

# webconsulting FTP transfer Documentation

Bernhard Kraft

January 22, 2013

## Contents

<b>1 Project information</b>	<b>iii</b>	<b>2 Administrator manual</b>	<b>iv</b>
1.1 Releases . . . . .	iii	2.1 Creating a task . . . . .	iv
1.2 Bugs and known issues . . . . .	iii	2.2 Transfer task configuration . . . . .	iv
1.3 Todo . . . . .	iii	2.3 Other path prefixes . . . . .	ix
1.4 Changelog . . . . .	iii	2.4 Container tasks . . . . .	xi
1.5 Release log . . . . .	iv	<b>3 Developer corner</b>	<b>xii</b>

**Extension name** webconsulting FTP transfer

**Extension key** webcon\_ftpttransfer

**Version** 1.0.0

**Description** This extension allows you to create a scheduler task which is responsible for transferring files locally or to/from an FTP account. All configuration is done from within the task configuration form.

**Language** en

**Author** Bernhard Kraft <kraft@webconsulting.at>

**Creation** 21-01-2013

**Generation** 01-22-2013 13:29

**Licence** Open content license available from [www.opencontent.org/opl.shtml](http://www.opencontent.org/opl.shtml)

The content of this document is related to TYPO3, a GNU/GPL CMS/framework available from [www.typo3.org](http://www.typo3.org)

What does it do? This extension allows you to create a scheduler task which is responsible for transferring files locally or to/from an FTP account. All configuration is done from within the task configuration form. A transfer task requires a “source” location and a “target” location. When the task gets executed it will transfer all files which can be found at the “source” location to the “target” location. The “source” location and also the “target” location can be a local directory or a directory on a FTP server. If the transfer to the “target” location doesn’t succeed it is possible to have the files get moved to a “failed” location.

## Table of Contents

▼ WEB

Seite

Anzeigen

Liste

Template

Arbeitsumgebungen

Info

Zugriff

Funktionen

Powermail

Statistik

Papierkorb

TemplaVoilà

Formhandler

▼ DATEI

Dateiliste

▼ BENUTZERWERKZEUGE

Aufgaben

Einstellungen

▼ ADMINWERKZEUGE

Benutzer

Erweiterungen

DB-Überprüfung

Konfiguration

Installation

Protokoll

Indexierung

Berichte

AWStats

Planer

Quixplorer

Geplante Tasks

Geplante Tasks

Task bearbeiten

Deaktivieren

☒

Klasse

FTP Transfer (webcon\_ftpttransfer)

Typ

einmalig

Start (HH:MM TT-MM-JJJJ)

00:00 01-01-2013

FTP configuration

FTP hostname

FTPSERVER.COM

FTP port

21

FTP username

USERNAME

FTP password

\*\*\*\*\*

Path configuration

Source path

/var/www/html/cms/upload

Target path

ftp:/web-upload

Move to target location instead of copy

☐

Failed path

/var/www/html/cms2008/fil

Move to failure location instead of copy

☒

E-Mail addresses

Success e-mail

email@server.com

Failed e-mail

email@server.com

Fatal e-mail

email@server.com

Sender email

www-data@server.com

Logging configuration

Perform logging

☒

Log page UID

1234

Speichern

Abbruch

Figure 1: The scheduler task configuration form for a webcon FTP transfer task

# 1 Project information

This project was created for a customer of webconsulting in the need of transferring uploaded files to a FTP server. The website featured a multi-page formular where the visitor could upload a PDF document to the webserver. Those PDF files should then get uploaded to a FTP server for further processing. Of course such a task could have been achieved with shell script being executed from a cron daemon. But as this project was developed with TYPO3 and there (AFAIK) are no similar extensions we decided to create such a tool. The result is this extension which of course could easily get extended to support more protocols than just FTP.

If you like to contribute simply contact us at [office@webconsulting.at](mailto:office@webconsulting.at)

## 1.1 Releases

- The [current development version](#) can be downloaded from [forge.typo3.org](http://forge.typo3.org) subversion repository using a SVN client
- The [latest stable](#) release of this extension can be found at the TYPO3 extension repository

## 1.2 Bugs and known issues

All current bugs are listed at [forge.typo3.org](http://forge.typo3.org) issue tracker [issuetracker](#) This is also the place where new bugs, feature requests or other issues can get submitted.

