) {
    var drives = System.IO.DriveInfo.GetDrives();
    for (var i=0; i<drives.Length; i++) {
        if (drives[i].Name.ToLower().Substring(0, 1)==drive.toLowerCase()[0])
            return drives[i].TotalFreeSpace;
    }
    throw "Drive not found: " + drive;
}

//to get total Drive space
function GetDriveSpace(drive) {
    var drives = System.IO.DriveInfo.GetDrives();
    for (var i=0; i<drives.Length; i++) {
        if (drives[i].Name.ToLower().Substring(0, 1)==drive.toLowerCase()[0])
            return drives[i].TotalSize;
    }
    throw "Drive not found: " + drive;
}

mail.smtp.server = "email server";
mail.smtp.port = "email port";
mail.smtp.userName = "my email";
mail.smtp.password = "my password";
mail.smtp.enableSSL = true;

var currentPercentFreeSpace = GetFreeDriveSpace(drive) * 100 / GetDriveSpace(drive);
if (currentPercentFreeSpace < minPercentFreeSpace)
    mail.send("my email", "admin email", "Warning: low drive space on server", "Server has only " + currentPercentFreeSpace + "% free space");
```

To enter this script into CompleteFTP, please follow these steps:

1. Add the script above into the Events panel of the CFTP Manager with a Scheduled event set for every 1 hour.



**Set Schedule**

Set schedule as:

☐ Specific dates/times

☒ Even time intervals

☐ Days of the week

☐ Cron expression (advanced)

Every

Note: Hour-counter restarts from zero every day

Next 50 events:

Thu 2020-04-23 18:00:00
Thu 2020-04-23 19:00:00
Thu 2020-04-23 20:00:00
Thu 2020-04-23 21:00:00
Thu 2020-04-23 22:00:00
Thu 2020-04-23 23:00:00
Fri 2020-04-24 00:00:00
Fri 2020-04-24 01:00:00
Fri 2020-04-24 02:00:00
Fri 2020-04-24 03:00:00
Fri 2020-04-24 04:00:00
Fri 2020-04-24 05:00:00
Fri 2020-04-24 06:00:00
Fri 2020-04-24 07:00:00
Fri 2020-04-24 08:00:00
Fri 2020-04-24 09:00:00
Fri 2020-04-24 10:00:00
Fri 2020-04-24 11:00:00
Fri 2020-04-24 12:00:00
Fri 2020-04-24 13:00:00
Fri 2020-04-24 14:00:00
Fri 2020-04-24 15:00:00
Fri 2020-04-24 16:00:00

OK Cancel

Name: Low drive-space email

Events: ☒ Scheduled (Every hour - [change](#))

[Choose](#)

Type: ☐ Program ☐ Batch script ☐ Powershell script ☒ JSS script ☐ FTP script

Script:

```

1 //define the minimum percentage of free space to send email
2 var minPercentFreeSpace = 5;
3
4 //to get free Drive space
5 function GetFreeDriveSpace(drive) {
6     var drives = System.IO.DriveInfo.GetDrives();
7     for (var i=0; i<drives.Length; i++) {
8         if (drives[i].Name.ToLower().Substring(0, 1)==drive.toLowerCase()[0])
9         {
10             return drives[i].TotalFreeSpace;
11         }
12     }
13     throw "Drive not found: " + drive;
14 }
15
16 //to get total Drive space
17 function GetDriveSpace(drive) {

```

Run as Default on Default [Show log](#)

2. Apply the changes.

**APPLY CHANGES**

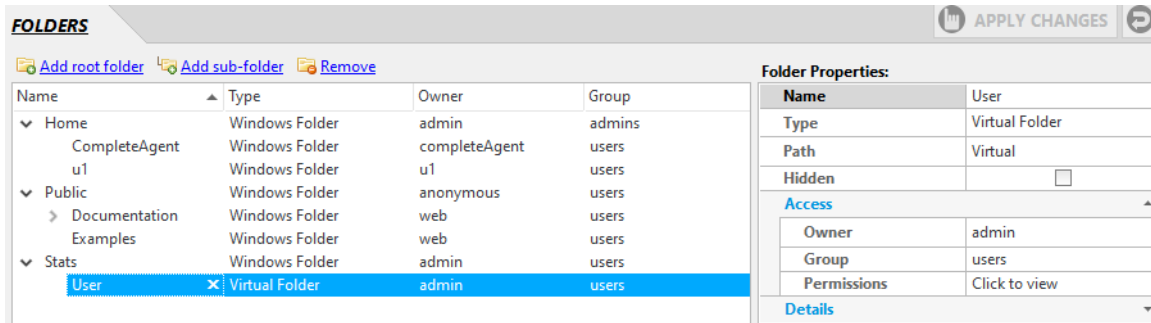
After applying the changes, every hour, it will check your drive, if your free space on a given drive falls below a certain percentage, an email should now be sent to admin in order to notify them about the space deficit.

# Maintain user stats

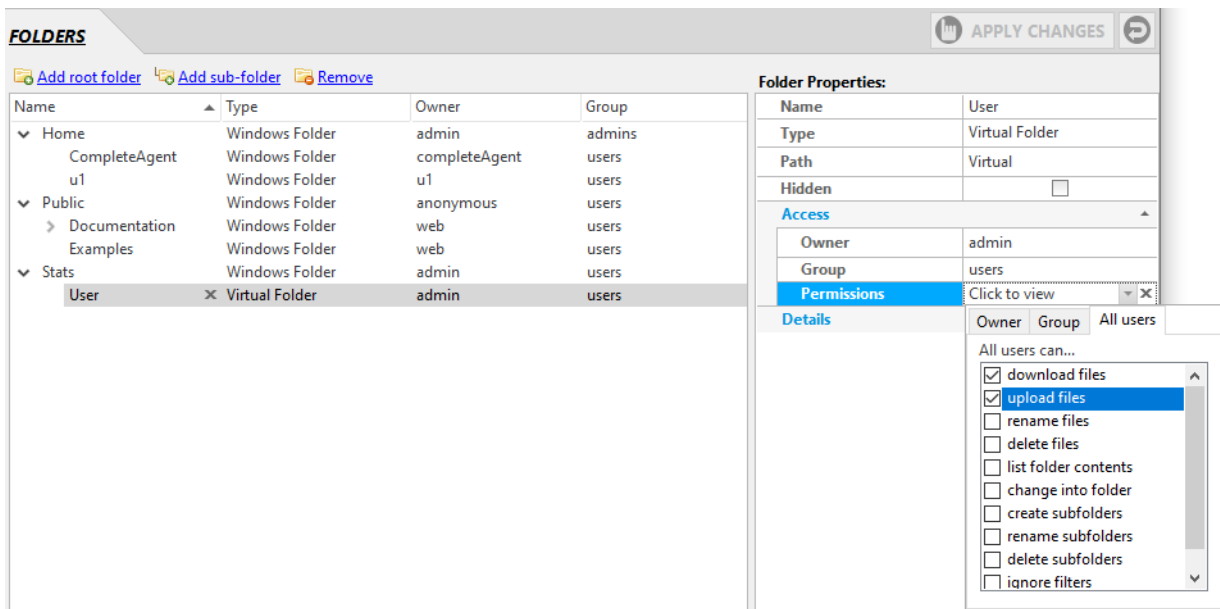
This tutorial will implement a JSS process trigger that maintains user statistics, such as number of uploads/downloads, logins, total bytes uploaded, etc..It will be an event-handler that writes to your chosen data-store, e.g. JSON DB, embedded database, Excel spreadsheet.

Now let's see how to set it up.

1. In Windows, create the folder: C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Stats.
2. Create a Virtual folder under this.



3. Enable the 'Download files' and 'Upload files' permissions under 'Folder Properties' for 'All users'.



4. Open the Events panel in the CompleteFTP Manager, then add the new process trigger, and set the name to Record user stats. After this, choose the Log in event, select JSS script as the script type and paste the following script into the script field:

```
var db = new JsonDB('/Stats');
var userStatRef = '/Stats/User/logins';
var userInfo
try {
  userInfo = db.read(userStatRef);
} catch (e) {
  userInfo = {
    loginCount: 0
  };
}
userInfo.loginCount++;
userInfo.lastLogin = new Date();
db.write(userStatRef, userInfo);
```

Name: Record user stats

Events: ☒ Log in [Choose](#)

Errors: ☒ Trigger on success ☐ Trigger on error

Folder filter: \* ☐ Inverse

User filter: (All users) ☐ Inverse

Site filter: (All sites) ☐ Inverse

Type: ☐ Program ☐ Batch script ☐ Powershell script ☒ JSS script ☐ FTP script

Script:

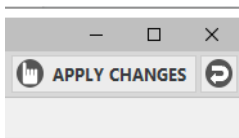
```

1 var db = new JsonDB('/Stats');
2 var userStatRef = '/Stats/User/Logins';
3 var userInfo
4 {
5   try {
6     userInfo = db.read(userStatRef);
7   } catch (e) {
8     userInfo = {
9       loginCount: 0
10    };
11  }
12  userInfo.loginCount++;
13  userInfo.lastLogin = new Date();
14  db.write(userStatRef, userInfo);

```

Grant permissions of: Default [Show log](#)

5. Click on 'Apply changes'.



Now check to see what happens when you login with Filezilla and upload a file.

Filename	Filesize	Filetype	Last modified
..			
test-image1		JPG File	8/22/2019 3:03:06 ...
test-image2		JPG File	5/4/2020 10:06:45 ...
test-image3		JPG File	5/4/2020 10:07:01 ...
test1.txt		Text Document	4/23/2020 3:15:24 ...
test2.txt		Text Document	3/20/2020 5:12:34 ...
test3.txt		Text Document	4/16/2020 1:16:32 ...

Filename	Filesize	Filetype	Last modified	Permissions
..				
Empty directory listing				

If you look in the folder: C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Stats, there should be a new folder with the name of the login user and it should include a json file which has the following content:

```

{
  "loginCount": 1,
  "lastLogin": "2020-04-21T07:18:48.771Z"
}

```

This PC > Local Disk (C:) > ProgramData > Enterprise Distributed Technologies > Complete FTP > Stats > user1

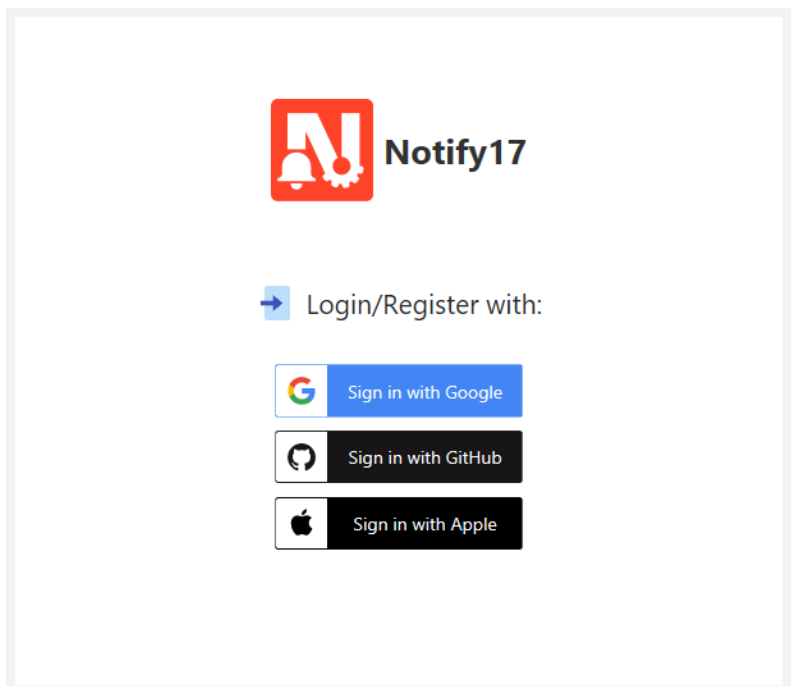
Name	Date modified	Type	Size
logins.json	5/4/2020 2:10 PM	JSON File	1 KB

# How to trigger mobile notifications directly from CompleteFTP by using Notify17

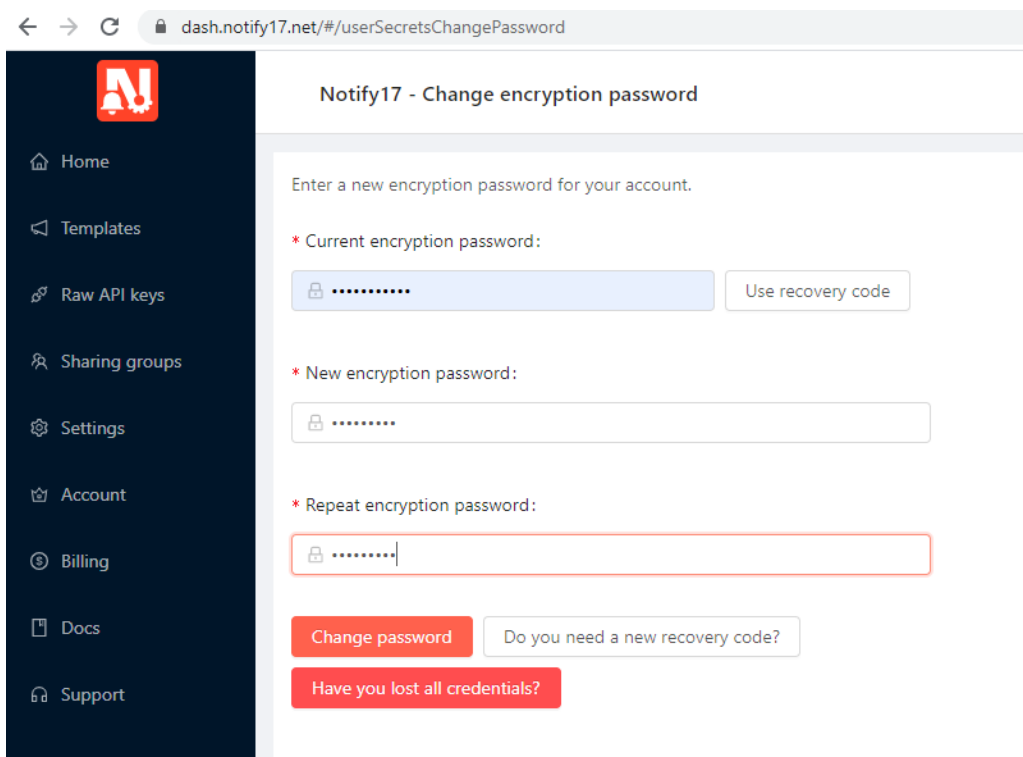
CompleteFTP is able to send email notifications to users regarding important events, such as user logins, file uploads and certificate expirations. But what if your email is inaccessible, or if your inbox is so full that you might miss these notifications - how can you receive the notification? To resolve this issue, CompleteFTP can be integrated with Notify17, which is an innovative system for mobile notifications. Notify17 lets you generate notifications via a simple web requests, and receive them on Android, iOS and the Web.

Shown below, we walk you through the integration process. The following steps will configure CompleteFTP to send a mobile notification, when a file is uploaded to a particular directory:

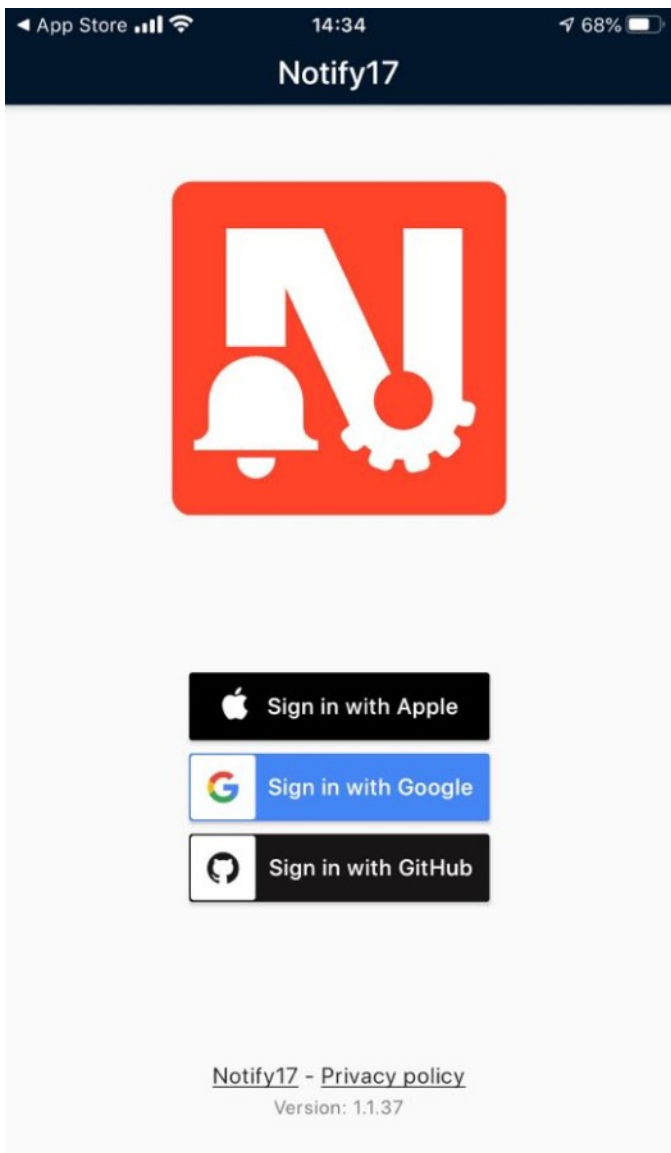
1. Sign in to the [Notify17](#) with Google or GitHub.



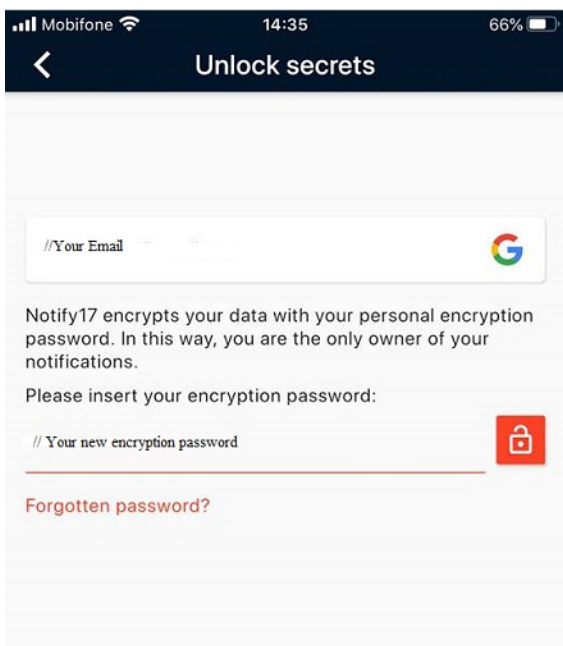
2. Create an encryption password.

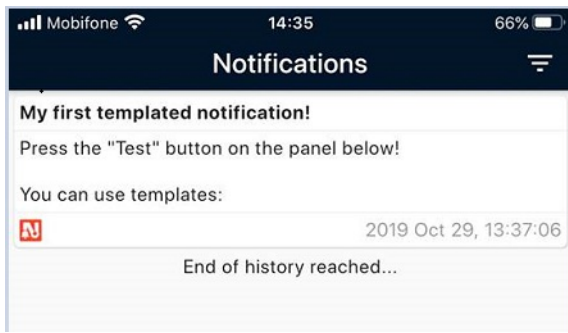


3. Install the Notify17 mobile app (Android or iPhone).

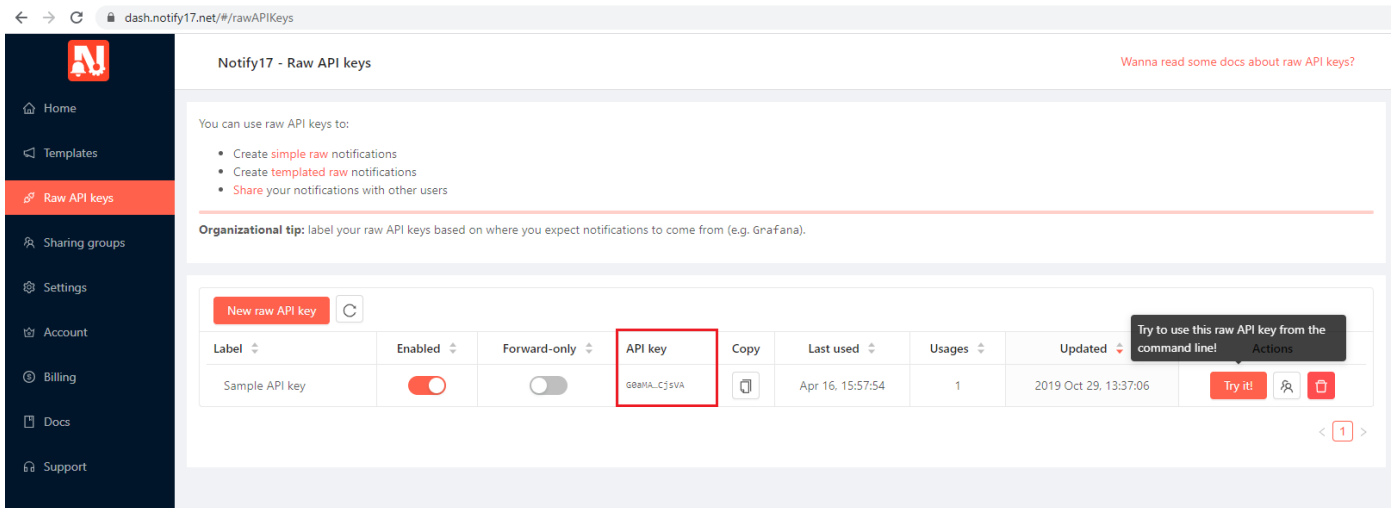


4. Sign in on your phone, and input the encryption password to unlock all your data.





- Go to [Notify Dashboard](#) to get the API key (in API key column).



- Open CompleteFTP manager.
- Go to the Events panel.
- Add a new Process Trigger with Upload file Event.
- Select JSS script as the Type.

Name:

Events: ☒ Upload file

Errors: ☒ Trigger on success ☐ Trigger on error

Folder filter:   ☐ Inverse

User filter:   ☐ Inverse

Site filter:   ☐ Inverse

Type: ☐ Program ☐ Batch script ☐ Powershell script ☒ JSS script ☐ FTP script

Script:

Run as:  on

- Copy the following code into the Script field replacing `...your API key...` in the code, with the Notify17 API key obtained in step 5:

```
var notify17ApiKey = "...your API key...";
var title = event.fileName + " uploaded";
var content = event.fileName + " has been uploaded";

http.post("https://hook.notify17.net/api/raw/" + notify17ApiKey, { title: title, content: content });
```

Name: Notify17

Events: ☒ Upload file [Choose](#)

Errors: ☒ Trigger on success ☐ Trigger on error

Folder filter: \* ☐ Inverse

User filter: (All users) ☐ Inverse

Site filter: (All sites) ☐ Inverse

Type: ☐ Program ☐ Batch script ☐ Powershell script ☒ JSS script ☐ FTP script

Script:

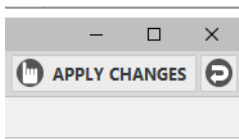
```

1 var notify17ApiKey = "...your API key...";
2 var title = event.fileName + " uploaded";
3 var content = event.fileName + " has been uploaded";
4
5 http.post("https://hook.notify17.net/api/raw/" + notify17ApiKey, { title: title, content: content