## 1.3 Todo

The following tasks are still open:

- Move the “container” scheduler task into an extension of its own.
- Instead of having two built-in location objects: “local” and “FTP” alter those classes to suit a plugin system which allows other extensions to add transfer methods for SCP, HTTP-POST, WEBDAV, etc.
- Probably make some aspects of location objects configuratble via TS-Config (i.e. handling of files already existing at target location)
- Improve e-mail notification system (better markers, eventually BE-stdWrap, etc.)
- Properly implement the autoloader feature of newer TYPO3 versions

## 1.4 Changelog

A changelog of the progress for this extension can be found on [forge.typo3.org](http://forge.typo3.org) [webcon\\_ftptansfer](#) subversion repository in the file “[Changelog.txt](#)“

## 1.5 Release log

The following releases to TER are available

**Version** 1.0.0

**Release Date** 2013-01-22 / 12:00 CET

**Changelog** Initial release to TER

## 2 Administrator manual

As already explained the extension allows you to transfer files from local or remote directories to another directory. So the application logic introduces so called “locations”. There are three types of locations: “source”, “target” and “failed”. A webcon\_ftptansfer task will retrieve a list of all files which can be found at the “source” location and then try to transfer them to the “target” location. If transfer of a file from “source” to “target” fails it will try to move or copy it to the “failed” location.

The first step to take if you wish to use file transfers automated on a time basis is to make sure the “scheduler” extension is installed. The “scheduler” extension is a TYPO3 system extension so it should be shipped with your TYPO3 installation usually. Use the TYPO3 extension manager to install the “scheduler” extension. Of course you can also test the webcon\_ftptansfer extension without having a cron job set up. In such a case you would have to manually start the FTP transfer task for testing purposes. But the “scheduler” extension is required in any case as webcon\_ftptansfer only makes another scheduler task type available.

When the “scheduler” extension is installed you can install the “webcon\_ftptansfer” extension either by fetching it from TER, uploading a T3X or otherwise making the extension directory available in “typo3conf/ext”.

### 2.1 Creating a task

If both extensions are properly installed open the “Scheduler” BE module and click on the top icon for creating a new task. If you use an older versions of the scheduler there probably is no “Create new task” icon and you have to select the option “Information” from the drop-down. On the next screen you can choose which type of task should get created. Choose “FTP Transfer (webcon\_ftptansfer)”.

### 2.2 Transfer task configuration

Now the created task has to get configured. The above options are the same as for every scheduler task. Set it according to your needs and the scheduler documentation. There are four sections of configuration for the FTP transfer extension:

- FTP configuration: Allows to set FTP server, port, username and password
- Path configuration: Defines from which “source” location to which “target” location a transfer is performed
- E-Mail addresses: If configured a mail is sent for every successfull/failed file transfer

File Edit View History Bookmarks Tools Help

Backend [TYPO3 4.5.22]

www.frischeis.at/typo3/backend.php

ToDo - web:consultin... ReST Syntax - TYPO3...

TYPO3 admin Logout

Scheduled tasks

Scheduled tasks

	ID	Task	Type	Frequency	Parallel Execution	Last Execution	Next Execution
<input type="checkbox"/>	5	FTP Transfer (webcon_ftpttransfer)	Single	-	No	2013-01-21 13:02:01 (Cron)	-
<input type="checkbox"/>	6	FTP Transfer (webcon_ftpttransfer)	Single	-	No	2013-01-21 13:00:01 (Cron)	-
<input type="checkbox"/>	7	Container (webcon_ftpttransfer)	Recurring	120	No	2013-01-21 13:02:01 (Cron)	-
<input type="checkbox"/>	1	Update Extension List (em)	Recurring	86400	No	2013-01-21 03:42:02 (Cron)	2013-01-22 03:41:00

Execute selected tasks

Status Legend

- Failure! An error occurred during the last execution (move over bullet for more details)
- Late, will run with next execution
- Currently running
- Scheduled, will run on next possible execution
- Disabled, will not be executed, except manually

Server time

All dates and times in the Scheduler are measured according to the server's time, as the Scheduler is run purely on the server-side.  
Current server time is 2013-01-21 14:35:46 CET (Europe/Berlin, GMT +01:00).

1. Open scheduler module

2. Create new task alternate. Use option "information" of drop-down to create new task

FILE

- Filelist

USER TOOLS

- Task center
- User settings

ADMIN TOOLS

- User Admin
- Extension Manager
- DB check
- Configuration
- Install
- Log
- Indexing
- Reports
- AWStats
- Scheduler
- Quixplorer

HELP

- TYPO3 Manual
- About TYPO3
- TypoScript Help
- About Modules

Figure 2: Use the scheduler module to create a new webcon\_ftpttransfer task

- Logging configuration: Sets if and where log records should get stored

The screenshot shows the TYPO3 scheduler task configuration interface. On the left is a sidebar with navigation menus for WEB, DATEI, BENUTZERWERKZEUGE, and ADMINWERKZEUGE. The main area is titled 'Geplante Tasks' and 'Task bearbeiten'. The configuration form includes the following sections:

- Deaktivieren:** A checkbox that is checked.
- Klasse:** FTP Transfer (webcon\_ftpttransfer)
- Typ:** einmalig
- Start (HH:MM TT-MM-JJJJ):** 00:00 01-01-2013
- FTP configuration:**
  - FTP hostname: FTPSERVER.COM
  - FTP port: 21
  - FTP username: USERNAME
  - FTP password: (masked with asterisks)
- Path configuration:**
  - Source path: /var/www/html/cms/upload
  - Target path: ftp://web-upload
  - Move to target location instead of copy: (unchecked)
  - Failed path: /var/www/html/cms/2008/fil
  - Move to failure location instead of copy: (checked)
- E-Mail addresses:**
  - Success e-mail: email@server.com
  - Failed e-mail: email@server.com
  - Fatal e-mail: email@server.com
  - Sender email: www-data@server.com
- Logging configuration:**
  - Perform logging: (checked)
  - Log page UID: 1234

At the bottom of the form are buttons for 'Speichern' (Save) and 'Abbruch' (Cancel).

Figure 3: The scheduler task configuration form for a webcon FTP transfer task

In the following sections each of the above configuration settings gets explained in more detail.

## 2.2.1 FTP configuration

This section is required if you would like to transfer files from/to an FTP server. You simply have to fill in a valid hostname (i.e. ftp.yourserver.com) the IP port of the FTP control connections (usually port 21) and a valid username/password combination of an account on the FTP server. You can then use a prefix “**ftp:**” in the following path configuration to denote a source/target/failed location on this FTP server.

As the primary goal of this extension was to transfer files from a local web upload directory to an FTP server this way of configuration has been chosen. This could maybe change in future versions. The default syntax for FTP url’s in web-browsers and other extensions is “**ftp://username:password@hostname:port/path**”. So it can be that in future versions of this extension the configuration section for FTP credentials gets dropped and instead full RFC compliant URLs with an “**ftp://**” prefix are allowed.

## 2.2.2 Path configuration

In this section there are three path (locations) which can be configured. The “source” and “target” path are mandatory the “failed” path is optional. Each of the paths can either be a relative or absolute path on the local filesystem. An absolute path is recognized by starting with a slash “/” or “C:” (or any other drive letter) on Windows. Please be aware that Windows support is not tested at all. If the path doesn’t start with a slash it is supposed to be a path relative to the TYPO3 root directory (TYPO3 constant “PATH\_site”).

If the path starts with a prefix of “**ftp:**” then the path is supposed to be on the FTP server from the above FTP configuration section. Please note that the prefix is only “ftp” followed by a semicolon. No double slash! So a valid target directory could be “**ftp:test-upload/**” which is relative to the FTP home directory for the configured FTP username/password credentials. Another example would be “**ftp:/home/ftp/uploads**” which is an absolute path on the remote FTP site. Please be aware that many FTP servers feature a virtual filesystem and do not allow direct access to the servers UNIX/Windows filesystem structure.

So what is the difference between “source”, “target” and this mysterious “failed” path - and what are the checkboxes visible in above screenshot for? Here are the answers:

### Source path

The “source” location defines a directory from where a list of files is retrieved. If a local filesystem path is used simply all files (not directories) from within the specified path/directory will be queued for transfer. If an FTP location is specified a list of files in this directory is retrieved - every entry is checked for whether it is a file or a directory and only files will be queued for transfer. So there is (currently) no way to do a recursive transfer including subdirectories of the specified location.