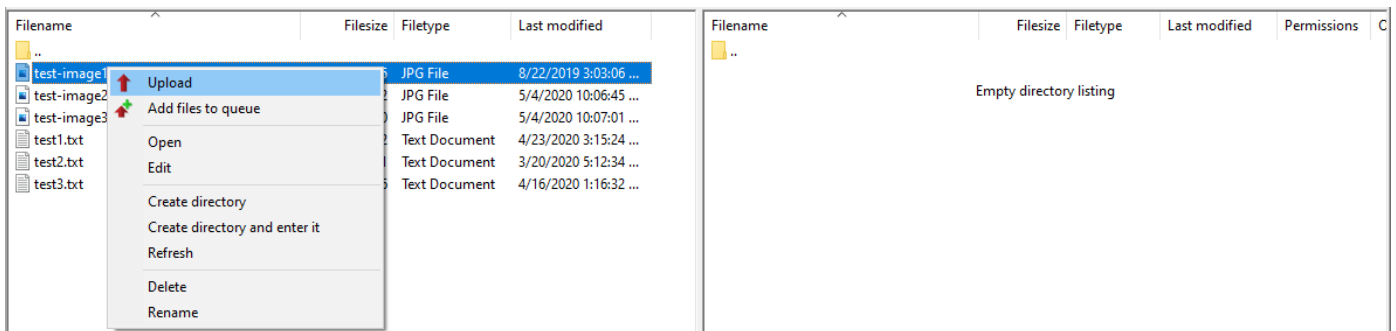
```

Run as: Default on Default [Show log](#)

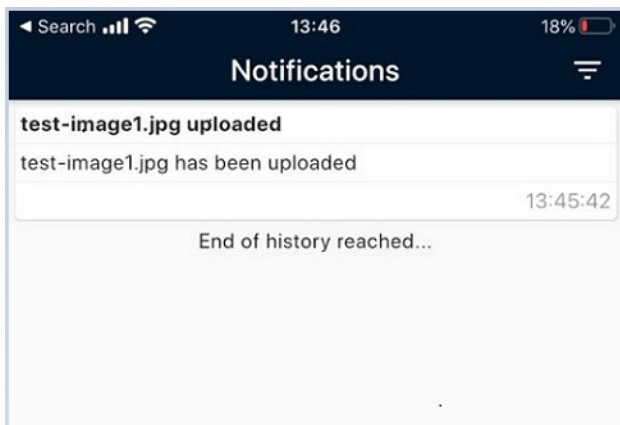
11. Apply the changes.



You should now be set up to receive notifications on your mobile phone. You can test this by uploading a file to CompleteFTP via Filezilla.



And checking your phone.



Note, that if you want to restrict the scope of the notifications, then you can filter by user and/or file-path.

# Add user to group from SSH terminal

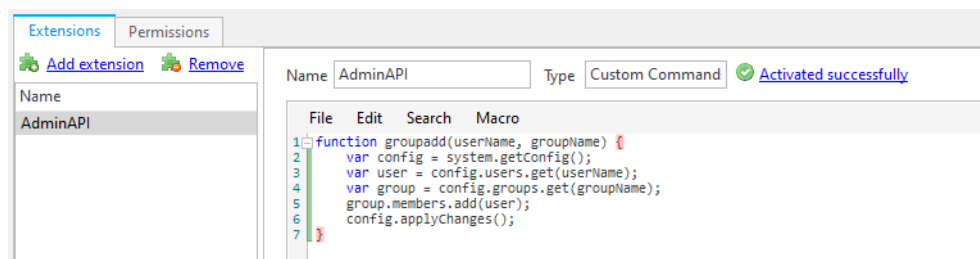
Besides adding a user to any group in the User panel, you can also add a user to a group via an SSH terminal.

Follow the steps shown below in order to do this:

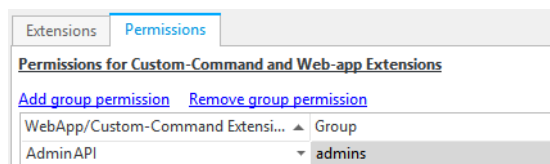
1. Open the CompleteFTP Manager and open the Extensions panel.
2. Click Add extension->Javascript(JSS) extensions->Custom Commands.
3. Enter `AdminAPI` into the Name field then enter the following JSS code:

```
function groupadd(userName, groupName) {  
    var config = system.getConfig();  
    var user = config.users.get(userName);  
    var group = config.groups.get(groupName);  
    group.members.add(user);  
    config.applyChanges();  
}
```

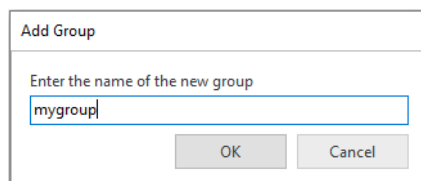
The process trigger should now look like this:



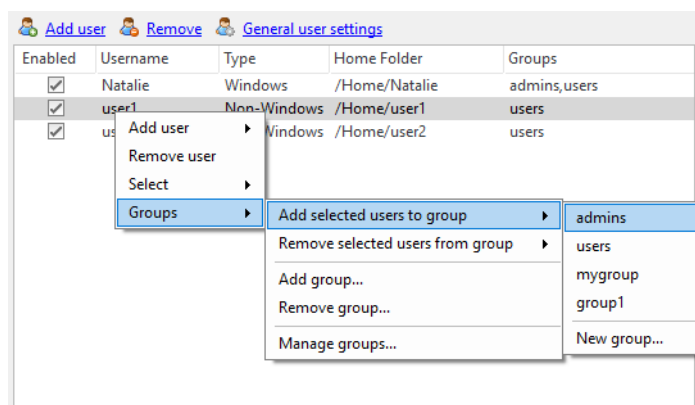
4. Select the Permissions tab.
5. Click Add user permission and select AdminAPI and your user.



6. Create a group into which the user will be added.



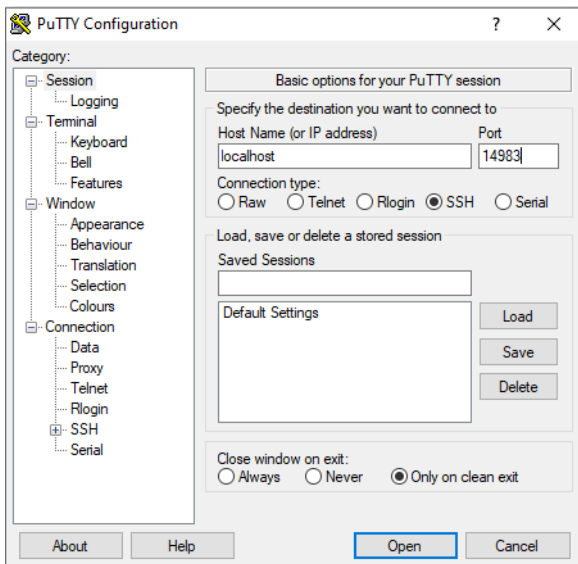
7. Add the login user to the `admins` group.



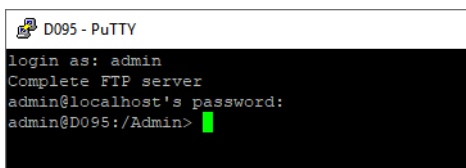
Now check if you can add the user to the group which you created above.

1. Open SSH.



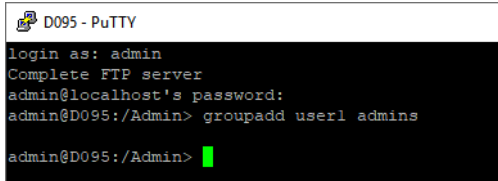


2. Login with SSH.



3. Add the user to the group with the command shown below.

```
groupadd myuser mygroup
```



4. Now check in the User panel to see if the user has been added to the group.

<a href="#">Add user</a> <a href="#">Remove</a> <a href="#">General user settings</a>				
Enabled	Username ▲	Type	Home Folder	Groups
<input checked="" type="checkbox"/>	user1	Non-Windows	/Home/user1	admins,users
<input checked="" type="checkbox"/>	user2	Non-Windows	/Home/user2	users
<input checked="" type="checkbox"/>	user3	Non-Windows	/Home/user3	users

You have now successfully added the user into the available group.

# Monitoring

The *Monitoring* panel in the CompleteFTP manager displays real-time logging; details about the currently connected users; the IP addresses that have been **auto-banned**; **server log files**; and performance statistics (in Professional and Enterprise MFT only).

## Real-time logging

Recent logging message from the server are shown in the real-time logging panel.

Server Messages

Pause

Display level: Debug

Filter

Clear

Copy to clipboard

Open in window

Time	Level	Category	IP Address	Session	User	Message
10:15:05,970	DEBUG	FTP	10.0.2.46	5	Mary	CMD> PROT P
10:15:05,970	DEBUG	FTP	10.0.2.46	5	Mary	setting protection level to Private
10:15:05,970	DEBUG	FTP	10.0.2.46	5	Mary	Processed command PROT P
10:15:05,970	DEBUG	FTP	10.0.2.46	5	Mary	CMD< 200 PROT P successful.
10:15:05,970	DEBUG	FTP	10.0.2.46	5	Mary	setting state to AwaitCommand
10:15:05,970	DEBUG	FTP	10.0.2.46	5	Mary	BeginSend: Response 34
10:15:05,970	DEBUG	FTP	10.0.2.46	5	Mary	OnSend: Response 34
10:15:05,971	DEBUG	FTP	10.0.2.46	5	Mary	CMD> PWD
10:15:05,971	DEBUG	FTP	10.0.2.46	5	Mary	Processed command PWD
10:15:05,971	DEBUG	FTP	10.0.2.46	5	Mary	CMD< 257 "/Home/Mary" is current directory.
10:15:05,971	DEBUG	FTP	10.0.2.46	5	Mary	setting state to AwaitCommand
10:15:05,971	DEBUG	FTP	10.0.2.46	5	Mary	BeginSend: Response 35
10:15:05,971	DEBUG	FTP	10.0.2.46	5	Mary	OnSend: Response 35

## Connections

In the *client connections* tab the current connections to the server are shown. The *Kill selected connection* option can be used to immediately disconnect a user from the server. The *admin* user cannot be killed.

Refresh ☒ Refresh automatically

Site	Username	Protocol	Remote Address	Local Port	Connected	Disk Usage
Default ...	javaftp	SFTP	127.0.0.1:64902	22	3/17/2016 9:42:17 AM	No quota
Admin	admin	SFTP	127.0.0.1:65204	14983	3/17/2016 10:14:47 AM	No quota
Admin	admin	SFTP	127.0.0.1:65209	14983	3/17/2016 10:15:12 AM	No quota

## Auto-bans

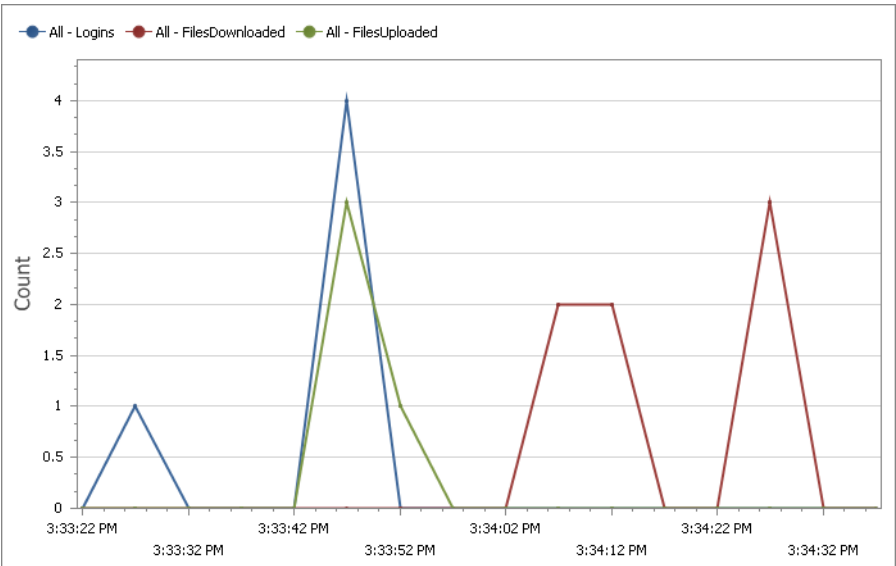
In the *auto-banned IP addresses* tab, all currently auto-banned IP addresses are shown. Selecting an IP address and clicking *Remove selected ban(s)* will unban an IP address. Choosing *Permanently ban select IP address(es)* will add a permanent filter banning the selected IP addresses. In Professional and Enterprise MFT, auto-banning can be configured as shown below and as described [here](#).

Auto-Banning

Number of failures allowed in period	10
Failure counting period (seconds)	300
Duration of ban (seconds)	3,600

## Performance

(Professional and Enterprise MFT only) A variety of real-time statistics can be charted, such as the number of connections, the number of file uploads and downloads, and the bytes uploaded and downloaded. These can be charted cumulatively and for each protocol.





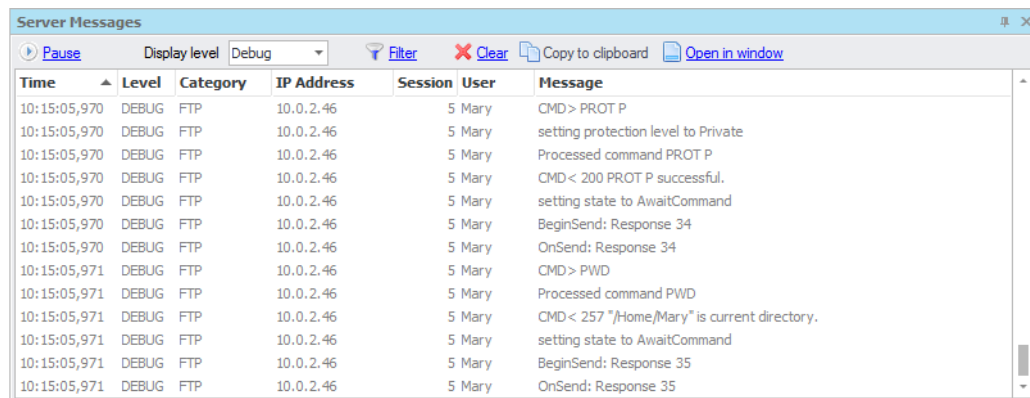
# Real-time Logging

The Real-time Logging panel shows server logging messages in real-time. This is particularly useful when testing a new configuration or trouble-shooting a problem. The panel is available in three places in CompleteFTP Manager:

1. As a standalone window that can be opened via the menu that's shown when the button in the bottom-left of the CompleteFTP Manager window is clicked.
2. As a dockable panel in the Overview tab.
3. As a tab in the Monitoring tab.

This description applies to all three instances as their behaviour is nearly identical.

Once the *Show* link is clicked, the Real-time Logging panel is ready to start reporting activity taking place on the server.



Time	Level	Category	IP Address	Session	User	Message
10:15:05,970	DEBUG	FTP	10.0.2.46	5	Mary	CMD> PROT P
10:15:05,970	DEBUG	FTP	10.0.2.46	5	Mary	setting protection level to Private
10:15:05,970	DEBUG	FTP	10.0.2.46	5	Mary	Processed command PROT P
10:15:05,970	DEBUG	FTP	10.0.2.46	5	Mary	CMD< 200 PROT P successful.
10:15:05,970	DEBUG	FTP	10.0.2.46	5	Mary	setting state to AwaitCommand
10:15:05,970	DEBUG	FTP	10.0.2.46	5	Mary	BeginSend: Response 34
10:15:05,970	DEBUG	FTP	10.0.2.46	5	Mary	OnSend: Response 34
10:15:05,971	DEBUG	FTP	10.0.2.46	5	Mary	CMD> PWD
10:15:05,971	DEBUG	FTP	10.0.2.46	5	Mary	Processed command PWD
10:15:05,971	DEBUG	FTP	10.0.2.46	5	Mary	CMD< 257 "/Home/Mary/" is current directory.
10:15:05,971	DEBUG	FTP	10.0.2.46	5	Mary	setting state to AwaitCommand
10:15:05,971	DEBUG	FTP	10.0.2.46	5	Mary	BeginSend: Response 35
10:15:05,971	DEBUG	FTP	10.0.2.46	5	Mary	OnSend: Response 35

Server messages will keep scrolling down the page until *Pause* is pressed.

The amount of detail shown in the server messages panel is controlled via the dropdown list labelled *Level*. Available levels are Error, Warning and Information. A higher level of detail called Debug becomes available when the server's logging level itself is set to Debug, which may be done in the Log Files panel.

Messages may be ordered by clicking on a column-heading. They may also be filtered by *Category*, *Session* or *User* by clicking either on the Filter link or on the small filter icon at the top right of these columns. When this is done a list of checkboxes shows up - one for each displayed value of the selected column. If no checkboxes are checked then no filtering will be done on that column. If one or more checkboxes are checked then only rows that have those values will be displayed.

The panel may be cleared at any time by clicking *Clear*. If one or more rows are selected then the *Copy to clipboard* link will be enabled. Clicking this will copy all of the selected rows to the clipboard.

# How to view the server log files

CompleteFTP has a sophisticated logging system based on [log4net](#).

Logs are extremely useful for diagnosing problems or simply for monitoring what is happening on the server. Logs can be viewed remotely via the CompleteFTP manager. The CompleteFTP manager also produces its own log file.

The Log Files tab in the Monitoring panel displays the server's log files. A log file may be viewed by selecting it in the top window that displays the logs.

Real-Time Logging

Log Files

Connections

Performance

Auto-bans

TCP Port Usage

Server logging level

Information

Configure auditing

Open logging folder

Save diagnostics

Name	Size	Date
Audit		6/22/2021 11:12 AM
Config.log	2,206 KB	6/22/2021 11:05 AM
Diagnostics.log	3 KB	6/22/2021 11:12 AM
Errors.log	3 KB	6/22/2021 11:12 AM

Log file: Diagnostics.log

Refreshed: 6/22/2021 11:12:44 AM

☐ Show admin logging

Refresh

Save to local disk

2021-06-22 11:12:12,329 INFO UserSession [Server.1] \_script\_ logged into Admin with protocol

2021-06-22 11:12:12,332 INFO Server [Server.1] Starting site: Admin

2021-06-22 11:12:12,332 INFO Site [Server.1] UpdateListeners() called for site Admin

2021-06-22 11:12:12,333 INFO Site [Server.1] FTP/FTPS not available on site Admin

2021-06-22 11:12:12,333 INFO Site [Server.1] FTPS-implicit not available on site Admin

2021-06-22 11:12:12,334 INFO Site [Server.1] HTTP not available on site Admin

2021-06-22 11:12:12,334 INFO ListenerManager [Server.1] Starting listener: SFTP:14983

2021-06-22 11:12:12,335 INFO ListenerManager [Server.1] Starting listener: HTTPS:14985

2021-06-22 11:12:12,335 INFO Site [Server.1] Site started Admin

2021-06-22 11:12:12,336 INFO Server [Server.1] Started sites: Default Site,Admin

2021-06-22 11:12:12,340 INFO Server [Server.1] Server running: Server 1 - Default Site, Admin

2021-06-22 11:12:12,342 WARN Console [Console.1] Q=quit, 1-9=toggle Server 1-9

The *Diagnostics.log* contains all available logging (depending on the level set). The *Errors.log* contains only logging at the ERROR level, and allows for quick inspection for potential issues.

As the log files are downloaded to the manager (as the manager may be running on a different machine), it is necessary to select the *Refresh* option on the right hand side of the view of the log file to see the latest logging information if some time has passed after initially viewing the file.

The *logging level* can be set in this window. By default, it is set to "Information". If problems occur, this should be set to "Debug" or "All" to obtain more details on what is going on in the server.

It is important to realise that setting the logging level to "Debug" or "All" can result in a large amount of logging, and can also have a significant impact on performance. These levels should only be used to diagnose problems, and the level should then be set back to "Information".

The *Save to local disk* option directly above the panel displaying the log file contents can be used to save a copy of the selected log file to disk. In addition, if CompleteFTP Manager is running on the same machine as CompleteFTP Server then the directory in which log files are written can be opened via the *Open logging folder* link.

## How to view the manager log files

The CompleteFTP manager also produces a log file, which may be useful to diagnose any connectivity problems or other issues.

The normal location of the manager's log file is *C:\Users\username\AppData\Local\EnterpriseDT\Complete FTP\Logs* where *username* is the name of the user account that is running the manager. Of course this log file is stored on the machine which is running the manager - which is not necessarily the machine running the CompleteFTP service.

# How to change the log file location

The log file locations can be changed using the following steps. The CompleteFTP manager should still be able to view the log files, after changing the location.

1. Open CompleteFTPService.exe.config, which is in the install directory. This may require administrative privileges. This is basically a log4net configuration file, and you can change the location of log files there.

2. Make a copy before modifying it!

3. Modify

```
%folder{{ '{' }}CommonApplicationData{{ '{' }}\\Enterprise Distributed Technologies\\Complete FTP\\logs\\Diagnostics.log
```

using similar paths to set your desired location, e.g.

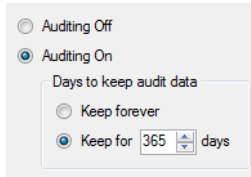
```
D:\\CompleteFTP\\logs\\Diagnostics.log
```

4. Restart the CompleteFTP service, in order to re-read the configuration file.

CompleteFTP should now be logging to your new log file location.

# How to enable auditing

CompleteFTP supports the use of an audit log. The audit log records all important file transfer operations as well as login/logout.

A screenshot of a software configuration window for auditing. It features two radio buttons at the top: 'Auditing Off' and 'Auditing On'. The 'Auditing On' option is selected. Below these is a section titled 'Days to keep audit data' which contains two more radio buttons: 'Keep forever' and 'Keep for [365] days'. The 'Keep for [365] days' option is selected. The number '365' is inside a text box with up and down arrow buttons on its right side. The word 'days' follows the text box.

To enable the audit log, go to the *Log Files tab* found on the *Monitoring* tab in the manager, and select the "Configure Auditing" link. Select the "Auditing on" radio button.

Once enabled, CompleteFTP will record every operation in an auditing log file. Log files are 'rolled' by day such that each file contains all the operations for a given day. The file 'Audit.log' contains today's operations. Older files are named using the convention 'Audit.log.YYYYMMDD'. The number of days stored may be changed in this panel (or select "Keep forever" to keep all log files).



# How to modify the logging configuration

CompleteFTP's sophisticated logging system is based on **log4net**. log4net is highly configurable. If modifications are to be made to the log4net configuration then the full log4net configuration should be placed into a file called `LogConfig.xml`, which should be placed in the directory, `C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Logs`. This will prevent changes from being overwritten when CompleteFTP is upgraded. The default log4net configuration may be found in `C:\Program Files\Complete FTP\Server\CompleteFTPService.exe.config`. The XML node, `<log4net>`, and all of its sub-nodes, should be copied into the `LogConfig.xml` file and used as a basis for a new configuration.

## How to log to syslog

It is sometimes a requirement to direct logging output to *syslog*. This is normally a daemon that listens on UDP port 514. This can be done via the following steps.

1. Create the `LogConfig.xml` file as described above
2. Add a new appender section:

```
<appender name="Syslog" type="log4net.Appender.RemoteSyslogAppender">
  <param name="RemoteAddress" value="127.0.0.2" />
  <layout type="log4net.Layout.PatternLayout" value="%-5p %type: %m%n"/>
</appender>
```

and add an appender reference to the list of existing appenders:

```
<root>
  <level value="ALL" />
  <appender-ref ref="Recent Logging Messages" />
  <appender-ref ref="Recent Error Messages" />
  <b>appender-ref ref="Syslog" />
  <appender-ref ref="Audit" />
  <appender-ref ref="Config" />
</root>
```

3. Restart the CompleteFTP service to reread the configuration file.

CompleteFTP should now be logging to syslog.

# How to produce clean log files

The best way to troubleshoot problems, is to send a 'clean debug log' to [support](#). This is a log file that contains, as far as possible, only information related to the problem at hand. Note that from version 8.0, the [server diagnostics file](#) is preferred, as it contains the configuration file as well as logging.

The summary steps for producing a 'clean debug log' are as follows:

1. Set the logging level to Debug (or All, for even more detail)
2. Stop the CompleteFTP service
3. Delete the existing log file
4. Start the CompleteFTP service again
5. Reproduce the error
6. Send [support](#) the (zipped-up) log file

This, of course, assumes that the service can be stopped without impacting users. If the service cannot be stopped, perform steps 1, 5 and 6 only.

The detailed steps:

1. **Set the logging level to Debug:**
  - a. Open CompleteFTP Manager and connect.
  - b. Select the Monitoring tab and then select Log Files.
  - c. Set the Logging level to 'Debug' and click 'Apply Changes'.
  - d. Close the CompleteFTP Manager.
2. **Stop the CompleteFTP service:**
  - a. Open the Services item in the Windows Control Panel.
  - b. Select CompleteFTP and hit the stop button.
3. **Delete the existing log file:**
  - a. Select 'Log Files' from the CompleteFTP menu group in the Windows start menu.
  - b. Find 'Diagnostics.log' and delete (or rename) it.
4. **Start the CompleteFTP service again:**
  - a. Open the Services item in the Windows Control Panel.
  - b. Select CompleteFTP and hit the start button.
5. **Reproduce the error:**
  - a. Do whatever it was that caused the error: E.g. connecting with an SFTP client and getting an error.
6. **Send the (zipped) log file to EDT support:**
  - a. Select 'Log Files' from the CompleteFTP start menu (in Windows).
  - b. Zip up the Diagnostics.log file and e-mail it to [support](#)

# How to get diagnostics for support

To solve technical issues, diagnostic log files that illustrate the issue should be zipped up and sent to [support](#). Please note that a current support agreement is required for technical support. If your support agreement has expired, please [contact us](#) to obtain a renewal quote.

There are three different scenarios listed below. Please send us the files listed for your scenario.

1. Server issues. These are the most common issues, and require a debug level server log file. See [here](#). If you are having difficulties with a client connecting, please also obtain a client log file at the highest possible debug level.

2. Manager issues. This is for when difficulties are experienced getting the CompleteFTP manager to connect to the server.

In the connect dialog box, choose "Other settings ..." and set the logging level to "Full diagnostic". Then try connecting to the server.

The log file should appear here:

```
C:\Users\<username>\AppData\Local\EnterpriseDT\Complete FTP\Logs\Diagnostics.log
```

It may also be helpful to provide a [server log file](#).

3. Installation and migration issues. This is when difficulties are experienced installing or migrating CompleteFTP from one machine to another.

Please send us C:\Program Files\Complete FTP\install.log and C:\Program Files\Complete FTP\Server\bootstrapper.log.

# Getting server diagnostics

Server diagnostic log files can be found in CompleteFTP's configuration directory, usually here: C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Logs. The most important log is `diagnostics.log`, but it also contains the error log and the audit log.

You can also download a diagnostic log directly from the CompleteFTP manager, as described below.

1. Open the CompleteFTP manager and go to the Log Files tab in the Monitoring panel.

Real-Time Logging | **Log Files** | Connections | Performance | Auto-bans | TCP Port Usage

Server logging level: **Information** | [Configure Auditing](#) | [Save Diagnostics to local disk](#)

Name	Size	Date
Audit		02/08/2017 2:28 PM
Config.log	54 KB	02/08/2017 12:20 PM
Diagnostics.log	42 KB	02/08/2017 2:28 PM
Errors.log	8 KB	02/08/2017 1:35 PM

Log file: Diagnostics.log Refreshed: 02/08/2017 2:28:15 PM ☐ Show admin logging [Refresh](#) [Save to local disk](#)

```
2017-02-08 14:27:46,084 INFO UserSession [Session.10:Default Site:u1:127.0.0.1] u1 logged out of Default Site with pr
2017-02-08 14:27:46,084 INFO SFTPConnection [Session.10:Default Site:u1:127.0.0.1] Connection closed
2017-02-08 14:28:00,145 INFO UserSession [Session.12:Default Site:u1:127.0.0.1] u1 logged into Default Site with pro
2017-02-08 14:28:10,202 INFO UserSession [Session.13:Default Site:u1:127.0.0.1] u1 logged into Default Site with pro
2017-02-08 14:28:10,230 INFO StoreCommand [Session.13:Default Site:u1:127.0.0.1] Opened file /Home/u1/file0001.dat fr
2017-02-08 14:28:10,351 INFO UserSession [Session.14:Default Site:u1:127.0.0.1] u1 logged into Default Site with pro
2017-02-08 14:28:10,357 INFO StoreCommand [Session.14:Default Site:u1:127.0.0.1] Opened file /Home/u1/file0002.dat fr
2017-02-08 14:28:10,506 INFO StoreCommand [Session.13:Default Site:u1:127.0.0.1] Closed file /Home/u1/file0001.dat a
2017-02-08 14:28:10,507 INFO StoreCommand [Session.14:Default Site:u1:127.0.0.1] Closed file /Home/u1/file0002.dat a
2017-02-08 14:28:10,544 INFO StoreCommand [Session.13:Default Site:u1:127.0.0.1] Opened file /Home/u1/file0003.dat fr
2017-02-08 14:28:10,564 INFO StoreCommand [Session.14:Default Site:u1:127.0.0.1] Opened file /Home/u1/file0004.dat fr
2017-02-08 14:28:10,573 INFO StoreCommand [Session.13:Default Site:u1:127.0.0.1] Closed file /Home/u1/file0003.dat a
2017-02-08 14:28:10,574 INFO StoreCommand [Session.14:Default Site:u1:127.0.0.1] Closed file /Home/u1/file0004.dat a
2017-02-08 14:28:10,620 INFO StoreCommand [Session.14:Default Site:u1:127.0.0.1] Opened file /Home/u1/file0005.dat fr
2017-02-08 14:28:10,622 INFO StoreCommand [Session.13:Default Site:u1:127.0.0.1] Opened file /Home/u1/file0006.dat fr
2017-02-08 14:28:10,631 INFO StoreCommand [Session.14:Default Site:u1:127.0.0.1] Closed file /Home/u1/file0005.dat a
2017-02-08 14:28:10,633 INFO StoreCommand [Session.13:Default Site:u1:127.0.0.1] Closed file /Home/u1/file0006.dat a
2017-02-08 14:28:10,651 INFO StoreCommand [Session.14:Default Site:u1:127.0.0.1] Opened file /Home/u1/file0007.dat fr
2017-02-08 14:28:10,657 INFO StoreCommand [Session.14:Default Site:u1:127.0.0.1] Closed file /Home/u1/file0007.dat a
2017-02-08 14:28:10,664 INFO StoreCommand [Session.13:Default Site:u1:127.0.0.1] Opened file /Home/u1/file0008.dat fr
2017-02-08 14:28:10,689 INFO StoreCommand [Session.14:Default Site:u1:127.0.0.1] Opened file /Home/u1/file0009.dat fr
2017-02-08 14:28:10,691 INFO StoreCommand [Session.13:Default Site:u1:127.0.0.1] Closed file /Home/u1/file0008.dat a
2017-02-08 14:28:10,694 INFO StoreCommand [Session.14:Default Site:u1:127.0.0.1] Closed file /Home/u1/file0009.dat a
2017-02-08 14:28:10,714 INFO StoreCommand [Session.14:Default Site:u1:127.0.0.1] Opened file /Home/u1/file0010.dat fr
2017-02-08 14:28:10,722 INFO StoreCommand [Session.14:Default Site:u1:127.0.0.1] Closed file /Home/u1/file0010.dat a
```

2. Ensure that the logging level is set to Debug (or All).

Real-Time Logging | **Log Files** | Connections | Performance | Auto-bans | TCP Port Usage

Server logging level: **Debug** | [Configure Auditing](#) | [Save Diagnostics to local disk](#)

Name	Size	Date
Audit		02/08/2017 2:28 PM
Config.log	54 KB	02/08/2017 12:20 PM
Diagnostics.log	42 KB	02/08/2017 2:28 PM
Errors.log	8 KB	02/08/2017 1:35 PM

3. Use "Apply changes" to save the changed settings.
4. Attempt to reproduce the problem.
5. Use the "Save Diagnostics to local disk" link to download a zip file containing a sanitised version of your configuration file for analysis by our support team (stripped of password information and server keys) and your `diagnostics.log` file.

One drawback of having the manager running while trying to reproduce a problem is that the server log can get cluttered with logging from the manager. It's often more helpful to close the manager after setting the logging level to debug, reproduce the issue, and then directly copy the log files from the location specified above.

Real-Time Logging | **Log Files** | Connections | Performance | Auto-bans | TCP Port Usage

Server logging level: **Debug** | [Configure Auditing](#) | [Save Diagnostics to local disk](#)

Name	Size	Date
Audit		02/08/2017 2:39 PM
Config.log	55 KB	02/08/2017 2:32 PM
Diagnostics.log	565 KB	02/08/2017 2:39 PM
Errors.log	8 KB	02/08/2017 1:35 PM

Log file: Diagnostics.log Refreshed: 02/08/2017 2:39:22 PM ☐ Show admin logging [Refresh](#) [Save to local disk](#)

# Reporting

CompleteFTP Professional and Enterprise MFT Editions offer a robust reporting feature that allows users to gain insights into the events that have occurred on the server. These report can help administrators monitor server activity, track user behavior, and ensure optimal performance. Data is presented in a structured manner, enabling users to view, filter, and analyze event information within a specified date range. For those using the Enterprise MFT Edition, there's added flexibility to create custom reports tailored to specific needs.

Report: 

Number of logins by date

Date range: Full range

 2/10/2023 - 12/01/2024 [Refresh](#)

...

			Event <div>▲</div>	
			Login	
<div>▼ Year ▲</div>	<div>▼ Month ▲</div>	<div>► Day ▲</div>	Successes	Failures
▼ 2023	▼ October	2	2	0
		5	2	0
		6	1	0
		8	1	0
		9	1	0
		11	2	0
		13	2	0
		24	2	0
		25	1	0
		26	1	0
		28	1	0
		30	1	0
	October Total		17	0
	▼ November	1	2	0
		6	1	0
		8	1	0
		11	1	0
		15	1	0
		20	1	0
		22	1	0
		28	2	0
	November Total		10	0
2023 Total			27	0
▼ 2024	▼ January	12	2	0
Grand Total			29	0

To get started with reporting, select the Reports item from the side menu in CompleteFTP Manager.

## Enabling Reporting

Before diving into the details of using the reporting feature, it's essential to ensure that reporting is enabled. If you access the reporting panel and find that all controls are disabled, it means that reporting is not active.

In the center of the reporting panel, you'll see a button labeled 'Enable reporting'. Clicking this button will prompt CompleteFTP to check for available audit files. If any are found, you'll be presented with statistics about the auditing data, such as the number of files, the date range of the data, and an estimated count of events. You'll then have the option to import this data. Note that if there's a significant amount of data, the import process might take a few minutes. Once imported, the reporting feature will be activated, and you can proceed to view or customize reports.

Reporting lets you analyze historical activity in CompleteFTP. When enabled, activity is recorded in a database, accessible whenever you need. If you've already enabled auditing, you can jump-start the database by importing that data. Click the button below to get started.

Enable Reporting

## Viewing Reports

Upon enabling the reporting feature, you can immediately start viewing the available reports. Here's how:

1. **Select a Report:** Use the dropdown list at the top of the reporting panel to choose a report of interest.

Report: Number of logins by date Date range: Full range

Successes

▼ Year ▲

▼ 2024 ▲

Number of logins by date

Number of logins by user

Last login by user

Number of logins by site

Login failures by username

Bytes transferred by user

File transfers by user

Data transfer by protocol

Events by type

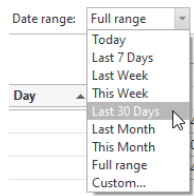
Failures

Success vs failure events by month

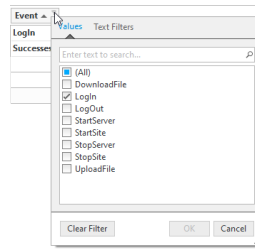
Most accessed files

Most active users

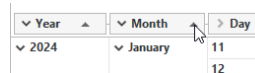
2. **Adjust Date Range:** Next to the report dropdown, you'll see the date range control. Click on it to adjust the date range as per your requirements. You can either specify a custom range or select from predefined options.



3. **Interact with the Report:** The main report display allows you to:
- Filter data by hovering over a field button and clicking the filter icon.



- Sort data by clicking on the up/down arrows column headers.



- Resize columns by dragging or double-clicking their borders.

Successes		Failures		Event	
				Login	
				Successes	Failures
Year	Month	Day			
2024	January	11		4	0
		12		10	4
		January Total		14	4

## Understanding Report Fields

CompleteFTP reports contain combinations of the following fields:

- Site:** Represents the name of the site where the event occurred.
- IP Address:** Denotes the IP address of the remote client.
- User-name:** The account being accessed.
- Path:** The file path being accessed.
- Protocol:** The protocol in use, such as FTPS, SFTP, or HTTPS.
- Bytes:** The number of bytes transferred during the event.
- Most Recent:** The latest time the event took place.
- Time Fields:** These include Year, Month, Day, Hour, and Minute, representing the exact time of the event. These fields are interrelated and can be grouped for a hierarchical view.
- Event:** The name or type of event.
- Successes:** A count of successful events.
- Failures:** A count of failed events.

## Custom Reports

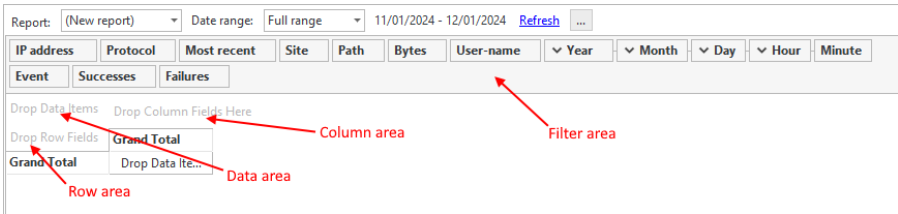
CompleteFTP Enterprise MFT allows you to design your own reports - see [Designing Custom Reports](#).

# Designing Custom Reports (Enterprise MFT Edition Only)

The **Reporting** topic described how reports are enabled and view. For users of the Enterprise MFT Edition, the ability to create custom reports offers a personalized view of event data. Designing custom reports allows administrators to tailor the data presentation to specific needs, ensuring that the most relevant information is always at their fingertips. Whether you're looking to track specific user activities, monitor protocol usage, or analyze data transfers, custom reports provide the flexibility to focus on what matters most.

## Understanding the Reporting Structure

At the heart of the reporting feature is a pivot grid. A pivot grid is a data summarization tool used in spreadsheets and business intelligence programs. It allows users to reorganize and summarize selected columns and rows of data to obtain a desired report. In the context of CompleteFTP's reporting feature, the pivot grid lets you view event data from different perspectives by "pivoting" data fields.



Here's a breakdown of the main components of a pivot grid:

- Row Area:** Fields placed here will be displayed as rows in the report. For example, if you want to see data categorized by **Protocol**, dragging the **Protocol** field to the row area will list all protocols as individual rows.
- Column Area:** Fields here will be displayed as columns. This is useful for breaking down data within each row category. For instance, if **Protocol** is in the row area and **Event** is in the column area, you'll see a breakdown of events for each protocol.
- Data Area:** This is where you define what data you want to analyze. If you drag the **Bytes** field here, the grid will show the total bytes transferred for each combination of row and column fields.
- Filter Area:** Fields here allow you to filter the data that's displayed in the grid. For example, if you only want to see data for a specific user, you can drag the **User-name** field to the filter area and select the desired user.

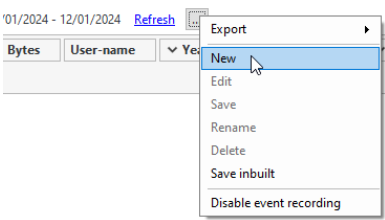
The beauty of a pivot grid is its flexibility. By simply dragging and dropping fields between these areas, you can quickly change the perspective of your data. Want to see data transfers by user instead of by protocol? Just drag the **User-name** field to the row area. It's that simple.

Now that you have a grasp of the pivot grid concept, let's dive into designing custom reports in CompleteFTP.

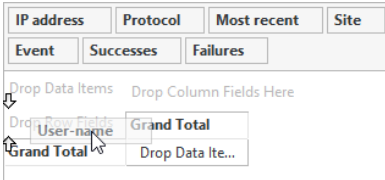
## Creating Custom Reports

Here's the general process for generating a new custom report:

- Start a New Report:** Click on the ellipsis button (⋮) and select 'New'.



- Drag and Drop Fields:** From the filter area, drag fields of interest into the row, column, or data areas. For instance, if you want to view the number of bytes transferred by each protocol, drag **Protocol** to the row area and **Bytes** to the data area.



- Apply Filters:** Hover over a field button to reveal a filter icon. Clicking this allows you to filter data based on specific criteria.
- Group Time Fields:** If you're interested in viewing events by a specific time, drag the **Year** field to the row area. The other time-fields will follow. You can then expand or collapse those fields to control the level of granularity you need.
- Save the Report:** Once satisfied with the design, click on the ellipsis button (⋮) and select 'Save'. Provide a name for your custom report.

## Managing Reports

The CompleteFTP reporting feature provides an intuitive interface to manage your custom reports. Whether you're creating a new report, editing an existing one, or organizing your report list, the management tools are just a click away.

To access the report management options:

- Click on the ellipsis button (⋮) located at the top-right corner of the reporting panel.

2. A dropdown menu will appear, presenting you with the following options:

- **New:** Start creating a new custom report from scratch.
- **Edit:** Modify the currently selected report. This allows you to change the fields, apply new filters, or adjust the layout of the report.
- **Save:** Once you've made changes to a report, use this option to save those changes. Remember, any unsaved changes will be lost if you navigate away or close the application.
- **Rename:** Give your report a new name. This is especially useful if the current name doesn't accurately reflect the report's content or purpose.
- **Delete:** Remove the selected report. Be cautious with this option as deleted reports cannot be recovered.

These management tools ensure that you have full control over your reports, allowing you to quickly adapt to changing requirements or data analysis needs.

Understanding Available Fields

When designing custom reports, it's essential to understand the available fields, their significance, and how they can be utilized to present the data you need. Each field represents a specific type of data or event attribute, and knowing how to use them effectively will allow you to create insightful reports.

Field Label	Description	Possible Areas
Site	Name of the site where the event occurred.	Row, Column, Filter
IP Address	IP address of the remote client.	Row, Column, Filter
User-name	Name of the account being accessed.	Row, Column, Filter
Path	Path of the file being accessed.	Row, Column, Filter
Protocol	Protocol in use (e.g., FTPS, SFTP, HTTPS).	Row, Column, Filter
Bytes	Number of bytes transferred during the event.	Data, Filter
Most Recent	The most recent time the event occurred.	Data, Filter
Year	Year the event occurred. Can be grouped with Month, Day, Hour, and Minute for a detailed time breakdown.	Row, Column, Filter
Event	Name or type of the event.	Row, Column, Filter
Successes	Count of successful events.	Data, Filter
Failures	Count of failed events.	Data, Filter

Some fields, like Year, Month, Day, Hour, and Minute, are interrelated and can be grouped for a hierarchical view. For instance, if you drag the Year field to the row area, you can expand it to view data by month, day, hour, and minute, providing a detailed time breakdown.

By understanding and effectively using these fields, you can tailor your reports to focus on the specific data and insights you're interested in.

Tips for Designing Reports

Going from a reporting requirement to configuring a pivot grid can be done using the following approach:

1. **Understand reporting requirement:** Before you can configure the pivot grid, you need to have a clear understanding of the requirement. What specific data is needed? What insights or patterns are you trying to uncover?
2. **Identify relevant fields:** Based on the requirement, list down the fields that will be relevant.
3. **Determine layout:** Decide how you want to present the data. Which fields should be rows, which should be columns, and which should be data values?
4. **Configure pivot grid:** Now, you can start configuring the pivot grid by dragging and dropping the identified fields into the appropriate areas.
5. **Grouping:** If your data has inherent hierarchies, decide how you want to group them. In the pivot grid, you can drag multiple fields into one area. Their order will determine the grouping hierarchy.
6. **Iterate:** The initial configuration might not provide the exact view or insight you were aiming for. The report will be updated in realtime, making it easy to adjust and refine the pivot grid setup until it aligns perfectly with the requirement.
7. **Save the report:** Once satisfied, save the report, giving it a useful name.

Examples

Below is a set of examples to illustrate how specific reports are designed according to requirements.

Example 1: Logins by user

If you're keen on understanding which users are most active on your server, tracking the number of successful logins can be insightful. This report will show you a breakdown of successful logins for each user over a specified period. Here's how to set it up:

- Adjust the date range as needed using the date range control, for instance, 'last 30 days' if you want a monthly overview.
- Start a new report.
- Drag User-name to the row area to list all users.
- Drag Event to the column area.
- Hover the mouse over the Event button and click on the little filter icon that appears in the top-right corner, then select Login.



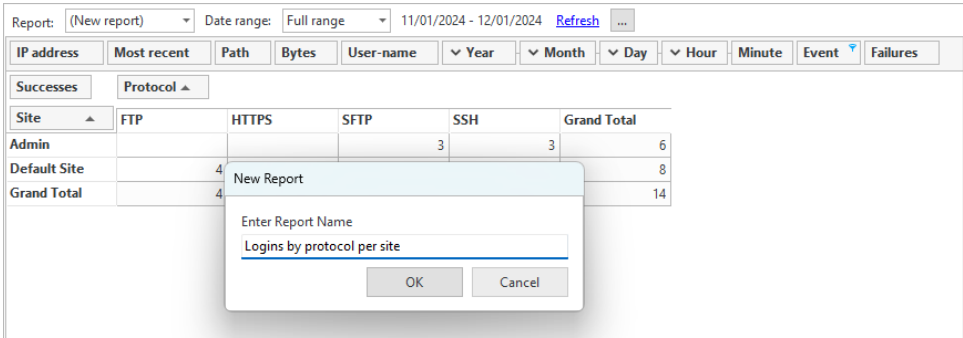
- Drag **Successes** to the data area to count the number of successful logins for each user.
- Save the report with a descriptive name, such as "Logins by user".

With this report, you'll have a clear overview of which users have been most active in terms of logging into the server, helping you identify frequent users or spot any unusual login patterns.

Example 2: Logins by protocol per site

Imagine you want to analyze the frequency of different protocols used across various sites over the past week. You aim to understand which protocols are most popular on specific sites. Here's how you can design this report:

- Adjust the date range as needed using the date range control, for instance, 'last 7 days' if you want a weekly overview.
- Start a new report.
- Drag **Site** to the row area to list all the sites.
- Drag **Protocol** to the column area. This will create columns for each protocol like FTPS, SFTP, HTTPS, etc.
- Drag **Event** to the column area next to the **Protocol** field.
- Hover the mouse over the Event button and click on the little filter icon that appears in the top-right corner, then select **Login**.
- Drag **Successes** to the data area to count the number of successful logins for each user.
- Save the report with a descriptive name, such as "Logins by protocol per site".



By following these steps, you'll have a report that displays a matrix of sites against protocols, allowing you to quickly identify which protocols are most frequently used on each site over the past week.

Example 3: Osman's data transfers

Suppose you're interested in monitoring the data usage of a particularly busy user with user-name, 'osman'. Here's how you can set up this report:

- Start a new report.
- Drag **User-name** to the column area.
- Hover the mouse over the **User-name** button and click on the little filter icon that appears in the top-right corner, then select 'osman'.
- Drag **Year** to the row area. When you do this all the other time fields will follow.
- Make sure that **Year** and **Month** are expanded, but that **Day** is not expanded (unless you want an hourly breakdown)
- Drag **Bytes** to the data area to sum up the amount of data transferred on each day.
- Save the report with a meaningful name, such as "Osman's data transfers".

Report: Osman's data transfers		Date range: Full range	2/10/2023 - 12/01/2024
Bytes		User-name	
Year		Month	
2023		osman	
		October	
		December	
2023 Total		10,907,341	
2024		January	
Grand Total		19,755,735	

With this report, you'll get a clear picture of Osman's data transfers. This can be particularly useful for security audits or understanding user behavior.

# Performance

We are often asked how well CompleteFTP will perform with x users. The short answer is, it depends.

Client applications vary enormously in what they do. Some applications are batch jobs that transfer a single file once per day, while others are transferring every minute. Some applications transfer tiny files, while others transfer large files that can take many minutes, or even hours. The primary constraint on performance will be how many simultaneous transfers are being made at any given time.

CompleteFTP has a low memory footprint so memory should rarely be an issue. For simultaneous transfers, the limiting factors are going to be CPU and bandwidth. CPU is particularly important for encrypted transfers (i.e. SFTP, FTPS and SCP), as encryption is computationally expensive. If CompleteFTP is being run in a VM, we recommend allocating *at least 2* vCPUs to the VM. If many clients are transferring simultaneously, obviously bandwidth is also going to limit transfer speeds.

If performance is mission critical, it is advisable to simulate the production scenario to ensure that CPU and bandwidth is adequate. It is quite easy to do a basic simulation of many clients - run multiple instances of a test client that continually transfers files back and forth. You may wish to run a few instances on different machines. That should provide a reasonable idea of how well the system is going to perform.

# Administrator Access

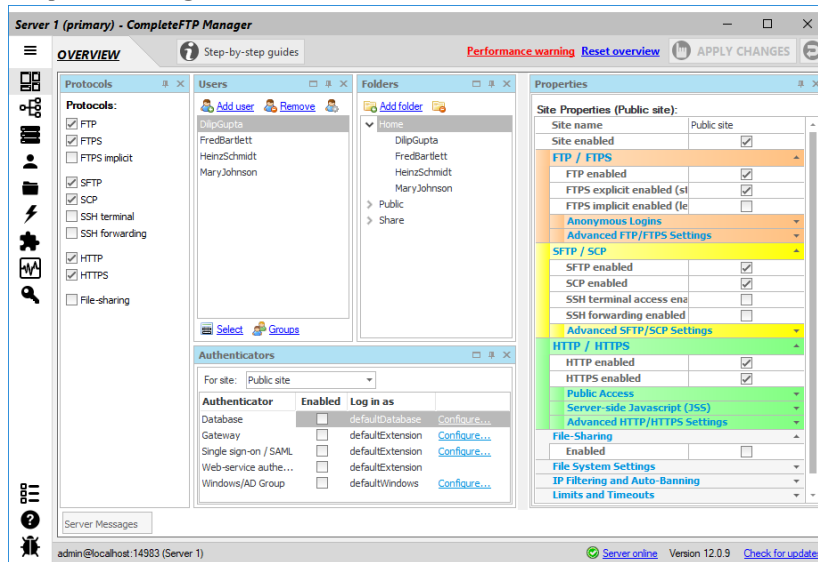
CompleteFTP offers a suite of options for administration/configuration:

- **CompleteFTP Manager** - Windows application offering full control over the server configuration [All editions].
- **CompleteFTP WebAdmin** - mobile-friendly (responsive) web-app useful for user management and monitoring [Professional Edition or better].
- **Command-Line Interface (CLI)** - automation of configuration tasks [Enterprise MFT only].
- **Configuration Object Model** - object model exposing (almost) all aspects of the server configuration accessible through **Javascript (JSS)** [Enterprise MFT only].

## CompleteFTP Manager

The *CompleteFTP Manager* Windows application provides the most flexibility, and this guide is primarily concerned with describing its capabilities. The manager connects to the administrator site hosted on the server, and the administrator site settings can be changed as described at [this link](#).

### CompleteFTP Manager:



# How to change the administrator site settings

The *Admin* panel (shown below) displays the settings for the Administrator site in the *Standard Edition* and *Professional Edition*. The administrator user name can be set in this panel, and the administrator password can also be changed.

**ADMIN**

Administrator username  
admin

[Change administrator password](#)

Administrator email address

Administrator full name  
CompleteFTP Administrator

[Show advanced administrator connection settings](#)

More detailed administrator settings can be modified by selecting *Show advanced administrator connection settings*. These settings apply only to the Admin site and are described in the [Site/Sites](#) section. For example, the protocol used to connect to the Admin site can be change.

In *Enterprise MFT*, the Admin site is displayed in *Sites* by selecting 'Show system users/folders/sites' in the main form's Options menu (shown below).

Site	Site label	CompleteFTP
Admin	FTP / FTPS	
Default Site		
	FTP enabled	<input type="checkbox"/>
	FTPS explicit enabled (standard FTPS)	<input type="checkbox"/>
	FTPS implicit enabled (legacy FTPS)	<input type="checkbox"/>
	Anonymous Logins	
	Advanced FTP/FTPS Settings	
	Port for FTP / FTPS explicit	14982
	Port to use for FTPS implicit	990
	Listening IP addresses	All available
	Folder listing format	Unix
	Character set	Default - Unicode (UTF-8)
	Passive (PASV) Transfer Settings	
	Security Settings	
	SFTP / SCP	
	SFTP enabled	<input checked="" type="checkbox"/>
	SCP enabled	<input checked="" type="checkbox"/>
	SSH terminal access enabled	<input type="checkbox"/>
	SSH forwarding enabled	<input type="checkbox"/>
	Advanced SFTP/SCP Settings	
	HTTP / HTTPS	
	HTTP enabled	<input type="checkbox"/>
	HTTPS enabled	<input checked="" type="checkbox"/>
	Advanced HTTP/HTTPS Settings	
	Server-side Javascript (JSS)	
	Public Access	
	File System Settings	
	IP Filtering and Auto-Banning	
	Limits and Timeouts	
	Messages	

Change skin

Show logging window

☒ Show context help

Show guide window

☒ Show system users/folders/sites

Import Filezilla Configuration

S settings

the HTTP and HTTPS protocols.

Server online

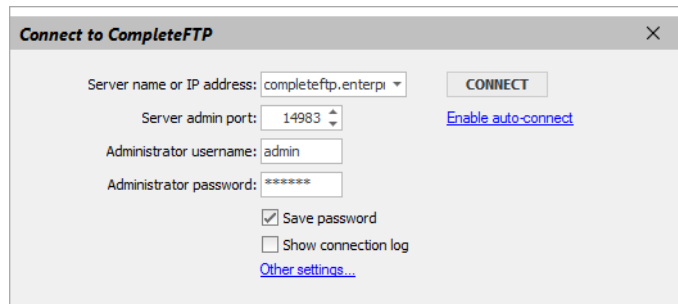
The Admin site is used to administer the server instance. The administrator connects to the Admin site via the Complete FTP Manager. By default the Admin site listens on port 14982 for FTP/FTPS, and port 14983 for SSH/SFTP. From version 6.1, the default protocol is set to SFTP, as it is a more secure protocol than FTP.

Note that the SSH Terminal option must be enabled for the Admin site if SSH access is required for command-line administration as described at [this link](#).

# How to manage the server remotely

The production installer allows the CompleteFTP manager application to be installed separately on any machine, and no license is required. The only requirement is that the server is reachable from the manager machine.

Once installed, the hostname of the server must be supplied (instead of the default *localhost*). The default port is 14983, and SFTP is used for the communication protocol. An example is shown below:



The screenshot shows a dialog box titled "Connect to CompleteFTP" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Server name or IP address:** A dropdown menu showing "completeftp.entervi".
- CONNECT:** A button to the right of the server name field.
- Server admin port:** A spinner box showing "14983".
- Enable auto-connect:** A blue hyperlink to the right of the port field.
- Administrator username:** A text box containing "admin".
- Administrator password:** A text box containing "\*\*\*\*\*".
- Save password:** A checked checkbox.
- Show connection log:** An unchecked checkbox.
- Other settings...:** A blue hyperlink at the bottom.

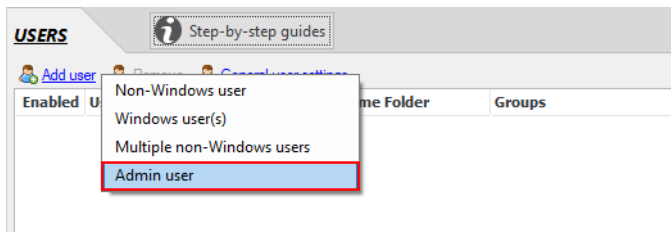
There are two configuration changes you may need to make:

1. If the manager machine is outside the local network, you may need to add firewall rules that permit it to connect to the server via SFTP on port 14983.
2. By default the "Admin" site on the server only permits connections from *localhost*. This is for security reasons. You will need to set an **IP filtering rule** for the "Admin" site that allows external IP addresses to connect to CompleteFTP. See [here](#) for how to change the Admin site settings.

# How to add extra administrator users (requires Enterprise MFT)

Often, users want to add extra CompleteFTP administrators, so that they have several accounts with administrator access. This is only possible in Enterprise MFT.

It is very straightforward to add local admin users, i.e. CompleteFTP users that have administrator privileges. Use the *Add user* link in the *Users* tab of the CompleteFTP manager, and select the *Admin User* menu item.



If you want an existing Windows user to be set as an administrator, it is more complicated. The steps are shown below:

1. Add the Windows user to CompleteFTP
2. Set the home directory of the user on the Admin site to /Admin (see note below)
3. Add the user to the *admins* group

By default, the Admin site is not shown the dialog that allows you to set the home directory of a user. You'll need to select "Show system users/folders/sites" in the Options menu at the bottom left of the window in order to make it appear there.

# How to do command-line administration

## (requires Enterprise MFT)

In Enterprise MFT, some administration tasks can be performed using the SSH command-line. The Admin site by default listens on port 14983 for SSH/SFTP. Note that the *SSH/Terminal* option for the Admin site must be enabled for SSH clients to connect (note that Admin site is displayed in Sites panel by selecting "Show system users/folders/sites" in the Options menu at the bottom left of the window).

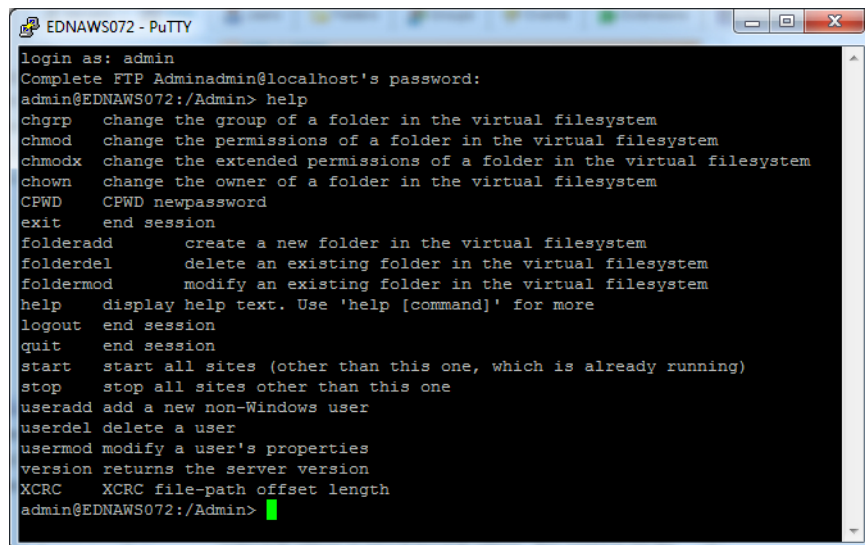
Any command-line SSH client can connect to this port (14983) using the admin user's credentials and use the available commands. On Windows, [PuTTY](#) is a good free option for an SSH client. On Unix systems, command-line SSH clients are usually freely available and installed by default.

Once the terminal window is shown after connecting, type 'help' to see [what commands are currently available](#). Type 'help command' to find out more about each command.

There are commands for adding and deleting users, adding and removing folders, controlling permissions, and for encrypting and decrypting files on the server, all detailed [here](#).

Multiple commands can also be placed into a script and run on the server as described [here](#).

An example of using PuTTY to access the Admin site is shown below:



```
EDNAWS072 - PuTTY
login as: admin
Complete FTP Adminadmin@localhost's password:
admin@EDNAWS072:/Admin> help
chgrp  change the group of a folder in the virtual filesystem
chmod  change the permissions of a folder in the virtual filesystem
chmodx change the extended permissions of a folder in the virtual filesystem
chown  change the owner of a folder in the virtual filesystem
CPWD   CPWD newpassword
exit   end session
folderadd  create a new folder in the virtual filesystem
folderdel  delete an existing folder in the virtual filesystem
foldermod  modify an existing folder in the virtual filesystem
help       display help text. Use 'help [command]' for more
logout     end session
quit       end session
start      start all sites (other than this one, which is already running)
stop       stop all sites other than this one
useradd    add a new non-Windows user
userdel    delete a user
usermod    modify a user's properties
version    returns the server version
XCRC      XCRC file-path offset length
admin@EDNAWS072:/Admin>
```

# Administration with UNIX-like scripts (requires Enterprise MFT)

In Enterprise MFT, **administrative commands** can be scripted, i.e. they can be saved to a file and the contents of the file can be executed as a script on the administrative site. For example, this could be used to add many users at once. This would be difficult to do using the CompleteFTP manager, and still tedious via the command-line. Instead, the script could be generated and then executed in one command.

Like command-line administration, scripts must be run with an SSH client that can run scripts remotely. On Windows, a popular free tool is **plink**, part of the **PuTTY** suite of free SSH tools.

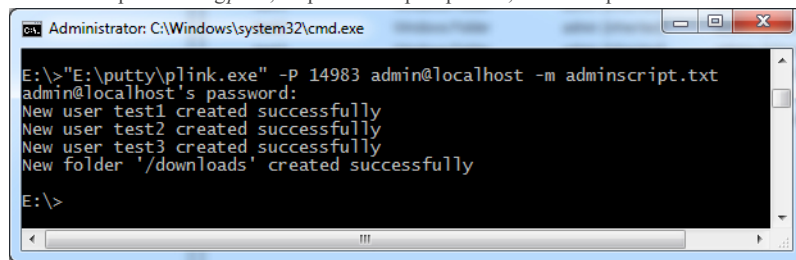
The example below shows how to run a script on the Admin site using *plink*. Note that the port number of the Admin site (default 14983) must be specified via the -P option:

```
plink.exe -P 14983 admin@localhost -m adminscript.txt
```

For example, the contents of a script that adds three users, and then adds a Windows folder mounted as "downloads" to the virtual file system that belongs to user5 is shown below:

```
useradd -c "user 1" -n user1 -r password test1
useradd -c "user 2" -n user2 -r password test2
useradd -c "user 3" -n user3 -r password test3
folderadd -t windows -v /downloads -p d:\downloads -u test1 -g users
```

When this script is run using *plink*, the password is prompted for, and the script is run on the Admin site of the server, as shown below. The scriptfile in this example is *adminscript.txt*:



```
Administrator: C:\Windows\system32\cmd.exe
E:\>"E:\putty\plink.exe" -P 14983 admin@localhost -m adminscript.txt
admin@localhost's password:
New user test1 created successfully
New user test2 created successfully
New user test3 created successfully
New folder '/downloads' created successfully
E:\>
```



# Administration with REST API - AdminAPI

CompleteFTP Enterprise MFT's AdminAPI is a REST API for performing administration tasks.

## Accessing the AdminAPI

The AdminAPI is disabled by default for security reasons. It can be enabled by enabling the AdminAPI web feature:

1. Open CompleteFTP Manager
2. Open the Sites panel
3. Make sure *Show system users/folders/sites* is checked in the *Options* menu at the bottom left
4. Select the Admin site
5. Open *Advanced HTTP/HTTPS Settings*
6. Select *Web features*
7. Click the ellipsis (...) at the right
8. Check the enabled checkbox next to *AdminAPI*
9. Click *Close* and *Apply changes*

Only accounts that are members of the *admins* group are able to access the AdminAPI. However, it's recommended to use an account other than the *admin* account, so that there's minimal disruption if it becomes necessary to remove access. It's therefore best to **create a new admin account** and use it only for this purpose. If specific IP filtering rules or network interfaces are to be used for AdminAPI access then a **new site could be added**, specifically for this purpose.

Authentication for the AdminAPI may be done with HTTP Basic Authentication using the user-name and password of the admin account created for this purpose (as described in the above paragraph). This authentication method is supported by most HTTP clients. For example, in **curl**, the `-u` option may be used to include the appropriate authentication information. For example, the following command will return a JSON object describing the account with user-name, *myuser*, using the admin account named, *myadmin*:

```
curl -u myadmin:mypassword https://myserver:14985/Admin/API/config/users/myuser
```

Note that, by default, curl validates the server's certificate against the Windows certificate store. This can be disabled with the `-k` argument, otherwise the certificate must either be a **CA certificate** or, if it's a self-signed certificate (the **default**) then it must be installed in the certificate store.

When starting out with the AdminAPI, it's a good idea to start by accessing it from the same machine on which CompleteFTP is installed. Once you've established that Admin API working in the way that you need when accessed from localhost, you can try to access it from another machine. By default, CompleteFTP blocks access to administration features from machines that are not on the same LAN, so the admin site's IP filters will need to be modified if such access is required.

## Using the AdminAPI

AdminAPI is a REST API, meaning that each operation is executed by sending an HTTPS request to CompleteFTP. Each request is either a HTTP GET or a HTTP POST. The latter request requires a body containing JSON. All requests return a JSON object with the following properties:

- **success** (boolean): either true or false
- **message** (string): ok if the call was successful, otherwise an error-message
- **apiVersion** (string): version of the API (currently 1.0.0)
- **type** (string): type of return value (e.g. Folder, FolderManager, Array, number)
- **value** (specified by **type** field): return value of API call

The base path is `/Admin/API`, although can be changed by editing the Admin API web feature. This version of the AdminAPI can only be used for performing configuration actions. These are done via the sub-path, `config`, so the path of all requests will start with `/Admin/API/config`. The default HTTPS port for the admin site is 14,985.

URLs for AdminAPI requests take the following form (in **WSN**):

```
https://hostname:14985/Admin/API/config/OBJECT[/METHOD[/ARGUMENTS]]
```

where **OBJECT** specifies an object such as a user, a group or a folder; **METHOD** (optional) is the name of a method of the specified object; and **ARGUMENTS** (optional) is one or more arguments (separated by `/`) passed to the method.

## Retrieving Objects

If an HTTP GET request is sent without specifying a method then the object is returned. For example, sending an HTTP GET request to `https://myserver:14985/Admin/API/config/users/MyUser` is the URL of the object representing the user, *MyUser*. It will return a JSON string representing this user:

```
{
  "success": true,
  "message": "ok",
  "apiVersion": "1.0.0",
  "uri": "https://myserver:14985/Admin/API/config/users/MyUser",
  "type": "User",
  "value": {
    "type": "Non-Windows",
    "inbuilt": false,
    "userName": "MyUser",
    "fullName": "My User",
    "description": "A user intended for testing.",
    "email": "myuser@test.com",
    "siteMapping": [{
      "siteName": "Default Site",
      "enabled": true,
      "homeFolder": "/Home/MyUser"
    }, {
      "siteName": "Admin",
      "enabled": false,
      "homeFolder": null
    }],
    ...
  }
}
```

## Setting object properties

Properties can be set by sending an HTTP POST to the same URL with either of the two following JSON strings:

```
{
  "fullName": "My test user"
}
```

or

```
{
  "properties": {
    "fullName": "My test user"
  }
}
```

## Invoking Methods

A method of an object may be invoked by placing its name after that of the object in the URL. If a method takes arguments then those may either be supplied in the URL itself or in the body of the request.

For example, a user may be deleted with the URL, <https://myserver:14985/Admin/API/config/users/remove/MyUser>. Alternatively, arguments for the method may also be passed to, <https://myserver:14985/Admin/API/config/users/remove>, via JSON, as follows:

```
{
  "arguments": [
    "MyUser"
  ]
}
```

While a user can be added with the URL, <https://myserver:14985/Admin/API/config/users/add/non-windows>, one would usually want to set a some properties of the new user. This may be done by posting the following JSON with the request:

```
{
  "userName": "MyUser"
}
```

If the arguments of a method are too complex to be placed in the URL *and* properties need to be set then a JSON object containing both an `arguments` field and a `properties` field. For example, to add a user and set the `userName` property, the following JSON would be posted to <https://myserver:14985/Admin/API/config/users/add>:

```
{
  "arguments": [
    "MyUser"
  ],
  "properties": {
    "userName": "MyUser"
  }
}
```

## Available Configuration Objects

The following configuration objects are available:

### **/Admin/API/config/folders**

A JSON array containing the full paths of all folders in the virtual file-system.

### **/Admin/API/config/folders/FOLDER\_NAME/{FOLDER\_NAME}**

A JSON object containing all properties of the specified folder. The returned object has the same properties and methods as the [JSS Folder Object](#).

### **/Admin/API/config/users**

A JSON array containing the user-names of all users.

### **/Admin/API/config/users/USER\_NAME**

A JSON object containing all properties of the specified user. The returned object has the same properties and methods as the [JSS User Object](#).

### **/Admin/API/config/groups**

A JSON array containing the names of all groups.

### **/Admin/API/config/groups/GROUP\_NAME**

A JSON object containing all properties of the specified group. The returned object has the same properties and methods as the [JSS Group Object](#).

### **/Admin/API/config/sites**

A JSON array containing the names of all sites.

### **/Admin/API/config/sites/SITE\_NAME**

A JSON object containing all the properties of the site. Note that spaces may be omitted, so `Default Site` is the same as `DefaultSite`. The returned object has the same properties and methods as the [JSS Site Object](#).

## Examples

The examples below demonstrate how to use the Admin API to perform a range of operations. Each example includes `curl` commands for both Windows and Bash consoles. The commands differ because quotes are handled differently on the two consoles.

### Add a user:

#### URL:

`https://localhost:14985/Admin/API/config/users/add`

#### JSON:

```
{
  "userName": "fred",
  "password": "Fred123"
}
```

#### Command (Windows):

```
curl -u username:password --data '{"userName":"fred","password":"Fred123"}' https://localhost:14985/Admin/API/config/users/add
```

#### Command (Bash):