For every file which is found in the source path also a log entry will get created after the transfer - may its outcome be successful (“success”), unsuccessful (“failed”) or even “**fatal**”. Uhm. “fatal”? What’s this now. That’s easy to explain: When a file transfer from “source” to “target” doesn’t succeed (probably because the target is a FTP server and it is offline or it’s disks are full) then the file will get moved/copied to the “failed” location: Its outcome/result is “failed”. But what is if even this transfer to the “failed” location fails? Then this transfer is considered to be “fatal”! No way to move, copy or properly handle the source file. You will recognize such a case as the scheduler task will stop working by returning “false”. This means that it also won’t get executed next time it is due. So a fatal outcome of a transfer usually disables further operation as its reason usually is improper configuration.

### Target path

The target path defines the location to which the files from the source path will get copied or moved. If a file will get copied or moved depends on the checkbox right below the target path field: “Move to target location instead of copy”. If it is checked the file is moved (deleted at the source location). If the checkbox is not checked the file is copied and exists twice afterwards.

One may ask what will happen if the file already exists at the target location. Nifty as programmers are this has also been taken into account (in fact this is the reason why programs always become more and more complicated: hundreds of possible things happen which a standard user did not expect). So if a file already exists at the target location simply a postfix of “\_01” will get added to the filename at the target location. If a file with the same name gets copied/moved to the target location again (A third time) a postfix

of “\_02” will get appended. And then “\_03”, “\_04” and so on until “\_10”. If there are already 11 versions of the same file(name) on the server something seems to be not fine and simply a 5-character random string gets appended to the filename. Please note that the postfix doesn’t simply get appended at the end of the filename+extension but rather only to the filename without extensions. So “userDocument.pdf” uploaded a second time becomes “userDocument\_01.pdf” and NOT “userDocument.pdf\_01”.

## Failed path

If the transfer of a file from “source” to “target” didn’t succeed then this extension will try to copy/move the file to the “failed” location. So if you are trying to upload files from a local directory to a FTP server and the target (FTP server) is offline won’t accept the uploaded file for some reason you have the possibility to move it to some other directory on the local machine for taking further actions.

The “failed” path can also be a “ftp:” location of course although it is suggested to use a local filesystem path for sane reasons.

Here you also have a checkbox labeled “Move to failure location instead of copy” which determines if the file from the “source” location will get moved to the “failed” location (removed from the “source” location) or if a copy is made.

## 2.2.3 E-Mail addresses

The concept of “success”, “failed” and “fatal” has already been shortly described in above section *Source path*. Again: When a file gets transferred the result can either be a successful transfer to the “target” location which is considered as “success”. Another outcome could be that transferring the file to “target” didn’t work but it could get moved to the “failed” location. Then the result is considered “failed”. If neither copy/move to “target” nor to “failed” properly works the outcome of this file transfer will be considered “fatal”.

Usually the scheduler will notice this by getting a negative (false) result from the FTP transfer task and will stop to further execute an instance of this task. For the purpose of prompt error handling (if for example a required directory on the server got accidentally deleted by a non-experience admin/user) there is the possibility of having a email notification. Of course you will also get notified of successful transfers if configured.

For this purpose there are three fields which can get filled with an e-mail address (Syntax: “*Recipient Name <recipient@server.com>*”). These are:

- Success e-mail: A mail will get sent to this addresse (addresses) upon successful transfer from “source” to “target”.
- Failed e-mail: When transfer from “source” to “target” doesn’t succeed an email will get sent to this address
- Fatal e-mail: When neither the transfer to the “target” nor to the “failed” location succeed an email will get sent to the “fatal” e-mail address

Depending on your TYPO3 version a different version of the **SWIFT** mailing framework is available in your system. Newer versions of SWIFT allow multiple e-mail addresses as recipient. To use this feature of SWIFT simply separate the multiple e-mail addresses in one of the above webcon\_ftpttransfer e-mail fields with a comma.



## 2.2.4 Logging configuration

The last part of the configuration options is about logging. The checkbox allows you to enable/disable logging at all. When a file gets transferred a database entry will get inserted into the table “tx\_webconftpttransfer\_log”. These are common TYPO3 TCA tables and you have to define a PID for them.

The screenshot shows the TYPO3 administration interface. On the left is a sidebar with a tree view containing 'few TYPO3 site', 'Home', 'TypoScript Templates', 'system\_configuration', 'extension\_configuration', 'menu\_configuration', 'page\_blocks\_configuration', and 'page\_configuration'. The main content area is titled 'Edit "tx\_webconftpttransfer\_LocationLocal: source: /var/w..." on page "TypoScript Templates"'. It contains several input fields: 'Timestamp' (9:25 22-1-2013), 'Filename' (empty), 'Size in bytes' (0), 'Target filename after successfull transfer' (empty), 'Target filename after failed transfer' (empty), 'Source path' (tx\_webconftpttransfer\_LocationLocal: source: /var/www/smb/t3\_intro/fileadmin/dataFiles), 'Target path' (tx\_webconftpttransfer\_LocationFtp: target: /test-upload), and 'Failed path' (tx\_webconftpttransfer\_LocationLocal: failed: /var/www/smb/t3\_intro/fileadmin/failedFiles). Below these is an 'Error messages' section with a text area containing 'WARNING: tx\_webconftpttransfer\_LocationFtp: target: /test-upload: Error changing FTP directory'. At the bottom left of the form is a checkbox labeled 'Show secondary options (palettes)'.

Figure 4: The log of a failed FTP transfer

If you view the log records afterwards you will notice that they are read-only. Which of course makes sense for log entries.

## 2.3 Other path prefixes

If you read above chapter carefully you will probably already have learned about the “ftp:” prefix which designates paths at a remote FTP server. In fact this prefix takes care of using another location handler than the one for the local filesystem. You can read about location handlers in the next chapter. But there is one more location handler which is shipped with webcon\_ftpttransfer: The “chain:” location handler. You can use this if you are in jail ... No. Wait. Then you made some other mistake. No. The “chain:” location handler

makes only sense in conjunction with container tasks. Two new things at once - this will get complicated so continue to read and you will understand.

Assume the following situation: You want to copy a file to a remote FTP server and on success move it to some other place in the local filesystem. You can easily achieve the copy to the FTP server by using a “ftp:” target location and by not checking the “Move instead of copy” checkbox for the “target” location. But how can you achieve the move to another local directory afterwards. You can not use the “failed” location as this is for handling failures of moving a file: source -> target.

This is where container tasks come into play. Container tasks are a feature which is integrated into webcon\_ftpttransfer but will probably be integrated in the default TYPO3 scheduler extension. A container task is simply a task where you can set a comma separated list of other tasks in the task configuration interface. See next chapter for more information.

So what we need in the above situation are two webcon\_ftpttransfer tasks. The first one is responsible for making a copy of each file to the remote FTP server and a second transfer task which is responsible for moving successfully uploaded files to another location in the local filesystem. And then a third task is required which is responsible for executing the other two tasks in a chained manner. The first two tasks should not get executed automatically on a timed basis but instead be set to get executed once at some already elapsed time. Only the container task should get executed via the scheduler cronjob.

The screenshot shows the TYPO3 Scheduler interface. The left sidebar contains navigation links for WEB, FILE, USER TOOLS, ADMIN TOOLS, and HELP. The main area displays a table of scheduled tasks. A red box highlights a container task (ID 12) and two other tasks (IDs 10 and 11) that are executed by it. A status legend and server time are also visible.

ID	Task	Type	Frequency	Successful Execution	Last Execution
10	FTP Transfer (webcon_ftpttransfer)	Single	-	No	22-01-13 09:25 (Manual)
11	FTP Transfer (webcon_ftpttransfer)	Single	-	No	16-01-13 15:05 (Manual)
1	Workspaces auto-publication (workspaces)	Recurring	0 * * * *	No	14-01-13 22:15 (Manual)
2	Caching framework garbage collection (scheduler)	Recurring	0 0 * * *	No	14-01-13 09:03 (Cron)
12	Container (webcon_ftpttransfer)	Recurring	60	No	16-01-13 15:05 (Manual)

Status Legend

- Failure! An error occurred during execution.
- Late, will run with next execution.
- Currently running
- Scheduled, will run on next post-scheduled execution.
- Disabled, will not be executed, e.g. because of a missing dependency.