```
curl -u username:password --data '{"userName":"fred","password":"Fred123"}' https://localhost:14985/Admin/API/config/users/add
```

### Set user's home folder:

#### URL:

`https://localhost:14985/Admin/API/config/users/fred/siteMapping/DefaultSite`

#### JSON:

```
{
  "homeFolder": "/Home/fred"
}
```

#### Command (Windows):

```
curl -u username:password --data '{"homeFolder":"","Home/fred"}' https://localhost:14985/Admin/API/config/users/fred/siteMapping/Default
```

#### Command (Bash):

```
curl -u username:password --data '{"homeFolder":"/Home/fred"}' https://localhost:14985/Admin/API/config/users/fred/siteMapping/Default
```

#### Get user data:

##### URL:

https://localhost:14985/Admin/API/users/fred

##### JSON:

none

##### Command (Windows):

```
curl -u username:password https://localhost:14985/Admin/API/config/users/fred
```

#### Command (Bash):

```
curl -u username:password https://localhost:14985/Admin/API/config/users/fred
```

#### Disable FTP for user:

##### URL:

https://localhost:14985/Admin/API/config/users/fred

##### JSON:

```
{
  "ftpEnabled":false
}
```

##### Command (Windows):

```
curl -u username:password --data '{"ftpEnabled":false}' https://localhost:14985/Admin/API/config/users/fred
```

#### Command (Bash):

```
curl -u username:password --data '{"ftpEnabled":false}' https://localhost:14985/Admin/API/config/users/fred
```

#### Add a folder:

##### URL:

https://localhost:14985/Admin/API/config/folders/add

##### JSON:

```
{
  "arguments": [
    "/Home/fred",
    "WindowsFolder",
    "C:\\Temp"
  ]
}
```

##### Command (Windows):

```
curl -u username:password --data '{"arguments":["/Home/fred","WindowsFolder","C:\\Temp"]}' https://localhost:14985/Admin/API/config/folders/add
```

#### Command (Bash):

```
curl -u username:password --data '{"arguments":["/Home/fred","WindowsFolder","C:\\Temp"]}' https://localhost:14985/Admin/API/config/folders/add
```

#### Get folder data:

**URL:**

https://localhost:14985/Admin/API/config/folders/Home/fred

**JSON:**

none

**Command (Windows):**

```
curl -u username:password https://localhost:14985/Admin/API/config/folders/Home/fred
```

**Command (Bash):**

```
curl -u username:password https://localhost:14985/Admin/API/config/folders/Home/fred
```

**Set folder mapping path:**

**URL:**

https://localhost:14985/Admin/API/config/folders/Home/fred

**JSON:**

```
{
  "mappingPath" : "C:\\Temp\\Downloads"
}
```

**Command (Windows):**

```
curl -u username:password --data '{"mappingPath":"C:\\Temp\\Downloads"}' https://localhost:14985/Admin/API/config/folders/Home/fred
```

**Command (Bash):**

```
curl -u username:password --data '{"mappingPath":"C:\\Temp\\Downloads"}' https://localhost:14985/Admin/API/config/folders/Home/fred
```

**Set owner of folder:**

**URL:**

https://localhost:14985/Admin/API/config/folders/Home/fred/access/owner

**JSON:**

```
{
  "userName": "fred"
}
```

**Command (Windows):**

```
curl -u username:password --data '{"userName":"fred"}' https://localhost:14985/Admin/API/config/folders/Home/fred/access/owner
```

**Command (Bash):**

```
curl -u username:password --data '{"userName":"fred"}' https://localhost:14985/Admin/API/config/folders/Home/fred/access/owner
```

# Administration with JSS scripts

## (requires Enterprise MFT)

CompleteFTP's configuration can be modified using Javascript (i.e. **JSS**). All administration-related tasks involve the **Config** object returned by the `system.getConfig()` function.

There are several ways to execute such scripts:

1. **Interactively** in the SSH terminal
2. **JSS script files** piped to CompleteFTP via SSH
3. **JSS custom commands** executed via SSH
4. **JSS web apps** used interactively through a browser

These will be detailed below.

### 1. Interactively entering JSS configuration commands

1. Make sure that SSH terminal is enabled for the Admin site
2. Make sure that SSH terminal is enabled for the admin user
3. Select JSS for the SSH terminal shell setting for the admin user (under the Scripting and Shells category)
4. Install an SSH client such as OpenSSH (e.g. in the Windows Subsystem for Linux) or [plink](#)
5. Connect to port 14983 as the admin user. If a public key-pair has been set up for the admin user then no password will be required, otherwise you will be prompted for the password.
6. Enter the JSS statements on line at a time.

For example, the following adds a user via OpenSSH run on Windows Subsystem for Linux using a public key for authentication:

```
john@Warp:~/tmp$ ssh admin@localhost -p 14983
admin@myserver:/Admin> var userName = "testuser";
admin@myserver:/Admin> var password = "password";
admin@myserver:/Admin> var homePath = "C:\\Temp";
admin@myserver:/Admin> var config = system.getConfig();
IMPORTANT! Changes to the config will not be saved until applyChanges() is called.
admin@myserver:/Admin> var newFolder = config.folders.add("/Home/" + userName, "windows", homePath);
admin@myserver:/Admin> var newUser = config.users.add();
admin@myserver:/Admin> newUser.userName = userName;
admin@myserver:/Admin> newUser.password = password;
admin@myserver:/Admin> newUser.siteMapping[0].homeFolder = newFolder.fullPath;
admin@myserver:/Admin> config.applyChanges();
```

### 2. Executing JSS script files via SSH

After repeating steps 1 to 4 from section:

1. Create a text file called `adduser.jss` containing the required JSS script

For example, the following script adds a user:

```
var userName = "testuser";
var password = "password";
var homePath = "C:\\Temp";
var config = system.getConfig();
IMPORTANT! Changes to the config will not be saved until applyChanges() is called.
var newFolder = config.folders.add("/Home/" + userName, "windows", homePath);
var newUser = config.users.add();
newUser.userName = userName;
newUser.password = password;
newUser.siteMapping[0].homeFolder = newFolder.fullPath;
config.applyChanges();
```

Note that all the constructs of EC3-level Javascript may be used.

2. Execute the script as follows:

```
john@Warp:~/tmp$ ssh admin@localhost -p 14983 < adduser.jss
```

### 3. Creating JSS custom commands for administration

1. Open CompleteFTP Manager
2. Select the Extensions panel
3. Click Add extension → JSS (Javascript) extension → Custom commands
4. Enter a Javascript function that performs the administrative task.

For example, the following function adds a user with a given user-name, password, and Windows home directory:

```
function CustomAddUser(userName, password, homePath) {  
  // get configuration  
  var config = system.getConfig();  
  
  // create the home folder  
  var newFolder = config.folders.add("/Home/" + userName, "windows", homePath);  
  
  // create a new user  
  var newUser = config.users.add();  
  newUser.userName = userName;  
  newUser.password = password;  
  newUser.siteMapping[0].homeFolder = newFolder.fullPath;  
  
  // apply the changes  
  config.applyChanges();  
}
```

5. Install an SSH client (it's not necessary to set JSS as the SSH terminal shell)
6. Execute the custom commands as follows:

```
john@Warp:~/tmp$ ssh admin@localhost -p 14983 "CustomAddUser testuser password C:\\Temp"
```

#### 4. Developing custom administration web-apps (Adminlets)

For information on developing 'adminlets', please refer to the [EnterpriseDT repository on GitHub](#).

# Administrator command-line

## Folder commands

Command	Description	Usage
chgrp	Change the group of a folder in the virtual filesystem	chgrp group virtualpath e.g. chgrp mygroup /test
chmod	Change the permissions of a folder in the virtual filesystem. Uses Unix-style permissions.	chmod [u g o][+ = -]permissions[...] virtualpath (permissions in Unix format, either octal or symbolic using r (read), w (write) and x (execute)) e.g. chmod u=rwx,g=rx,o=r /test  The operator + causes the selected mode to be added to the existing file mode; - removes the selected mode; and = causes them to be assigned and all other modes removed.  'u' designates owner permissions, 'g' represents group permissions, and 'o' represents the rest of the world's permissions.
chmodx	Change the extended permissions of a folder in the virtual filesystem. Uses CompleteFTP permissions.	chmodx [u g o][+ = -]extendedpermission[...] virtualpath (permissions <b>must</b> be one of those listed below e.g. chmodxu+DirList /test)  The operator + causes the selected mode to be added to the existing file mode; - removes the selected mode; and = causes them to be assigned and all other modes removed.  'u' designates owner permissions, 'g' represents group permissions, and 'o' represents the rest of the world's permissions.  <b>Permission Description</b> <b>None</b> No permissions <b>FileRead</b> File reading permission <b>FileWrite</b> File writing permission <b>FileDelete</b> File delete permission <b>FileAppend</b> File appending permission <b>FileRename</b> File rename permission <b>FileExecute</b> File execute permission <b>DirList</b> Directory list permission <b>DirChange</b> Change into directory permission <b>DirCreate</b> Create a subdirectory permission <b>DirRename</b> Rename directory permission <b>DirDelete</b> Delete directory permission <b>IgnoreFilters</b> Ignore file filters for this directory <b>AllRead</b> All read permissions <b>AllWrite</b> All write permissions <b>AllExecute</b> All execute permissions <b>ReadOnly</b> Read only permissions <b>All</b> All available permissions
chown	Change the owner of a folder in the virtual filesystem	chown username virtualpath e.g. chown myuser /test
folderadd	Create a new folder in the virtual filesystem. For Windows virtual folders, the Windows folder specified will be created if it does not exist.	folderadd -t virtual\windows\networkmacro -v virtualpath [-p path] [-u username] [-g groupname] [-c]  <b>-t virtual\windows\networkmacro</b> the type of folder to create <b>-v virtualpath</b> the virtual path of the new folder <b>-p path</b> the Windows path of the new folder (for Windows and network folders) <b>-u username</b> the name of the user owning the new folder (optional) <b>-g groupname</b> the name of the group owning the new folder (optional) <b>-c</b> create the corresponding Windows folder for type "windows" if it does not exist (optional)  e.g. folderadd -t windows -v /test/temp -p d:\\temp -c (to create a virtual Windows folder pointing to d:\\temp)
encrypt	Encrypt a file in the virtual filesystem	encrypt [-h] -s sourcefile -d destination e.g. encrypt -s /home/myuser/test_plain.txt -d /home/myuser/test_encrypted.txt
decrypt	Decrypt a file in the virtual filesystem that has previously been encrypted.	decrypt [-h] -s sourcefile -d destination e.g. decrypt -s /home/myuser/test_encrypted.txt -d /home/myuser/test_plain.txt
folderdel	Delete an existing folder in the virtual filesystem	folderdel -v virtualpath



foldermmod	Modify an existing Windows folder in the virtual filesystem	foldermmod -v virtualpath -p pathtochangeto  e.g. foldermmod -v /test/temp -p d:\downloads (to modify /test/temp to point to d:\downloads). Note that foldermmod can only be used for Windows folders.
------------	---	--

User commands

Command	Description	Usage
useradd	Add a new non-Windows user. Optionally, the MD5 hash of the password can be set rather than using the raw password (which will be converted to a hash).	<code>useradd [-h] [-i] [-c comment] [-d Windows_home_dir] [-n fullname] (-p hashedpassword -r rawpassword) [-k publickeystring] username</code>  <b>-h</b> show help text <b>-i</b> ignore error if user already exists <b>-c comment</b> descriptive comment for user (optional) <b>-n fullname</b> full name of user (optional) <b>-d Windows_home_dir</b> Windows home directory of user (optional) <b>-p hashedpassword</b> hashed password for user (MD5 or SHA 1, not recommended) <b>-r rawpassword</b> supplying raw password of user which is recommended <b>-k publickeystring</b> the user's public key as a base64 encoded string <b>username</b> the username of the user to be created  e.g. useradd -c "Test user" -n "Tester" -r test1234 test e.g. useradd -c "Test user" -n "Tester" -r test1234 -d d:\temp test (sets d:\temp as test's home directory) e.g. useradd -r test1234 -k AAAAB3NzaC...7gKtM= test
usermod	Modify an existing user.	<code>usermod [-h] [-c comment] [-d home_dir] [-e true false] [-n fullname] [-k publickeystring] [-p hashedpassword -r rawpassword] [-t tcpipprotocols] username</code>  <b>-h</b> show help text <b>-c comment</b> descriptive comment for user (optional) <b>-d Windows_home_dir</b> new Windows home directory of user (optional) <b>-e true false</b> enable (true) or disable user (false) <b>-n fullname</b> full name of user (optional) <b>-k publickeystring</b> the user's public key as a base64 encoded string <b>-p hashedpassword</b> hashed password for user (MD5 or SHA 1, not recommended) <b>-r rawpassword</b> supplying raw password of user which is recommended <b>-t tcpipprotocols</b> where tcpipprotocols is a string listing protocols to be set, added to or removed from the user and in the format: =protocol1,protocol2,... or -protocol1,protocol2,... or +protocol1,protocol2,... Protocols available are: ftp, ftps, sftp, scp, ssh, sshforwarding, http, https <b>username</b> the username of the user to be modified  e.g. usermod -t =ftp,ftps,sftp,scp username (to set these protocols for this user) e.g. usermod -t +ssh,sshforwarding username (to add these protocols for this user) e.g. usermod -d d:\temp username (to set d:\temp as /Home/username) e.g. usermod -e false username (to disable this user)
userdel	Delete a user. This cannot be undone!	userdel username

Site commands

Command	Description	Usage
start	Start all sites (other than the admin site, which is required to be running to access this command)	start
stop	Stop all sites (other than the admin site, which is required to be running to access this command). Use with caution, as calling it will stop all public sites.	stop

General commands

Command	Description	Usage
help	Display help text for commands	help command e.g. help useradd
logout	End current administrator session	logout
quit	End current administrator session	quit
version	Returns the current server version	version

# Administration Log

CompleteFTP's Administration Log is a vital feature designed to empower users with the ability to trace all configuration changes made within the application. This logging capability ensures that administrators can monitor, audit, and, if necessary, roll back changes, providing an additional layer of oversight and control over the system's configuration.

## Location of the Administration Log

The Administration Log is a text file located at: `C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Logs\Config.log`

## Understanding the Administration Log

When you open the `Config.log` file, you'll find a chronologically ordered record of all configuration changes. Each entry in the log is separated by a blank line, making it easier to distinguish between individual events.

## Structure of a Log Entry

Each log entry begins with a timestamp indicating when the change was made, followed by the server's name, various details about the change, and, finally, the identity of the user who made the change. If a change was recorded by the server itself, the user will be denoted as `(server)`.

## Reading the Log

To understand the changes recorded in the log:

- **Start with the Timestamp:** This helps you pinpoint when a particular change was made.
- **Identify the User:** Knowing who made the change can be crucial, especially when troubleshooting or auditing.
- **Examine the Change Details:** Look at what was added, deleted, or modified. This will give you insight into the nature and scope of the change.

## Configuration Entities

The Administration Log captures changes to various configuration entities within CompleteFTP. Understanding these entities and their functions is essential for interpreting the log's content. Here's a brief overview of the primary entities:

- **User:** Represents an individual user.
- **Group:** Denotes a user group.
- **Membership:** The association of a user with a group.
- **Node:** A folder in the virtual file-system.
- **Site:** A specific site within CompleteFTP.
- **SiteUser:** Maps a user to a site.
- **Server:** Represents a server. Clusters have multiple servers, including one primary and potentially several secondaries.
- **ServerSite:** Maps a site to a server.
- **WebApp:** Denotes a web-app.
- **SiteWebApp:** Maps a web-app to a site.
- **Authenticator:** An alternative means of authentication.
- **UniqueUser:** Refers to logins. Each authenticator can associate multiple unique-users with each User entity. These associations are termed as UniqueUsers.

## Best Practices

Always ensure you back up the `Config.log` regularly, monitor its size, and limit its access to only authorized personnel.

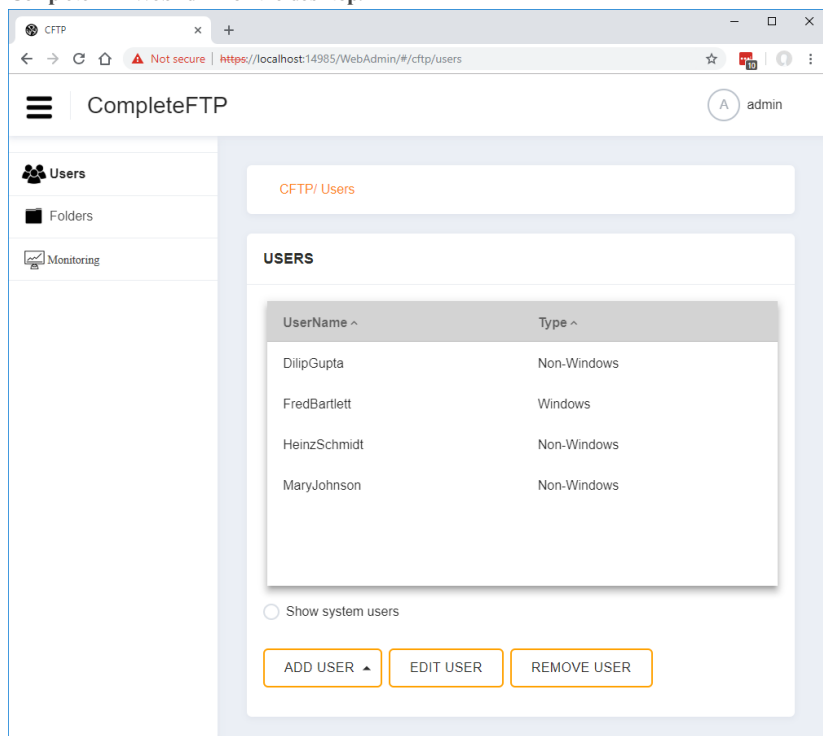
## Conclusion

The Administration Log in CompleteFTP offers a comprehensive record of configuration changes, aiding in system monitoring, auditing, and troubleshooting. Familiarity with the various configuration entities and the log's structure ensures effective use of this feature.

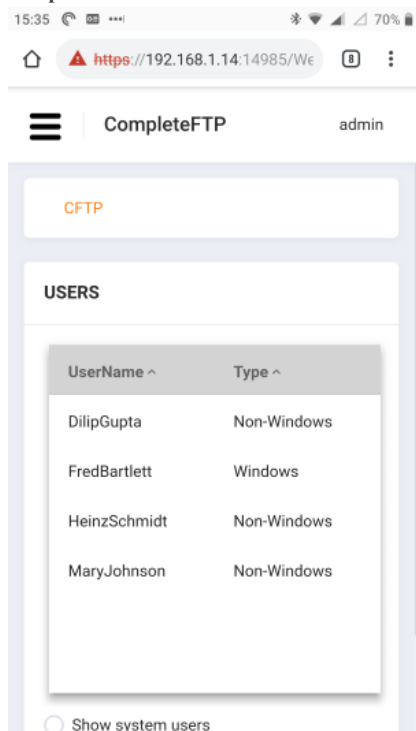
# CompleteFTP WebAdmin: Web-App for CompleteFTP Administration

CompleteFTP WebAdmin is CompleteFTP's web-app for administration. It provides basic, day-to-day administration features and is not meant to replace CompleteFTP Manager, which provides full control over CompleteFTP's configuration.

CompleteFTP WebAdmin on the desktop:



CompleteFTP WebAdmin on a mobile:



WebAdmin's features includes:

- adding, editing and removing user accounts
- adding, editing and removing folders in the virtual file-system
- monitoring current activity

## Accessing WebAdmin

The WebAdmin web-app is available via HTTPS on port 14,985 of your CompleteFTP server. If you're viewing this article on a server that's running CompleteFTP then you should be able to open the app via the following link: <https://localhost:14985/>. When logging in you need to use the same credentials used when logging with CompleteFTP Manager. Note that by default IP filtering rules on the admin site are restricted to localhost and LAN address spaces, so it's not accessible from other IP addresses.

## Future Development of WebAdmin

WebAdmin is a relatively new product and therefore has a limited feature-set. We're keen to get feedback from users on which features to prioritise development, so please [let us know](#) what you'd like to see added.

# WebAdmin: Adding Users

You can add two types of users: Windows users and non-Windows users.

## Adding Windows Users

Adding a Windows user in CompleteFTP allows a user to log into CompleteFTP using their Windows credentials and to access the files for which that account has the corresponding permissions.

To add an existing Windows user, click the 'Add User' link in the Users panel and select 'Windows User' from the list.

NON-WINDOWS USER

WINDOWS USER

ADD USER

EDIT USER

REMOVE USER

Once selected this will bring up the 'Add Windows User' dialog page, and you will see a list of Windows user accounts that is currently associated with your computer. select the Windows user who you wish to add, and then select next.

Add Windows User

Step 1: Set Username

Step 2: Select home folder

Step 3: Verify Details

UserName	FullName	Description
JimBob	Jim Bob	JimBob@Domain.com
PeteOsbourne	Peter Osbourne	PeteOsbourne@Domain.com
SallyRyan	Sally Ryan	SallyRyan@Domain.Com

CANCEL

NEXT

Select the way you want the user's home folder to be created. There are two options: 1) By creating a default home folder

Create default home folder

/Home/benja (C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Users\JimBob)

Or 2) by selecting a non-default home folder

Select non-default home folder

Name	Type
C-drive	WindowsFolder
> Home	WindowsFolder
> Public	WindowsFolder
WindowsFolder	WindowsFolder

Once you have chosen a default home folder you can proceed to the next page which will give you the option to verify your details. Once you have verified the details, you should choose 'done'. The windows user should now have been added.

Add Windows User

Step 1: Set Username

Step 2: Select home folder

Step 3: Verify Details

- Username: JimBob
- Full name: Jim Bob
- Description: JimBob@Domain.com
- Home folder: /Home/JimBob

CANCEL

PREVIOUS

DONE

## Adding Non-Windows Users

'Adding a non-Windows user', should be selected in order to add a user, who does not already exist as a current Windows user on the server machine (or domain). This type of user does not have to exist as a Windows user.

NON-WINDOWS USER

WINDOWS USER

ADD USER

EDIT USER

REMOVE USER

You will then be instructed to fill in the username and details which you will see at 'Step 1' in the Dialog box.

Step 1: Set Username

Step 2: Set Authentication

Step 3: Select home folder

Step 4: Verify Details

Username (required)

JackJohnson

Full name (optional)

Jack Johnson

Description (optional)

Email (optional)

JackJohnson@Domain.com

CANCEL

NEXT

A password must be set in 'Step 2' with at least 6 characters.

Step 1: Set Username

Step 2: Set Authentication

Step 3: Select home folder

Step 4: Verify Details

Password

Show characters

\*\*\*\*\*

Confirm Password

\*\*\*\*\*

CANCEL

PREVIOUS

NEXT

In step 3, the home directory for this user must be selected. By default, the new user's home directory, will be created in the default 'Home' directory

Step 1: Set Username

Step 2: Set Authentication

Step 3: Select home folder

Step 4: Verify Details

Create default home folder

/Home/JackJohnson (C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Users\JackJohnson)

Select non-default home folder

Name	Type
C-drive	WindowsFolder
> Home	WindowsFolder
> Public	WindowsFolder
WindowsFolder	WindowsFolder

ADD ROOT FOLDER

ADD SUB FOLDER

CANCEL

PREVIOUS

NEXT

Step 4, and you will be asked to verify the details, so make sure that the details are correct and then select the 'done' button to add the new user to the server.

Step 1: Set Username

Step 2: Set Authentication

Step 3: Select home folder

Step 4: Verify Details

- Username: JackJohnson
- Full name: Jack Johnson
- Description:
- E-mail: JackJohnson@Domain.com
- Home folder: /Home/JackJohnson

CANCEL

PREVIOUS

DONE

Once completed you should see your new user on the user page.

UserName ^	Type ^
BobBrown	Non-Windows
JackJohnson	Non-Windows
JaneJanice	Non-Windows

# WebAdmin: Editing Users

To edit a user you should select the user you wish to modify, and then select the 'edit user' dialog box

USERS

UserName ^	Type ^
BobBrown	Non-Windows
JackJohnson	Non-Windows
JaneJanice	Non-Windows

☐ Show system users

ADD USER

EDIT USER

REMOVE USER

You will then be directed to the 'edit user' page, where you are able to change any of the details which were initially used to create the account, and you are also able to change the protocols for the user. Once you have completed editing the user you are then able to return to the user page by clicking on the done button.

Edit User

Username

JackJohnson

CHANGE PASSWORD

Home Folder

Name	Type
C-drive	WindowsFolder
> Home	WindowsFolder
> Public	WindowsFolder
WindowsFolder	WindowsFolder

Allowed Protocols

☒ FTP ☒ FTPS ☒ SFTP ☒ SCP ☐ SSH terminal ☒ HTTP ☒ HTTPS

Full name

Jack Johnson

Description

Email

JackJohnson@Domain.com

CANCEL

DONE



# WebAdmin: Removing Users

To remove a user select the user you wish to remove then select the remove user dialog box

USERS

UserName ^

BobBrown

JackJohnson

JaneJanice

☐ Show system users

ADD USER ^

EDIT USER

REMOVE USER

A dialog box will appear to confirm that you want to delete the user, press ok

Delete users

×

Are you sure you want to delete these users: JackJo...?

OK

CANCEL

Once you have pressed ok you will see on the Users page that the user you wanted to delete is no longer there.

USERS

UserName ^	Type ^
BobBrown	Non-Windows
JaneJanice	Non-Windows

# WebAdmin: Adding Folders

You can add two types of Folders: Windows Folders and Virtual Folders.

## How to Add Windows folders

Windows folders are standard folders found on the local machine. They can be inserted into the virtual folder hierarchy at any point, including the root. they are added by selecting "Add root folder" or "Add sub-folder", and choosing "Windows folder".

VIRTUAL FOLDER

WINDOWS FOLDER

ADD ROOT FOLDER

ADD SUB FOLDER

EDIT FOLDER

REMOVE FOLDER

In the next screen you will be shown the available drives on the local machine, select the drive you wish to add the windows folder to and then select "ADD"

Add Windows Root Folder

Name

Type

> C:\

Drive

CANCEL

ADD

Once the Windows folder has been added, it will then be shown in the folder screen as a "WindowsFolder"

Name	Type
C-drive	WindowsFolder
> Home	WindowsFolder
> Public	WindowsFolder
WindowsFolder	WindowsFolder

## How to Add a Virtual folder

Virtual folders don't physically exist. They are used to create virtual hierarchies of folders under which real folders can be placed. CompleteFTP administrators should carefully plan out their folder structure, and create virtual folders as appropriate. For example, a virtual folder called 'home' might be created at the root level, by selecting "Add root folder" and choosing the 'Virtual folder' option, as shown below.

VIRTUAL FOLDER

WINDOWS FOLDER

ADD ROOT FOLDER

ADD SUB FOLDER

EDIT FOLDER

REMOVE FOLDER

Once the Virtual folder has been selected, there will be a screen for the user to input the folder name. Once a folder has been named, the user will then be able to add the folder.

Add Virtual Root Folder

Input the folder name

VirtualHome

CANCEL

ADD

Once the virtual folder has been added, it will then be shown on the folder screen as a "VirtualFolder"

Name	Type
C-drive	WindowsFolder
> Home	WindowsFolder
> Public	WindowsFolder
VirtualHome	VirtualFolder
WindowsFolder	WindowsFolder

# WebAdmin: Editing Folders

To edit a folder the user should select the folder that they wish to edit. Once it has been selected, they should select the 'edit folder' button.

FOLDERS

Name	Type
C-drive	WindowsFolder
> Home	WindowsFolder
> Public	WindowsFolder
VirtualHome	VirtualFolder
WindowsFolder	WindowsFolder

☐ Show system folders

ADD ROOT FOLDER

ADD SUB FOLDER

EDIT FOLDER

REMOVE FOLDER

In the edit folder screen, the user will be able to change the name of the folder, change the drive destination where they want the folder to be located on the local machine, as well as change the user's permissions for the folder.

Edit Folder

Input the folder name

VirtualHome

Access

Owner

Group

All Users

Owner:

admin

Owner permissions

☒ Download Files

☒ Upload Files

☒ Rename Files

☒ Delete Files

☒ List Folder Contents

☒ Change Into Folder

☒ Create Subfolders

☒ Rename Subfolders

☒ Delete Subfolders

☒ Ignore Filters

INHERIT PARENT PERMISSIONS

CANCEL

DONE

The user is only able to modify non System windows folders. This means that virtual folders cannot be modified; A dialog warning box will show that the folder cannot be edited.

This folder cannot be edited

Only non-System Windows Folder and Virtual Folder can be edited, please choose again

OK

# WebAdmin: Removing Folders

To remove a folder, the user should select the folder they wish to remove, and then choose the 'remove folder' button.

FOLDERS

Name	Type
C-drive	WindowsFolder
> Home	WindowsFolder
> Public	WindowsFolder
VirtualHome	VirtualFolder
WindowsFolder	WindowsFolder

☐ Show system folders

ADD ROOT FOLDER

ADD SUB FOLDER

EDIT FOLDER

REMOVE FOLDER

A dialog box will appear to confirm whether you wish to delete the folder, so press OK.

Delete folders

Are you sure you want to delete this folder: VirtualHome?

OK

CANCEL

Once the folder has been deleted , it will no longer appear in the 'folders' screen.

FOLDERS

Name	Type
C-drive	WindowsFolder
> Home	WindowsFolder
> Public	WindowsFolder
WindowsFolder	WindowsFolder

# WebAdmin: Monitoring Connections

The Monitoring panel in the CompleteFTP manager displays real-time logging. It gives details about the currently connected users and the corresponding IP addresses which are connected.

MONITORING

User Name	Protocol	Site Name	Remote Address	Local Port	Connected	Disk Usage
bobbrown	FTPS	Default Site	127.0.0.1 : 56581	21	11:29:25 AM	No quota
janejanice	FTPS	Default Site	127.0.0.1 : 56586	21	11:29:38 AM	No quota
bobbrown	HTTP	Default Site	127.0.0.1 : 56515	80	11:27:16 AM	No quota
Jane/Janice	HTTPS	Default Site	127.0.0.1 : 56543	443	11:27:50 AM	No quota

Connections: 4 | Last update: 3/8/2019, 11:29:48 AM

☐ Show Admin Connections

☐ Auto Refresh

REFRESH

KILL CONNECTION

You are also able to refresh the users who are or have been logged in, by selecting the refresh button.

REFRESH

KILL CONNECTION

Or you can simply just select the auto refresh button and this will automatically refresh and update the monitoring page.

☐ Show Admin Connections

☒ Auto Refresh

REFRESH

KILL CONNECTION

# WebAdmin: Killing Connections

To Delete a user's connection , you should select the user that is currently connected and whom you wish to remove. Once they have been selected, click on the 'kill connection' option. box and this will delete there connection to the server

MONITORING

User Name	Protocol	Site Name	Remote Address	Local Port	Connected	Disk Usage
bobbrown	FTPS	Default Site	127.0.0.1 : 56581	21	11:29:25 AM	No quota
janejanice	FTPS	Default Site	127.0.0.1 : 56586	21	11:29:38 AM	No quota
bobbrown	HTTP	Default Site	127.0.0.1 : 56515	80	11:27:16 AM	No quota
JaneJanice	HTTPS	Default Site	127.0.0.1 : 56543	443	11:27:50 AM	No quota

Connections: 4 | Last update: 3/8/2019, 11:31:47 AM

Show Admin Connections

Auto Refresh

REFRESH

KILL CONNECTION

Or you can simply right click the user you wish to remove, and click 'kill connection'.

User Name	Protocol	Site Name	Remote Address	Local Port	Connected	Disk Usage
bobbrown	FTPS	Default Site	127.0.0.1 : 56581	21	11:29:25 AM	No quota
janejanice	FTPS	Default Site	127.0.0.1 : 56586	21	11:29:38 AM	No quota
JaneJanice	HTTPS	Default Site	127.0.0.1 : 56543	443	11:27:50 AM	No quota

Kill Connection

Once you have deleted the connection, you should then see that the user has been disconnected.

MONITORING

User Name	Protocol	Site Name	Remote Address	Local Port	Connected	Disk Usage
bobbrown	FTPS	Default Site	127.0.0.1 : 56581	21	11:29:25 AM	No quota
janejanice	FTPS	Default Site	127.0.0.1 : 56586	21	11:29:38 AM	No quota

# Security overview

Security is a never-ending battle when it comes to servers - there is always the threat of a malicious attacker gaining access to your systems and causing damage.

Fortunately, there are a number of steps that can be taken as detailed in [how to secure your server](#) that will minimize the possibility of your CompleteFTP server being compromised.

The most important of these is to keep your software up to date with the latest patches (both CompleteFTP and Windows), and to actively monitor your system for signs of intruders.

# How to disable SSL 3.0, TLS 1.0, and TLS 1.1

## SSL 3.0

The world-wide standard for securing TCP connections began as the Secure Sockets Layer, or SSL. SSL 3.0 was succeeded by Transport Layer Security (TLS) in 1999, and together they are often known as SSL/TLS. SSL 3.0 is now obsolete and considered a security risk. It is disabled by default in CompleteFTP.

## TLS 1.0 and TLS 1.1

Recently (March 2021), the Internet Engineering Task Force (IETF) [announced](#) that TLS 1.0 and 1.1 are formally deprecated. They should no longer be enabled because of their lack of support for recommended cryptographic algorithms. CompleteFTP server administrators should disable TLS 1.0 and TLS 1.1. This can be done via the CompleteFTP management application. In the FTP/FTPS Settings, navigate to Advanced FTP/FTPS Settings, and then Security Settings. Set the Minimum SSL Version to TLS 1.2, and click on Apply Changes.



# Licensing overview

The *Licensing* panel displays licensing details for the CompleteFTP server installation that the manager is currently connected to. This panel is slightly different depend on what kind of license is applying.

## Trial license



In trial mode, you are offered a fully-operational trial version of CompleteFTP. The trial period expires after 31 days and you can view how many days you have left in the trial via the *Licensing* panel.

The Edition displays the product edition that's installed (Free, Standard, Professional or Enterprise MFT). Some features are only available in higher editions. Click the 'compare editions' link to see a [comparison](#).

The View prices/purchase license link opens [the CompleteFTP Pricing page](#) on the enterprisedt.com website. Licenses may be purchased from there either online or by purchase order. Online orders are processed instantly, allowing you to activate your server within a minute or two.

Once a license has been purchased, you must use it to activate CompleteFTP via the Apply purchased license link. See [here](#) for more details on how to activate the paid editions.

During trial period, you can easily switch between editions. In case you switch to the lowest one, i.e. Free Edition, the *Licensing* panel will be given a different name, i.e. 'Activation'.

**ACTIVATION**

Step-by-step guides

[Activate for free](#)

APPLY CHANGES

Currently running **Free Edition**  
Please [activate for free](#) within 31 days.

Switch to: [Enterprise MFT](#)  
[Professional Edition](#)  
[Standard Edition](#)  
Free Edition

Compare

**Switch to Standard Edition**

Switch to CompleteFTP Standard Edition. Click 'Compare' to compare features of the four editions.

Free Edition can be activated free of charge for permanent use via the 'activate for free' link. See [here](#) for more details on how to activate the Free Edition.

## Paid license

Once you have [activated a paid license](#), the *Licensing* displays your purchased license's details including the purchased edition and the purchase reference.

**LICENSING**

Step-by-step guides

APPLY CHANGES

Your purchase reference is [REDACTED]

Currently running **Enterprise MFT**

[Apply new license](#)  
[Release activation](#)  
[Extend/renew support agreement](#)

**Apply purchased license**

Once you've purchased a license you must use it to activate CompleteFTP. If the machine, on which CompleteFTP server is running, has an Internet connection then this may be done very quickly from within CompleteFTP. If it doesn't then access to another machine that does have an Internet connection and a web-browser is required.

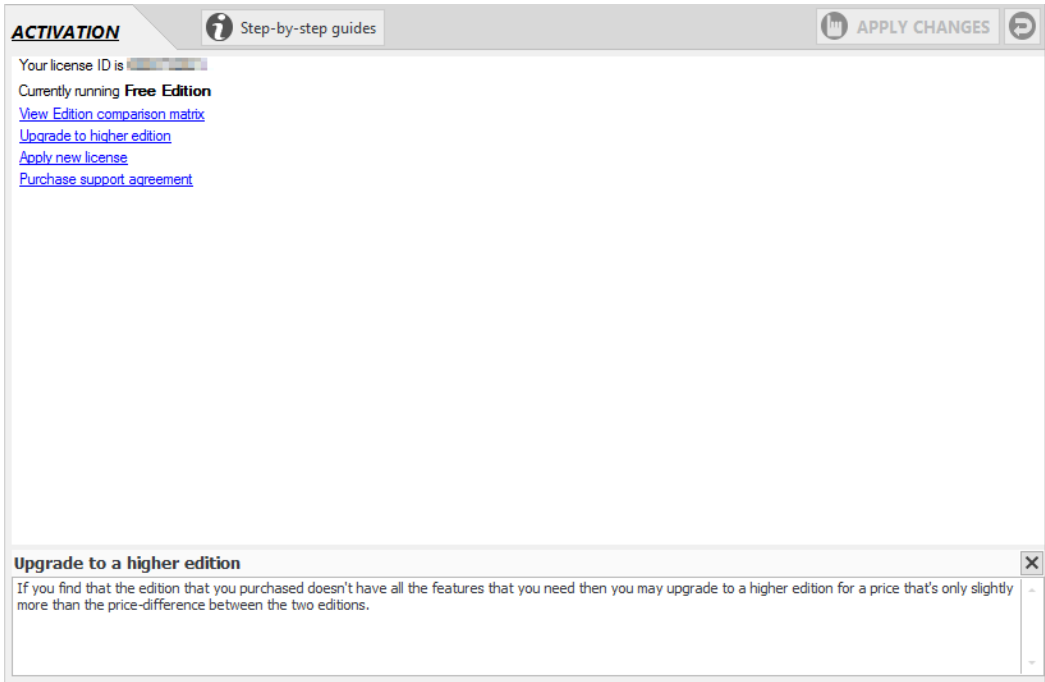
The paid license never expires but you may want to activate a new license over the top of your existing license via the Apply new license link. This is useful when upgrading from a lower edition to a higher edition.

If you need to move/migrate this installation of CompleteFTP to another machine then you must first release the activation on the existing machine by clicking the Release activation link and following the instructions. Once you done this you will be able to activate CompleteFTP on the new machine using the same license.

Your initial purchase of CompleteFTP included a 12 month support agreement, which provided you with updates and access to support by experienced engineers. If you'd like to maintain this support agreement then you may like to purchase an extension via the Extend/renew support agreement link.

Free license

Once you have **activated the Free Edition**, the *Activation* panel displays your free license's details.



If you find that the Free Edition that you previously activated doesn't have all the features that you need then you may want to upgrade to a higher edition. Click the Upgrade to higher edition link to **purchase a paid edition**. As soon as a license is purchased, use the Apply new license link to activate your purchased license.

Technical support is not included in the Free license, but you can **purchase a support package** separately via the Purchase support agreement link.

# How to resolve restricted mode

If CompleteFTP detects an invalid or missing license, it goes into *restricted mode*. In this mode it only permits local connections (i.e. connections originating from the same machine it is installed on).

This problem is generally because of a [migration to a new machine](#). It is also possible if CompleteFTP is running in a virtual machine that the machine's ID has been changed. In both these scenarios the machine ID encoded in the license does not match that of the new machine.

The resolution is to [activate](#) the installation. This necessarily uses up one of your purchased licenses. If you are moving your CompleteFTP installation, you can [contact support](#) to request an additional license be allocated.

# How to automatically import a license

Sometimes it can be extremely useful to be able to automatically import an activation key into a CompleteFTP configuration file (config.db).

For example, this may be required in a failover scenario. If CompleteFTP is set up on a primary machine, and a secondary machine maintains a backup, administrators may choose to regularly copy config.db across to the failover machine.

However the primary machine's config.db file will contain the activation key for the primary machine. The activation key for the failover machine must be inserted in config.db for CompleteFTP to operate with this configuration file on the failover machine.

This is easily accomplished by entering the activation key via *Enter Activation Key* in the Licensing panel. However this is a manual process, and it may be desirable for the process to be automated.

Updating the activation key automatically can be accomplished with the *license* utility that can be found in the Server directory of the installation (typically C:\Program Files\CompleteFTP\Server). This utility permits importing or exporting of the activation key to or from config.db:

```
license -c config.db [-i infile|-e outfile|-d] [-h]
-c config.db : CompleteFTP configuration file
-i infile : file to import activation key from
-e outfile : file to dump activation key to
-d : delete the activation key from the database
-h : show this help
```

For example, to import an activation key saved in a text file:

```
license -c C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\config.db -i license.txt
```

Note that this tool will not work with CompleteFTP **clustered servers** (as replicating the configuration is the task of the cluster). Enterprise MFT is required for cluster support. You can contact us to upgrade to Enterprise MFT from the Standard or Professional Editions if you wish to access this functionality.

# How to set up disaster recovery (DR) server

Disaster recovery planning is essential to ensure minimal downtime and data loss during unforeseen events. If you're using CompleteFTP, you can set up a disaster recovery (DR) server by maintaining a secondary server with a copy of the configuration from your primary server. However, simply copying the configuration file (config.db) between servers is not enough, especially because each server must use its own unique activation key. This article explains how to automate this process using CompleteFTP's built-in license tool, allowing your disaster recovery setup to switch over seamlessly.

## Steps for Setting Up a Disaster Recovery (DR) Server

### 1. Set Up the Secondary Machine (DR Server)

Install **CompleteFTP** on the secondary server, ensuring it's compatible with your primary setup. Ensure that both servers have the same version of CompleteFTP to avoid any compatibility issues when transferring configurations.

### 2. Backup Configuration from Primary Server

To ensure the secondary server mirrors the primary, you need to regularly copy the *config.db* file from the primary server to the DR server. The *config.db* file stores all CompleteFTP configurations, including users, settings, and permissions.

You can find the *config.db* file at the following location on the primary server:

```
C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\config.db
```

Regularly copying this file ensures that the secondary server is ready to take over with the most recent configurations in the event of failure.

You may also need to regularly copy the user home directories, to ensure that users have the latest data available. They can be found here:

```
C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\Users\
```

If you have customized other files in CompleteFTP, you will also need to copy these to the secondary server.

### 3. Handling the Activation Key

The *config.db* file contains the activation key for the server it is located on, as it is programmatically tied to its Windows installation. Therefore when the config.db is copied from the **primary server** to the DR server, it will contain the wrong activation key, and the DR server will be unlicensed and will not function correctly (it will not permit external connections to it).

To resolve this, the DR server needs to have its own activation key restored, which means a backup copy of it must be stored **prior** to overwriting its *config.db* file. This assumes you have previously activated the **DR server**. To do this, you must export its activation key to a file. After copying over the **primary server** configuration file, you then need to import the DR license back into its (new) *config.db* file. The export and import are done using the **license tool**. This utility is located in the CompleteFTP installation folder, typically at:

```
C:\Program Files\Complete FTP\Server
```

### 4. Exporting the DR activation key

To export the activation key from the DR server, use this command:

```
license -c "C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\config.db" -e "C:\path\to\output\license.txt"
```

### 5. Importing the DR activation key

After the primary server's configuration file has been copied over the configuration file of the DR server, the activation key that has been exported from the DR server needs to be imported back into its configuration file. To do so, **stop the DR CompleteFTP service** and then use this command:

```
license -c "C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\config.db" -i "C:\path\to\license.txt"
```

After importing the activation key, restart the CompleteFTP service on the DR server. It should be a replica of the primary server with its own activation key.

### 6. Important Considerations

**Clustered Servers:** The license utility cannot be used with **CompleteFTP clustered servers**. Clusters handle configuration replication automatically, and you need **Enterprise MFT** for full cluster support.

**Regular Backups:** Ensure that you schedule regular backups of the *config.db* file to the DR server, together with any user home directories and other files that are required to keep the DR server in sync with the primary server. You can automate the process, including use of the license tool to update the DR activation. Then the DR server will be ready for quick deployment.

# How to Activate Free Edition

All that is required to activate CompleteFTP Free Edition is a valid e-mail address. Information related to CompleteFTP, including the announcements of updates, will be sent to this e-mail. You can opt out of these e-mails at any time. Your e-mail address will not be shared with any other organisation.

To activate CompleteFTP Free Edition, please complete the following steps:

1. Open CompleteFTP Manager and connect to your server.
2. Select the Activation tab.
3. Click 'activate for free'.
4. You will be asked whether or not you have an account with EnterpriseDT. If you do then click 'Yes' and go to [Registering additional Free Edition servers](#) below, otherwise click 'No'.
5. Enter a valid e-mail address, your name and, optionally, the name of your company.
6. Click the button labelled 'Request verification code'. This will tell the EnterpriseDT server to send a verification code to the e-mail address that you entered. It should arrive within a minute or so.
7. Copy the verification code from the e-mail to the clipboard.
8. Paste the code into the last field on the User Registration Form.
9. Click 'Register'. After a few seconds your account will have been registered and your server activated.

## Registering additional Free Edition servers

You can register an unlimited number of Free Edition servers using a single EnterpriseDT account. To do this, follow steps 1 to 3 (above) and then click 'Yes' when asked if you already have an account. You will then be prompted for your user-name and password. Use the same user-name that you received when you first registered your account. When you click the Activate button CompleteFTP Manager will automatically activate your server.

## Registering Free Edition without a direct Internet connection

If the machine on which CompleteFTP Manager is running doesn't have an Internet connection then please follow the steps below:

1. Open CompleteFTP Manager and connect to your server.
2. Select the Activation tab.
3. Click 'activate for free'.
4. You will be asked whether or not you have an account with EnterpriseDT. Click 'Yes' or 'No'.
5. If you've previously registered for a free license then use the License ID that was e-mailed to you. If you haven't previously registered then follow the 'register now' link in step 0 to register your account and receive your License ID (by e-mail).
6. Enter the License ID in step 1.
7. Click the link marked 'copy' in step 2 to copy it to the clipboard.
8. Send the link a computer that does have Internet access and open it.
9. On the web-page that opens enter your user-name and password.
10. Copy the Activation Key shown on the web-page to the clipboard.
11. Send the Activation Key to the computer that's running CompleteFTP.
12. Paste the Activation Key into the field in step 4.
13. Click OK to complete activation.

For instructions with pictures on how to activate the free edition, please refer to Step-by-step guide: [Activate CompleteFTP free](#).

# How to Activate Standard, Professional or Enterprise MFT

An installation of CompleteFTP Standard, Professional or Enterprise MFT may be activated once a license has been purchased. Licenses may be ordered from the [EnterpriseDT website](#).

Once your order has been processed a message will be sent to the technical contact e-mail address that you provided. This message will contain the Purchase Reference of your order, your user-name and a link that you can use to set your password. You will need your password to activate your installation. Once you have your password please complete the following steps:

1. Open CompleteFTP Manager and connect to your server.
2. Select the Licensing tab.
3. Click 'Apply purchased license'.
4. If the computer on which CompleteFTP Manager is running has an Internet connection then enter the Purchase Reference, user-name and password and click Activate. CompleteFTP Manager will then complete the activation automatically.
5. If the computer on which CompleteFTP Manager is running has no Internet connection then click the link at the top of the form to open the Browser-Based Activation form, then
  - a. Enter the Purchase Reference in step 1.
  - b. Click the link marked 'copy' in step 2 to copy it to the clipboard.
  - c. Send the link a computer that does have Internet access and open it.
  - d. On the web-page that opens enter your user-name and password.
  - e. Copy the Activation Key shown on the web-page to the clipboard.
  - f. Send the Activation Key to the computer that's running CompleteFTP.
  - g. Paste the Activation Key into the field in step 4.
  - h. Click OK to complete activation.

## Activating with a Universal Activation Key

- A. Run the CompleteFTP manager and connect to the CompleteFTP server instance that you wish to activate (the server instance - the CompleteFTP Windows service - may be running on another machine).
- B. Select the "Licensing" menu item from the list on the left in the manager, and then select the "Apply purchased license" link.
- C. In the 'Direct Activation' form, click the 'here' link at the top to open the 'Browser-Based Activation' form.
- D. Paste the universal AK you have been sent into the CompleteFTP manager in Step 4. Select "OK" and it should now be activated, and the license details should be displayed.

For instructions with pictures on how to activate the paid editions, please refer to Step-by-step guide: [Activate the paid editions](#).

If you need to move an instance of CompleteFTP from one server to another, or are having difficulties with activating, please open a ticket at the [EnterpriseDT Helpdesk](#).



# Universal activation keys

A *universal activation key* (UAK) can be used for activating CompleteFTP installations without needing Internet access. It is only possible to obtain a UAK in conjunction with a Volume License for at least twenty (20) servers.

Normally, activation keys are tied to a specific Windows installation. Installations need to be activated via the Internet, although there is provision for off-line activation: if the specific machine being activated does have Internet access, the activation key can be obtained using a different machine with Internet access. A UAK removes this requirement.

A UAK is not tied to a specific Windows instance, so the same activation key will work on all installations. This means it can be hard-coded into silent installations. The UAK has the name of the licensed owner encrypted in it, which is displayed in the CompleteFTP manager's title bar.

# Support

The Enterprise Distributed Technologies website offers a rich [question & answer board](#) for discussion of the CompleteFTP server. It is fully searchable and contains advice and answers to many issues that administrators have faced when using the server.

The [Frequently Asked Questions](#) should also be consulted before [contacting support](#).

# How to contact support

Customers and those trialing a product may [open a support ticket](#). Before doing so, please check our [support page](#), as your question may be answered there.

When contacting support with a problem, always include the following:

- your [purchase reference](#)
- a detailed explanation of the issue. If your description is vague, we'll have to reply to ask for more details, and that means resolving the issue will take longer.
- the [relevant diagnostics files](#)
- any relevant information or logging from a client application if it is having problems connecting. Ensure the maximum log level is being used.
- the version of CompleteFTP you are using, if that is not shown in the log file

Please include as much information as possible; it will save you the time that it takes us to ask for it.

Please check your spam folders for our replies if you haven't seen a reply from us within 24 hours.

# Frequently Asked Questions

Frequently Asked Questions are an excellent source that should always be consulted prior to contacting support

# CompleteBox revision history

Version	Description
Version 1.3.0.0 (31 March, 2022)	<ul style="list-style-type: none"><li>Added details view, similar to that in Windows File Explorer. Files can be ordered by any property. Columns can be resized and moved.</li></ul>
Version 1.2.1.0 (17 May, 2019)	<ul style="list-style-type: none"><li>Fixed error that occurs when clicking the CompleteBox icon in the system tray.</li></ul>
Version 1.2.0.0 (31 March, 2019)	<ul style="list-style-type: none"><li>Major new feature - support for password-protected sharing.</li><li>Numerous minor bug fixes.</li></ul>
Version 1.1.6.0 (1 November, 2018)	<ul style="list-style-type: none"><li>Fix errors after sleep/resume.</li></ul>
Version 1.1.5.0 (6 February, 2018)	<ul style="list-style-type: none"><li>Added support for encrypted file shares (requires CompleteFTP 11.0).</li></ul>
Version 1.1.4.0 (17 July 2017)	<ul style="list-style-type: none"><li>Minor bug fix that meant after some time CompleteBox was unresponsive.</li></ul>
Version 1.1.3.0 (14 April 2017)	<ul style="list-style-type: none"><li>Minor GUI fixes. Now highlights file just uploaded.</li></ul>
Version 1.1.2.0 (18 March 2016)	<ul style="list-style-type: none"><li>Fix issue where an attempt to share a file that is locked for reading was retrying repeatedly.</li></ul>
Version 1.1.1.0 (20 July 2015)	<ul style="list-style-type: none"><li>Fix auto copy to clipboard.</li><li>Fix failure to refresh file-list when changing user.</li><li>Fix "server copy changed" bug when server time out of synch with client.</li></ul>
Version 1.1.0.0 (7 July 2014)	<ul style="list-style-type: none"><li>Made localizable.</li><li>Fixed documentation links.</li><li>Minor bug fixes.</li></ul>
Version 1.0.0.3 (27 December 2013)	<ul style="list-style-type: none"><li>Fixed Reshare bug. If resharing after an idle period, the interface would be locked and displaying a message about retrying to connect to server.</li></ul>
Version 1.0.0.2 (12 December 2013)	<ul style="list-style-type: none"><li>Fix CompleteBox right-button click from Windows Explorer stopped working bug.</li></ul>
Version 1.0.0.1 (2 December 2013)	<ul style="list-style-type: none"><li>First release of file-sharing client CompleteBox, designed to work with <a href="#">CompleteFTP</a> server.</li></ul>

# Protocol Overview

FTP (File Transfer Protocol) is a long established Internet standard designed to transfer files (and information about files) across computer networks.

FTP is defined in the standards document known as [RFC959](#).

FTP requires a client program (an FTP client) and a server program (an FTP server). The client can fetch files and file details from the server, and also upload files to the server. The server is generally password protected.

FTP commands are initiated by the client, which opens a TCP connection called the control connection to the server. This control connection is used for the entire duration of a session between the client and server. A session typically begins when the client logs in, and ends when the quit command is sent to the server. The control connection is used exclusively for sending FTP commands and reading server replies - it is never used to transfer files.

Transient TCP connections called data connections are set up whenever data (normally a file's contents or file listings) is to be transferred. For example, the client issues a command to retrieve a file from the server via the control channel. A data connection is then established, and the file's contents transferred to the client across it. Once the transfer is complete, the data connection is closed. Meanwhile, the control connection is maintained.

See also:

[Active and Passive Modes](#)

[FTP Commands](#)

[Sample Scenarios](#)

[Data Types](#)

[Session Commands](#)

[File Commands](#)

[Directory Commands](#)

# Active and Passive Modes

Data connections may be set up in two different ways, active and passive. Note that active and passive refer to the operation of the FTP server, not the client.

## Passive mode

In passive mode, the client sends a PASV command to the server. This tells the server to listen for a connection attempt from the client, hence the server is passively waiting. The server replies to PASV with the host and port address that the server is listening on. The client deciphers this reply and when a data connection is required, attempts to initiate the connection to the server at this address.

## Active mode

In active mode, the server actively connects to the client. To set up active mode, the client sends a PORT command to the server, specifying the address and port number the client is listening on. When a data connection is required, the server initiates a connection to the client at this address.

Generally the server is responsible for closing data connections.

Next: [FTP Commands](#)

# FTP Commands

FTP commands sent across the control connection consist of simple text strings (and follow the Telnet protocol - see RFC 854). For example, to retrieve a file, the client sends "RETI filename" on the control connection to the FTP server. To transfer a file, the client sends "STOR filename".

The FTP server acknowledges each command with an FTP reply, which consists of a threedigit number followed by human-readable text. The first digit indicates if the response is good, bad, or incomplete. If an error occurred, the second digit may be used to indicate what type of error occurred. Similarly, the third digit can indicate more details of the error.

The first digit is the most important, and the five possible values are described below:

Reply	Description
1yz	Positive Preliminary reply. The request action has been initiated, but another reply is to be expected before the client issues another command.
2yz	Positive Completion reply. The requested action has successfully completed, and the client may issue another command.
3yz	Positive Intermediate reply. The command has been accepted, but more information is required. The client should send another command in reply.
4yz	Transient Negative reply. The command failed, but it can be retried.
5yz	Permanent Negative Completion reply. The command failed, and should not be repeated.

Next: [Sample Scenarios](#)



# Sample Scenarios

## Example 1

For example, to change directory the client sends:

```
CWD dirname
```

The server responds with:

```
250 CWD command successful
```

As the reply begins with a ‘2’, we know the command sequence is completed.

However if we attempt to change directory to one that does not exist:

```
CWD nonexistentdir
```

The server responds with:

```
550 nonexistentdir: The system cannot find the file specified.
```

As the reply begins with a ‘5’ we know that the command failed.

## Example 2

To transfer a text file, we issue a ‘RETR’ command to the server. However to transfer the file we require a data connection to be set up. This can be done using active or passive mode.

As discussed previously, in active mode, the client sends a ‘PORT’ command, indicating what address and port number the server should connect to:

```
PORT 192,168,10,1,4,93
```

The first four digits are the IP address, and the last two encode the port number (in two 8-bit fields, the first being the high order bits of the 16-bit port number).

The server responds with:

```
200 PORT command successful.
```

This indicates that the data connection has been established.

Next, the ‘RETR’ command is issued:

```
RETR abc.txt
```

The server responds with:

```
150 Opening ASCII mode data connection for abc.txt(70776 bytes).
```

The reply begins with a ‘1’, so we know that the command was successful, but the server will be sending another reply – the client cannot issue another command until this is received.

Eventually, the server sends:

```
226 Transfer complete.
```

The command sequence is complete, the file has been transferred, and the client can issue another command.

See [RFC959](#) for details about the second digit, and more extensive examples.

Note that most standard command-line FTP clients support debug mode, which displays the FTP commands that are being sent to the server, and the reply strings that are received back. Typing “debug” at the prompt will usually put the client into debug mode.

Next: [Data Types](#)

# Data Types

The two most common data types in usage are ASCII and binary.

ASCII is the default data type, and is intended for the transfer of text files. A line of text is followed by <CRLF>. Note that different operating systems use different end of line terminators.

Binary type (known as IMAGE in FTP) is used to transfer binary files. A byte-by-byte copy is made of the source file. Graphical images, video and executable files are all binary files. If they are transferred as ASCII, they will be corrupted.

Next: [Session Commands](#)

# Session commands

To begin an FTP session, the USER command is sent to the server:

```
> USER ftpuser
```

The server responds with:

```
331 Password required for ftpuser.
```

The client must respond with the password:

```
> PASS mypassword
```

The server responds with:

```
230 User ftpuser logged in.
```

The session is now established, and the user can begin issuing various file and directory- related commands.

To end the session, the client sends:

```
> QUIT
```

The server responds with:

```
221
```

The session is now closed, and any further attempt to send commands to the server will fail.

Next: [File Commands](#)

# File Commands

FTP supports numerous file-related commands.

Files can be deleted (DELE) and renamed (RNFR,RNTO) as well as stored (STOR) and retrieved (RETR). When a file is stored, it can be written over or appended to (APPE).

See the [Sample scenarios examples](#) for more details.

Next: [Directory Commands](#)

# Directory Commands

FTP supports a variety of directory-related commands.

Directories can be created (MKD), removed (RMD), or changed into (CWD, CDUP).

Two types of directory listings can be made with FTP.

The LIST method obtains a list of files (and possibly directories). If a directory is specified, the server returns a list of files in the directory, together with system specific information about the files. The file information sent will vary depending on the server system. The data type should be set to ASCII for this file name list. If no directory is specified, details of the current working directory listing are sent.

The NAME LIST (NLST) method is similar to LIST, but only file names are returned. No other information about the files is sent. Again, the data type should be set to ASCII

Next: [FTPS - Securing FTP with TLS](#)

# FTPS - Securing FTP with TLS

A significant problem with "plain" FTP is that it is not secure - usernames, passwords and data are sent across the network in the clear. This means eavesdroppers sniffing the network will have little trouble obtaining credentials and copies of the transferred files.

FTPS was designed to fix this by encrypting communications using the SSL/TLS protocol, which is specifically designed for securing network connections.

FTPS is an IETF standard as described in [RFC 4217](#).

See also:

- [Implicit FTPS and Explicit FTPS](#)
- [Securing Control and Data Channels](#)
- [FTPS Commands](#)
- [FTPS Usage](#)

# Implicit FTPS and Explicit FTPS

An early version of FTPS is now known as *implicit FTPS*. Upon connecting, implicit FTPS clients automatically start securing the connection using SSL/TLS.

This is a problem for unencrypted FTP clients - they will no longer be able to connect on a server port that requires securing the connection immediately. If FTP and FTPS (implicit) are to be supported on the same server, they require different port numbers. Typically, implicit FTPS uses port 990 rather than the standard FTP port 21.

Explicit FTPS fixed this by requiring the AUTH command to be sent by the client prior to securing the connection. This meant that unencrypted FTP clients can connect on the same port as FTPS clients - the unencrypted client simply never send an AUTH command and the session remains unencrypted. FTPS clients send the AUTH command prior to logging in so that credentials are secured.

Implicit FTPS clients can still be found, but explicit mode is always preferred. Unless implicit mode is required it is best to have implicit mode FTPS disabled in CompleteFTP.

Next: [Securing Control and Data Channels](#)

# Securing Control and Data Channels

FTP sessions use two channels: *control* and *data*. Only one control-channel is used in each session, but several data-channels may be used – one for each data transfer. The AUTH command only secures the control-channel. Data-channels are not secured until PBSZ and PROT commands are issued. These commands tell the server whether or not subsequent data-channels should be secure.

Clients can connect to FTPS servers in unencrypted mode, and then switch to secure mode upon request. To do so, the client issues an AUTH command, upon which the client and the server negotiate a secure connection. After the switch, all FTP commands are encrypted, but importantly, data is not encrypted unless further commands are provided.

Next: [FTPS Commands](#)



# FTPS Commands

Three commands are used, AUTH, PBSZ, and PROT. One of these, PBSZ, appears to be redundant and is probably included only to satisfying the RFC specification.

## AUTH (AUTHentication)

The AUTH command takes a single parameter which defines the security mechanism to be used, usually 'SSL' or 'TLS'.

```
AUTH TLS
```

With this command, an attempt is made to negotiate a TLS connection on the control channel. The server tries to validate itself to the client by sending its certificate (Server Validation). It may also involve the client sending its certificate to the server (Client Validation).

## PBSZ Protection Buffer SiZe)

The PBSZ command is intended to define the buffer-size to be used by the security mechanism when it is encrypting data on the data-channel. However for TLS this setting is redundant and a value of '0' is always passed as a parameter.

```
PBSZ 0
```

While this call is redundant, it is required and must precede the PROT command.

## PROT (data channel PROTECTION level)

PROT defines whether or not the data channel is to be protected or not. Either the data channel is *Clear* (the default), or *Private*. Clear means that no security is used on the data-channel (meaning files are transmitted without encryption), and Private means that the data-channel should be encrypted. So there are two possible PROT commands:

```
PROT C
```

for an unsecured data-channel, and

```
PROT P
```

for an encrypted data-channel.

Next: [FTPS Usage](#)

# FTPS Usage

A typical explicit FTPS session might consist of the following sequence of commands:

```
> USER (user-name)
Provide user-name
> PASS (password)
Provide password
> LIST
Get a directory listing
> AUTH TLS
Switch to TLS on control-channel
> RETR (file-name)
Download a file (without security)
> PBSZ 0
> PROT P
Switch to TLS on the data-channel
> STOR (file-name)
Upload a file (with security)
> QUIT
End session
```

In this example, the first three commands (USER, PASS, and LIST) are standard FTP and therefore insecure. The AUTH command causes the rest of the commands to be sent to the server securely, in other words, an attacker cannot see which commands are issued. The RETR command (to fetch a file from the server), being after AUTH, is protected, but the actual file that is transferred is not protected since it precedes the PBSZ and PROT commands. PBSZ and PROT tell the server to use TLS on all future data-channels, thus the file transferred in the STOR command (which stores a file on the server) is secure.

## Rules

There are two rules regarding the issuing of explicit FTPS commands that must be followed:

- 1. AUTH must precede PBSZ
- 2. PBSZ must precede PROT

Apart from these, a FTPS server has policies regarding access permissions to its resources. These policies will also determine the order in which commands must be issued. There are too many possible policies to list here, but a few examples of such policies are given below along with their consequences in terms of the issuing of commands.

Policy	Consequences
<ul style="list-style-type: none"><li>• No unprotected commands</li><li>• Certain users are not permitted to log in without security.</li></ul>	<p>AUTH must be issued before any other commands.</p> <p>The USER command is rejected for particular users unless preceded by a successful AUTH command.</p>
<ul style="list-style-type: none"><li>• No unprotected data may be transferred</li></ul>	<p>A 'PROT P' command (preceded by a PBSZ command) must be issued before any files are transferred.</p>
<ul style="list-style-type: none"><li>• Allow TLS authentication instead of USER/PASS authentication</li></ul>	<p>A client certificate must be supplied and USER/PASS commands are not required.</p>

Next: [The Essentials of FTP Security](#)

# The Essentials of FTP Security

While users of FTPS may safely remain ignorant of most of the underlying security mechanisms, it is necessary that they understand the basics of public keys and certificates. In some cases, it is also necessary to understand which encryption/decryption algorithms (or *ciphers*) are available and what level of protection they offer.

See also:

- [Public Key Cryptography](#)
- [Certificates and Certificate Authorities \(CAs\)](#)
- [Obtaining Keys and Certificates](#)
- [Server and Client Validation](#)
- [Hostname Checking](#)
- [Selecting Ciphers](#)

# Public Key Cryptography

Public Key Cryptography (PKC) is a paradigm which uses a pair of keys in a given communication; one key is used for encrypting the message and the other key is used for decrypting it. Each key may serve either purpose, but a message encrypted using one key may ONLY be decrypted by the other key.

The following illustration demonstrates how such a pair of keys may be employed for secure communication:

*Party A has generated a key-pair. They retain one key, the **private key**, and distribute the other key, the **public key**, to Party B in a trustworthy way (see Section 4.1.2).*

*1. A encrypts a message using the private key and sends it to B. If B is able to decrypt the message using the public key, then B may be confident that the message did in fact originate from A, since only A has the private key.*

*2. B encrypts a message using the public key and sends it to A. Since the message may only be decrypted using the private key, and only A has this key, B may be confident that only A will be able to read the message.*

Thus, using A's private/public pair of keys, B can ensure that (1) A is who they purport to be, and (2) Any messages that are sent to A can be read by that party only. However, the following weaknesses remain: (1) A cannot be confident that B is who B purports to be, and (2) Communications from A to B may be read by anyone who has the public key.

While both of these weaknesses may easily be overcome if B has their own key-pair and has provided the public key to A, this is often not practical due to the amount of effort involved. However, the latter shortcoming may easily be overcome in the following way:

*3. B automatically generates a temporary key-pair. Since B may be confident that its messages to A are only read by A, B can securely provide one of the keys to A. Once A has received this key, they may use it to encrypt any messages sent to B. They may therefore be confident that only B can read any subsequent messages.*

Thus, a single private-public key-pair has the potential to offer the following security:

I. Parties receiving messages from the owner of the key-pair can verify that encoded messages originated from the owner.

II. Secure messages may be sent between the owner and other parties.

As mentioned, this presumes that the owner of the key-pair is able to distribute their public keys in a trustworthy manner. In practice, this is achieved by means of public key certificates and Certificate Authorities.

Next: [Certificates and Certificate Authorities \(CAs\)](#)

# Certificates and Certificate Authorities (CAs)

The previous section emphasized the need for public keys to be distributed in a trustworthy manner. Simply sending the key electronically should not be considered safe as the message might be tampered with on the way, though this is done in many cases. Having one person physically hand another a memory-stick containing the key would be probably acceptable, but is usually impractical. The solution used on the Internet is to use a trusted third party called a Certificate Authority (CA).

A CA is an organization which specializes in issuing public key certificates. They only issue certificates to parties (or *subjects*) after they have provided sufficient documentary evidence of their identity. There are only a few CAs in the world and, since their viability relies on their trustworthiness, they can usually be relied on to do a good job of validating their subjects.

Each CA has its own private-public key-pair and its own certificate (called a *root certificate*). Since there are so few CAs and since they rarely change their keys, it is feasible for software to be distributed with a list of the certificates of all existing CAs. For example, Microsoft's and Netscape's browsers are both distributed with files containing lists of root certificates.

A certificate issued to a subject by a CA contains:

- the identification of the subject,
- the public key of the subject,
- the identification of the CA which issued the certificate, and
- the digital signature, provided by the CA, which verifies the contents of the message.

A subject's certificate may be validated in the following way:

1. Use the identification of the CA to find the appropriate root certificate.
2. Use that root certificate to check the digital signature and thereby prove that the CA issued the certificate and that the information in the certificate has not been tampered with.
3. Verify the identification of the subject.

Once this has been done, the party can trust that the public key in the certificate is indeed the public key that they expected. This public key may henceforth be used to establish secure communications with the subject in the manner described in Section 12.

Note that although the CA has signed a certificate containing the public key of the subject, the CA does not have the subject's private key. To issue a certificate, the CA only requires access to the public key. The CA obtains this public key from the subject applying for a certificate. It is therefore up to the subject to generate the private-public key-pair before applying to a CA for a certificate.

Next: [Obtaining Keys and Certificates](#)

# Obtaining Keys and Certificates

A private-public key-pair is easy to generate. Likewise, an “uncertified” certificate may be generated with little effort, but obtaining a trustworthy certificate from a CA necessarily requires some work, time, and (usually) cost. It involves interacting with the CA to prove identity, and waiting for the CA to digitally sign the certificate.

Alternatively OpenSSL may be used. The reader is referred to the [OpenSSL Key HOWTO](#) for instructions on generating key-pairs and to the [OpenSSL Certificate HOWTO](#) for generating certificates. It is recommended that keys with a length of at least 768 bits are used. OpenSSL may be obtained from <http://www.openssl.org/source/> or <http://www.openssl.org/related/binaries.html>.

Next: [Server and Client Validation](#)

# Server and Client Validation

The previous sections described how public key certificates may be used to validate parties involved in secure communications. They also explained why some work, time, and cost is involved in obtaining certificates.

The nature of Internet usage is such that it is important to distinguish between server validation and client validation. Server validation allows a client to know that it is talking to the intended server. Conversely, client validation allows a server to know that it is talking to the intended client.

It is often more important for a client to know that it is talking to the intended server than the converse. The reason for this is often financial. For example, if a client is purchasing an item from an online retailer, the client needs to be certain that their credit card details are going to the intended destination. While it might be nice for the retailer to be certain where the money is coming from, it is not usually essential. Therefore, server validation is nearly always used in such transactions, but client validation is less often used. Other applications, such as Internet banking often use both client and server validation.

In FTPS, both server and client validation by certificate are optional. Though the server's certificate is always sent, it is up to the client whether or not it validates the certificate. It is up to the client whether or not it will try to validate itself to the server, but some servers have a policy of not allowing unvalidated clients to access some or all its resources.

It is important to note that, although many FTPS servers don't request client certificates, most require a user-name and password to be sent. If these are sent over a secure control channel then a reasonable level of client validation is inherent.

Next: [Hostname Checking](#)

# Hostname Checking

One of the steps in server validation is host-name checking. Host-name checking is a simple check that is performed when a secure connection is being established. It involves comparing the following two items:

- the host-name that is (usually) contained in the certificate which the FTPS server is presenting
- the host-name of the machine on which the FTPS server is running

If they match then one can be confident that the server to which the client is connected is in fact the server to which the certificate was issued. If they do not match, then there's a possibility that the certificate has been stolen and that the server, to which the client is connected, is attempting to "impersonate" the actual server to which the client is actually connected. This is a form of "man-in-the-middle" attack, which gives the attacker complete control over the data being sent and received.

Unfortunately, the most widely compatible version of the X.509 certificate standard does not specify exactly how a host-name should be defined within a server certificate. The convention is that the Common Name (CN) field of the certificate should be used, and, while this is followed by the majority of Certificate Authorities (CAs), it is not universal.

If it is possible to configure the FTPS server's certificate then the Common Name (CN) field of the certificate must be the same as the host-name of the machine on which the FTPS server is running.

Disabling host-name checking is strongly discouraged and should only be done as a last resort if the FTPS server's certificate cannot be configured so that its CN parameter contains its host-name.

Next: [Selecting Ciphers](#)



# Selecting Ciphers

SSL and TLS can use a variety of encryption/decryption algorithms (called *ciphers*). In a single secure connection, as many as four different ciphers may be used for various purposes; this set of ciphers is called a *cipher suite*. Each party in a secure connection must designate which cipher suites it is going to support. When a new secure connection is made, the parties involved try to agree on which cipher suite to use. There must be at least one cipher suite that is available on both sides of the connection for this to be possible.

Different cipher suites have different levels of security and performance. The lower the level of security, the easier the cipher is to break. Unfortunately, stronger ciphers usually offer slower performance. Hence, there is a certain level of trade-off between the two. For this reason, the decision on which cipher suites to support is left to the developers and/or users of SSL and TLS applications.

Every cipher suite has a standard name (e.g. TLS\_RSA\_WITH\_RC4\_128\_SHA). This name reveals which ciphers are used in the suite.

Some guidelines which may be useful when selecting cipher suites are:

- Triple DES offers high security but relatively poor performance (look for names with the characters, 3DES).
- DES (not 3DES) can be cracked relatively easily
- 128-bit RC4 offers high performance and reasonable security (look for names with the characters RC4\_128).
- SHA is preferable to MD5 (choose algorithms whose names ending with SHA ahead of those ending with MD5).
- Export version of algorithms are deliberately weakened; avoid them if possible (i.e. avoid 40-bit suites).

# SFTP and SCP

SFTP is an abbreviation for SSH File Transfer Protocol, and is exactly that - a protocol for transferring files over an SSH connection.

SFTP is **not** the standard FTP protocol running over SSH. Although SFTP has similar capabilities and even similar commands to standard FTP, these similarities are purely superficial. The protocol is completely different and incompatible with FTP and its secure extension, FTPS.

SCP is also a file transfer protocol that runs over SSH connections. It is a precursor to SFTP, and allows the copying of files and directories over SSH.

In order to understand SFTP and SCP, it is helpful to have a basic understanding of SSH.

See also:

- [SSH - Secure Shell](#)
- [SCP - Secure Copy](#)
- [SFTP - SSH File Transfer Protocol](#)
- [Comparison of FTPS and SFTP](#)

# SSH - Secure Shell

SSH is a standard designed to allow logging in and execution of commands on a remote computer in a manner similar to telnet, rlogin, and rsh. Unlike these protocols, it does this through an encrypted network connection thus offering a much higher level of security.

The first version of the standard, SSH-1, was designed in 1995 by Tatu Ylönen. The second version, SSH-2, is being standardized by the IETF SECSH working group. It offers a higher level of security than its predecessor.

In order for a computer to be able to accept SSH connections, it must be running an SSH server, such as *sshd*, on a publicly accessible port (usually port 22). The client computer must have an SSH client, such as *CompleteFTP*, and be known to the server.

Private/public key-pairs in SSH typically use either the DSA or RSA asymmetric key algorithms. Most SSH servers support both.

Clients perform server validation in SSH via a known hosts file. The client maintains a file containing the hostname (or IP address) of the SSH server, together with the server's public key. When clients connect to the server, they are sent a copy of the server's public key which they can compare with their own record of the server's public key.

The server authenticates clients who connect to it. The client must be previously set up as an SSH user (or in some cases as a user on the server machine). In password authentication, the client supplies their password which the server authenticates as belonging to that user. In public key authentication, the client uses its private key to sign some data, and sends the signature to the server. The server uses the client's public key to verify the signature. In this case the client's public key must be available on the server.

# SCP - Secure Copy

Since the early days of SSH, file transfers have been supported through a command called SCP. This command simply securely copies files or directories between remote computers. It provides no other file operations such as listing, deleting, renaming, and directory navigation.

SCP itself does not provide authentication and security - it relies on the underlying protocol, usually SSH.

CompleteFTP offers full support for SCP.

# SFTP – SSH File Transfer Protocol

The SFTP protocol was developed as a solution to the limitations of **SCP**. It provides all the operations that have come to be expected with FTP, and is generally preferable.

SFTP does not provide authentication and security itself - it relies on the underlying protocol, usually SSH. The two protocols are commonly used together - so much so that SFTP generally means that it is running over SSH.

CompleteFTP offers full support for SFTP.

# Comparison of FTPS and SFTP

While FTPS and SFTP are completely different protocols, they offer the same basic feature -secure file transfers. It is therefore common to be faced with the choice of one or the other. This section provides some pros and cons of these two protocols.

## Security

Under ideal conditions SFTP and FTPS are able to offer comparable levels of security, but many SFTP deployments suffer from a vulnerability that is an artifact of SFTP's close relationship with SSH. The problem arises when you want to allow client SFTP access on a server but not SSH access. This is generally not a problem for pure SFTP servers (such as CompleteFTP), but for SSH/SFTP servers such as OpenSSH it can be quite complex and error-prone. So if you are not very careful when you set up your servers, users on machines with the SFTP client installed will be able to use an SSH client to log into the server and execute commands. This is not a problem with FTPS since this is purely a file transfer protocol and not a remote console protocol.

## Upgrading

FTPS is a straight-forward extension to an existing FTP infrastructure. It is supported by most commercial servers and many open source servers (e.g. wu-ftpd and proftpd), so enabling FTPS on a server is usually just a matter of adding a few configuration options. There is no need to run additional servers since FTPS servers invariably also support FTP. There is also no need to open additional ports in firewalls since FTPS uses the same ports as FTP. It is important to note that data-transfer problems can sometimes arise when changing from FTP to FTPS - see "Firewalls" section below.

## Certificates

SFTP uses keys rather than certificates. This means that it can't take advantage of the "chains of trust" paradigm facilitated through Certificate Authorities. This paradigm makes it possible for two entities to establish a trust relationship without directly exchanging security information, which is important for some applications. FTPS uses certificates and therefore can take advantage of this paradigm. SFTP clients must install keys on the server.

## Firewalls

SFTP often works better through some firewalls since it does not rely on multiple connections like FTP does. As explained in an earlier chapter, FTP and FTPS both use a control channel to send commands, and a new data connection for each file transfer. While the control channel is usually easily connected, it is common to experience firewall-related problems when connecting data-channels. This is particularly so in FTPS where the FTP-specific features of most firewalls are ineffective due to encryption. Since SFTP relies on a single network connection, it does not suffer from these problems.

# HTTP Protocol Overview

HTTP (HyperText Transfer Protocol) is a well established Internet protocol designed to transfer content(e.g. HTML files and images) across networks. HTTP is defined in numerous RFCs (e.g. RFC 2616), which can be obtained from the [Internet Engineering Task Force](#).

HTTP is a request-response protocol. A client such as a Web browser sends an HTTP request to an HTTP server, which is hosting content such as HTML files and images. The server sends a response message to the client, which will generally include the content the client requested if it is available. The content (or resources) hosted by the HTTP server is identified by Uniform Resource Locators, or URLs, using the http or https notation.

The original version of HTTP (HTTP/1.0) used a separate TCP connection to the server for every request-response. This meant that HTML pages with images required a connection for the page as well as for each image. This was not easily scalable, and HTTP/1.1 can reuse connections multiple times.

See also:

- [HTTP methods](#)
- [HTTP sessions](#)
- [HTTPS](#)

# HTTP methods

HTTP defines various request methods that clients can send to HTTP servers. Generally they involve performing an action on the specified resource. The most commonly used request methods are listed below:

Method	Description
GET	Retrieve the specified resource (usually an HTML page or file).
HEAD	Same as GET, except only retrieve the response header, not the body (i.e. the contents of the requested resource).
POST	Submit data to be processed. The data is in the body of the request, and usually comes from an HTML form.

Next: [HTTP sessions](#)



# HTTP sessions

HTTP is a stateless protocol. This means the server is not required to maintain state for a user in between a user's HTTP requests. One common solution is to use HTTP cookies, which are text strings stored by a user's web browser. Typically, a session identifier provided by an HTTP server is stored in a cookie. This cookie is then sent with every request to the server, and the server is able to match the session id with stored state for the user. Session identifiers normally expire within a few minutes, hours or days, depending on the server settings.

Next: [HTTPS](#)

# HTTPS

HTTP is not a secure protocol, in the same way that FTP is not a secure protocol - all data is transmitted unencrypted, including usernames and passwords. The HTTPS (HyperText Transfer Protocol Secure) protocol was introduced to provide a way to securely send data via HTTP. It is identical in syntax, but uses SSL/TLS to encrypt the data being sent to and from the server.

HTTPS is especially useful for encrypting usernames and passwords that are being sent to an HTTP server, as well as sensitive individual and corporate data such as credit cards or other payment transactions.

# IETF Standards

- [RFC 959 - File transfer protocol \(FTP\)](#)
- [RFC 2228 - FTP security extensions](#)
- [RFC 4217 - Securing FTP with TLS](#)
- [RFC 4253 - The Secure Shell \(SSH\) Transport Layer Protocol](#)
- [SSH file transfer protocol](#)

# About this manual

Enterprise Distributed Technologies would like to thank Golmote, Jannik Zschiesche and the rest of the PrismJS team for developing and maintaining PrismJS which we use for syntax highlighting.

# CompleteFTP Custom Configurer: cftpconfig

The cftpconfig tool may be used for modifying and querying the CompleteFTP configuration database (config.db). Whether run with or without options the tool will always update config.db (schema and/or data) to the latest version of CompleteFTP (unless it's already up-to-date).

By default cftpconfig operates on the configuration file at the default location (i.e. C:\ProgramData\Enterprise Distributed Technologies\Complete FTP\config.db), but this can be changed using the options.

Please note that as cftpconfig operates directly on the configuration file, the CompleteFTP service will need to be restarted to reload the configuration. Until this is done changes will not be able to be seen in the CompleteFTP manager. Normally this tool should only be used when the service is stopped.

### Available options:

/c [file]	Customize installation using SQL commands in this file or read from stdin if path is absent or '-' (ctrl+Z to mark end of input).
/f file	Configure the given configuration file instead of the configuration file at the default location
/h	Show help
/i file	Import the configuration in the given file instead of upgrading the configuration at the default location
/o	Create a new configuration file rather than upgrading the existing config.db. All existing configuration data will be overwritten.
/p password	Set administrator password to this
/q query	Execute this SQL query (INSERT, UPDATE, DELETE or SELECT) on the configuration. Output generated by SELECT statements is tab-delimited.
/r	Generate (and write) random administrator password
/v	Verbose - write status messages to console

The schema of the CompleteFTP configuration database is available [upon request](#). The SQL syntax adds the UPDATE-FROM commandThe syntax of the UPDATE-FROM command is

```
UPDATE TableA SET FieldA1=AliasB.FieldB1 WHERE FieldA2=value1
FROM TableB AliasB WHERE FieldB2=value2
```

A tool such as [DB Browser for SQLite](#) may be useful in viewing your config.db file.

# CompleteFTP third-party copyright notices

CompleteFTP uses free software packages from a number of different third-party sources, and their copyright notices are included below.

## Mentalis SSL/TLS

by *The KPD-Team*

Copyright (c) 2002-2003, The KPD-Team  
All rights reserved.  
<http://www.mentalis.org/>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Neither the name of the KPD-Team, nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2002-2003, The KPD-Team

## BigInteger

by *Chew Keong TAN*

BigInteger Class Version 1.03

Copyright (c) 2002 Chew Keong TAN  
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Disclaimer

-----  
Although reasonable care has been taken to ensure the correctness of this implementation, this code should never be used in any application without proper verification and testing. I disclaim all liability and responsibility to any person or entity with respect to any loss or damage caused, or alleged to be caused, directly or indirectly, by the use of this BigInteger class.

Comments, bugs and suggestions to  
(<http://www.codeproject.com/csharp/biginteger.asp>)

## Granados SSH Library for .NET

by *Routrek Networks, Inc.*  
*Chew Keong TAN*

Copyright (c) 2003 Routrek Networks, Inc. All Rights Reserved.  
Copyright (c) 2002 Chew Keong TAN

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by Routrek Networks, Inc." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Routrek Networks" or "Routrek" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact [info@routrek.co.jp](mailto:info@routrek.co.jp).

5. Products derived from this software may not be called "Granados", appear in their name, without prior written permission of the Routrek Networks, Inc.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ROUTREK NETWORKS, INC. OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## **ZLIB.NET**

by *ComponentAce*

Copyright (c) 2006, ComponentAce  
<http://www.componentace.com>  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of ComponentAce nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## **FastColoredTextBox**

by *Pavel Torgashov*

THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE.

License: GNU Lesser General Public License (LGPLv3)

Email: [pavel\\_torgashov@ukr.net](mailto:pavel_torgashov@ukr.net).

Copyright (C) Pavel Torgashov, 2011-2014.

## **CryptoConvert and PrivateKey**

by *Motus Technologies Inc.*  
*Novell, Inc*

PrivateKey.cs - Private Key (PVK) Format Implementation

Author:  
[Sebastien Pouliot sebastien@ximian.com](mailto:Sebastien Pouliot sebastien@ximian.com)

(C) 2003 Motus Technologies Inc. (<http://www.motus.com>)  
Copyright (C) 2004 Novell, Inc (<http://www.novell.com>)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the

Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## **elFinder**

by *Studio 42*

Copyright (c) 2009-2012, Studio 42

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of the Studio 42 Ltd. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL "STUDIO 42" BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## **JsonSerializer**

by *The Outercurve Foundation*

The MIT License (MIT)

Copyright (c) 2011, The Outercurve Foundation.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## **OIOSAML.NET**

by *Digitaliseringsstyrelsen*

## **Bouncy Castle APIs**

by *The Legion of the Bouncy Castle Inc.*

Copyright (c) 2000 - 2018 The Legion of the Bouncy Castle Inc. (<https://www.bouncycastle.org>)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



**ngx-admin**

by *akveo.com*

The MIT License (MIT)

Copyright (c) 2017 akveo.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## NAME

**mustache** -- Logic-less templates.

## SYNOPSIS

A typical Mustache template:

```
Hello {{name}}
You have just won ${{value}}!
{{#in_ca}}
Well, ${{taxed_value}}, after taxes.
{{/in_ca}}
```

Given the following hash:

```
{
  "name": "Chris",
  "value": 10000,
  "taxed_value": 10000 - (10000 * 0.4),
  "in_ca": true
}
```

Will produce the following:

```
Hello Chris
You have just won $10000!
Well, $6000.0, after taxes.
```

## DESCRIPTION

Mustache can be used for HTML, config files, source code - anything. It works by expanding tags in a template using values provided in a hash or object.

We call it "logic-less" because there are no if statements, else clauses, or for loops. Instead there are only tags. Some tags are replaced with a value, some nothing, and others a series of values. This document explains the different types of Mustache tags.

## TAG TYPES

Tags are indicated by the double mustaches. **{{person}}** is a tag, as is **{{#person}}**. In both examples, we'd refer to **person** as the key or tag key. Let's talk about the different types of tags.

### Variables

The most basic tag type is the variable. A **{{name}}** tag in a basic template will try to find the **name** key in the current context. If there is no **name** key, nothing will be rendered.

All variables are HTML escaped by default. If you want to return unescaped HTML, use the triple mustache: **{{{name}}}**.

You can also use **&** to unescape a variable: **{{& name}}**. This may be useful when changing delimiters (see "Set Delimiter" below).

By default a variable "miss" returns an empty string. This can usually be configured in your Mustache library. The Ruby version of Mustache supports raising an exception in this situation, for instance.

Template:

```
* {{name}}
* {{age}}
* {{company}}
* {{{company}}}
```

Hash:

```
{
  "name": "Chris",
  "company": "<b>GitHub</b>"
}
```

Output:

```
* Chris
*
* &lt;b&gt;GitHub&lt;/b&gt;
* <b>GitHub</b>
```

### Sections

Sections render blocks of text one or more times, depending on the value of the key in the current context.

A section begins with a pound and ends with a slash. That is, **{{#person}}** begins a "person" section while **{{/person}}**

ends it.

The behavior of the section is determined by the value of the key.

### False Values or Empty Lists

If the **person** key exists and has a value of false or an empty list, the HTML between the pound and slash will not be displayed.

Template:

```
Shown.
{{#nothin}}
  Never shown!
{{/nothin}}
```

Hash:

```
{
  "person": true,
}
```

Output:

```
Shown.
```

### Non-Empty Lists

If the **person** key exists and has a non-false value, the HTML between the pound and slash will be rendered and displayed one or more times.

When the value is a non-empty list, the text in the block will be displayed once for each item in the list. The context of the block will be set to the current item for each iteration. In this way we can loop over collections.

Template:

```
{{#repo}}
  <b>{{name}}</b>
{{/repo}}
```

Hash:

```
{
  "repo": [
    { "name": "resque" },
    { "name": "hub" },
    { "name": "rip" },
  ]
}
```

Output:

```
<b>resque</b>
<b>hub</b>
<b>rip</b>
```

### Lambdas

When the value is a callable object, such as a function or lambda, the object will be invoked and passed the block of text. The text passed is the literal block, unrendered. **{{tags}}** will not have been expanded - the lambda should do that on its own. In this way you can implement filters or caching.

Template:

```
{{#wrapped}}
  {{name}} is awesome.
{{/wrapped}}
```

Hash:

```
{
  "name": "Willy",
  "wrapped": function() {
    return function(text) {
      return "<b>" + render(text) + "</b>"
    }
  }
}
```

Output:

```
<b>Willy is awesome.</b>
```

### Non-False Values

When the value is non-false but not a list, it will be used as the context for a single rendering of the block.

Template:

```
{{#person?}}  
  Hi {{name}}!  
{{/person?}}
```

Hash:

```
{  
  "person?": { "name": "Jon" }  
}
```

Output:

```
Hi Jon!
```

### Inverted Sections

An inverted section begins with a caret (hat) and ends with a slash. That is `{{^person}}` begins a "person" inverted section while `{{/person}}` ends it.

While sections can be used to render text one or more times based on the value of the key, inverted sections may render text once based on the inverse value of the key. That is, they will be rendered if the key doesn't exist, is false, or is an empty list.

Template:

```
{{#repo}}  
<b>{{name}}</b>  
{{/repo}}  
{{^repo}}  
  No repos :(  
{{/repo}}
```

Hash:

```
{  
  "repo": []  
}
```

Output:

```
No repos :(
```

### Comments

Comments begin with a bang and are ignored. The following template:

```
<h1>Today{{! ignore me }}.</h1>
```

Will render as follows:

```
<h1>Today.</h1>
```

Comments may contain newlines.

### Partials

Partials begin with a greater than sign, like `{{> box}}`.

Partials are rendered at runtime (as opposed to compile time), so recursive partials are possible. Just avoid infinite loops.

They also inherit the calling context. Whereas in ERB you may have this:

```
<%= partial :next_more, :start => start, :size => size %>
```

Mustache requires only this:

```
{{> next_more}}
```

Why? Because the `next_more.mustache` file will inherit the `size` and `start` methods from the calling context.

In this way you may want to think of partials as includes, or template expansion, even though it's not literally true.

For example, this template and partial:

```
base.mustache:
<h2>Names</h2>
{{#names}}
  {{> user}}
{{/names}}

user.mustache:
<strong>{{name}}</strong>
```

Can be thought of as a single, expanded template:

```
<h2>Names</h2>
{{#names}}
  <strong>{{name}}</strong>
{{/names}}
```

### Set Delimiter

Set Delimiter tags start with an equal sign and change the tag delimiters from `{{` and `}}` to custom strings.

Consider the following contrived example:

```
* {{default_tags}}
{{=<% %>=}}
* <% erb_style_tags %>
<%={{ }}=%>
* {{ default_tags_again }}
```

Here we have a list with three items. The first item uses the default tag style, the second uses erb style as defined by the Set Delimiter tag, and the third returns to the default style after yet another Set Delimiter declaration.

According to [ctemplates](#), this "is useful for languages like TeX, where double-braces may occur in the text and are awkward to use for markup."

Custom delimiters may not contain whitespace or the equals sign.

### COPYRIGHT

Mustache is Copyright (C) 2009 Chris Wanstrath

Original CTemplate by Google

### SEE ALSO

`mustache(1)`, `mustache(7)`, <http://mustache.github.com/>

# How to activate an installation of CompleteFTP

When a production license has been purchased, the trial installation must be *activated*.

This usually involves accessing the *CompleteFTP Activation System* on the Internet, although this can be done from another machine if the licensed machine does not have Internet access. Note that some Volume Licenses provide access to a **universal activation key** which does not require Internet access.

The following steps should be followed if you do **not** have a universal AK, which will be the majority of users.

- A. Run the CompleteFTP manager and connect to the CompleteFTP server instance that you wish to activate (the server instance - the CompleteFTP Windows service - may be running on another machine). You should run the manager on a machine with internet access (although there is a work around) if you do not have a universal activation key.
- B. Select the "Licensing" menu item from the list on the left in the manager, and then select the "Activate your license" link.
- C. Enter your purchase reference in the edit box indicated (Step 1), and then open the supplied URL. You will need the login details provided in your purchase email. If the credentials and purchase reference are correct, you will see a web page called "CompleteFTP Activation System".
- D. Your activation key should be displayed in the text box, and it can be copied and pasted into the CompleteFTP manager in Step 4. Select "OK" and it should now be activated, and the license details should be displayed.

The following steps should be followed if you **do** have a universal AK:

- A. Run the CompleteFTP manager and connect to the CompleteFTP server instance that you wish to activate (the server instance - the CompleteFTP Windows service - may be running on another machine).
- B. Select the "Licensing" menu item from the list on the left in the manager, and then select the "Activate your license" link.
- C. Paste the universal AK you have been sent into the CompleteFTP manager in Step 4. Select "OK" and it should now be activated, and the license details should be displayed.

If you need to move an instance of CompleteFTP from one server to another, or are having difficulties with activating, please **email us**.