Server time

All dates and times in the Scheduler are in UTC. Current server time is 22-01-13 15:05:00.

Container task.  
Usually gets executed automatically on a timed basis. Here manually for testing.

Tasks executed by container.  
These are set to be executed once at some already elapsed time as they get called by the container task.

Figure 5: Overview of scheduler tasks showing Container task and two webcon\_ftpttransfer tasks

But how can you make the second task to only move those files to some other location for which the FTP upload did succeed? If you move away “failed” files to another location you could of course only make a second task which has set the same source directory and moves the remaining (success?) files in to another place. But this is a hole for race conditions. What would happen if someone puts a new file in the “source” location between the first and second task being executed. To overcome such a race condition there is a “chain:” location handler.

In the second task you have to set “chain:success” as the “source” path. Please note that the “chain:” location handler is only valid for source paths. This can get explained by internal implementation details: When a webcon\_ftpttransfer task is transferring one file after the other from “source” to “target” it registers each file in an array having three subarrays: “success”, “failed” and “fatal” depending on the outcome of a file transfer. This array gets passed on to the next task in a chain. If the next task uses a “chain:” location object as “source” it fetches the files to transfer from this array variable. But of course there is no real implementation for storing a file anywhere - so “chain:” location handlers are only valid as “source”.

Of course you could add a third task having “chain:failed” as “source” which cares about handling files which didn’t succeed to get transferred to “target” by the primary task. It is NOT possible to take actions upon the outcome of secondary or third tasks in a chain.

## 2.4 Container tasks

In the previous section the usage of “chain:” location handler was explained. The use of this location handler is strongly coupled with the use of a container task. Currently a container task is some kind of task which is available because webcon\_ftpttransfer is installed. But maybe this task type will get integrated into future versions of the TYPO3 scheduler extension.

The screenshot shows the 'Edit task' form in the TYPO3 Scheduler. The form is for a 'Container (webcon\_ftpttransfer)' task. The 'Type' is set to 'Recurring'. The 'Start' date is '13:10 16-01-2013'. The 'End' date is empty. The 'Frequency' is '60' seconds. The 'Allow Parallel Execution' checkbox is unchecked. The 'UID of chained tasks as comma separated list' field contains '10,11' and is circled in red. A red arrow points from this field to a blue box labeled 'List of chained tasks'. The 'Server time' section at the bottom indicates the current server time is '22-01-13 10:02 CET (Europe/Berlin, GMT +01:00)'.

Figure 6: A container task configuration form

When creating a container task the only possible configuration option (except default task configuration fields) is a field in which you have to fill a comma separated list of task UIDs to execute in an ordered

chained manner. No matter whether a task returns false or true - every task listed in this field will get executed one after the other.

### 3 Developer corner

In the previous chapter(s) location objects were already mentioned a few times. So what are these location objects and how can I create a new one. Currently there are location object to handle local locations, FTP locations and chained locations. A local location is simply a directory on your local machine filesystem. Of course this directory could be a SMB or NFS mount in UNIX - the extension doesn't care about this. The FTP location object is used for accessing files on a (remote) FTP server. And the chained locations make sense for handling the results of a previous `webcon_ftpttransfer` task in conjunction with container tasks.

If you require something similar as featured by this extension: Copy the contents of a directory to another location on a timed regular basis but using another protocol (i.e: SCP or RSYNC) then what you need is a new location handler object. Currently location handlers are not implemented as pluggable system. So if you create a new location handler you will have to modify the source code of `webcon_ftpttransfer` to recognize your new location handler. Maybe future versions feature some kind of plugin system which allows to simply add new location handlers by creating a php class for them and registering them within the `webcon_ftpttransfer` extension by adding an entry to a `TYPO3_CONF_VARS` subarray.

Another feature which was already written about in the previous chapter are "Container Tasks". These are tasks which execute other tasks upon their own execution. Their sole purpose is to act as container for other tasks allowing them to pass data from one to the other. You will probably notice that such a task could get used for many purposes and does surely not belong to the `webcon_ftpttransfer` extension. Effort is underway to make such container tasks available by default in the TYPO3 scheduler extension.

If you have any other questions regarding the development of this extension or you require professional support don't hesitate to contact us at [office@webconsulting.at](mailto:office@webconsulting.at)