

dkd_staticupload

Extension Key: **dkd_staticupload**

Copyright 2000-2002, andreas.otto@dkd.de, <dkd_staticupload>

This document is published under the Open Content License
available from <http://www.opencontent.org/opl.shtml>

The content of this document is related to TYPO3
- a GNU/GPL CMS/Framework available from www.typo3.com

Table of Contents

dkd_staticupload.....	1	Configuration.....	3
Introduction.....	1	Using TS to configure a BE module.....	3
What does it do?.....	1	Overcoming "max_execution_time".....	4
Screenshots.....	1	Reference.....	4
Users manual.....	3	Known problems.....	5
Adminstration.....	3	To-Do list.....	5
		Changelog.....	5

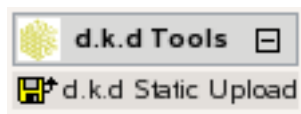
Introduction

What does it do?


The "Static Upload" module is a module that enables you to transfer the static HTML pages to a remote server.

Screenshots

Below are some screendumps which show the workflow of an upload process.






d.k.d Static Upload

 www.tycon3.dkd.de

Available services ▼


Path:

AVAILABLE SERVICES FOR STATICUPLOAD:

	Services:	Types:	Priority:	OS:	External:	Avail.:
	staticUpload (Service Type) Upload services for static HTML pages.					
	staticUpload: WINSXP [tx_dkdstaticuploadwinscp_sv1] Uses WINSXP.	scp	50	win	winscp3	
	staticUpload: SCP [tx_dkdstaticuploadscp_sv1] Uses SCP.	scp	50	unix	scp	

The "External" column shows which external programs are needed. If a service is not available, it might be the case that the program is not installed or can't be executed.

d.k.d Static Upload

 www.tycon3.dkd.de

Static upload ▼




Path:

STARTING STATIC UPLOAD

www.tycon3.dkd.de ▼

Upload pages

d.k.d Static Upload

   Home




Static upload ▼

Path: /Home/

STARTING STATIC UPLOAD

The file transfer to "tycon3-static.gimli.dkd.de" is running.
 An eMail will be sent to "dkd-otto@gimli.dkd.de" when the file transfer is finished.

d.k.d Static Upload

   Home

Static upload ▼

Path: /Home/

STARTING STATIC UPLOAD

The file transfer to "tycon3-static.gimli.dkd.de" is running.
 An eMail will be sent to "dkd-otto@gimli.dkd.de" when the file transfer is finished.

Users manual

After the module has been installed and configured the module will appear under “d.k.d Tools”.

Click on “d.k.d Static Upload” and if the module is used for the first time you will be taken to an overview of services which are available on your server.

If a service is marked as not available it might be that the specific service cannot be used on the OS of your server. It could also be the case that an external program is needed which is not installed on your server.

Select “Static upload” from the functions menu and you will be presented with a drop-down list containing all TYPO3 domain records from the current TYPO3 installation.

If you have selected one of the listed records, the upload process will be started. In case you wonder why you receive instant feedback on the screen – the upload process runs in the background. When the upload process has finished an eMail will be sent with more information.

Adminstration

After the extension “d.k.d Static Upload” has been installed make sure that at least one upload service, depending on your OS, has been installed as well. Currently two upload services are available:

1. staticUpload: SCP
2. staticUpload: WinSCP

The extensions named “staticUpload: SCP” and “staticUpload: WINSXP” are implemented as services and each of them offers a different method for transferring files from the TYPO3 server to the remote server.

During the installation of the services no configuration needs to be done. All configuration settings for the module “Static Upload” are done using TS.

To use a TS configuration for “Static Upload” two criteria must match:

1. A TYPO3 domain record must exist on the root page.
2. Inside the TS an object with the name of the domain must exist where “.” are replaced by “_”.

Using TS for the configuration has the advantage that a TYPO3 administrator can change settings without the need to change any PHP code.

Example TS configuration

```
0: service.staticUpload {
1:     www_tycon3_dkd_de {
2:         server = www.tycon3-static.dkd.de
3:         username = root
4:         path_to_php = /usr/bin
5:         path_to_scp = /usr/bin
6:         options = -i /var/lib/wwrun/.ssh/dkd_maintenance_dsa
7:         local_dir = /srv/www.tycon3.dkd.de/htdocs/publish/* /srv/www.tycon3.dkd.de/htdocs/fileadmin
/srv/www.tycon3.dkd.de/htdocs/uploads /srv/www.tycon3.dkd.de/htdocs/typo3temp
8:         remote_dir = /srv/www.tycon3-static.dkd.de/htdocs
9:         logfile_dir = /tmp
10:        service = scp
11:        email = andreas.otto@dkd.de
12:    }
13: }
```

Configuration

Using TS to configure a BE module

We use TS for the configuration settings of the BE module. It is therefor very important to include the right classes to get this to work. Since this topic pops up on the mailing lists from time to time we included what we have done in this documentation.

The following worked for us:

```
1: // DEFAULT initialization of a module [BEGIN]
2: unset($MCONF);
3: require ('conf.php');
4: require ($BACK_PATH.'init.php');
5: // DEFAULT initialization of a module [END]
6:
7: require_once( t3lib_extMgm::extPath('dkd_staticupload') .
'mod1/class.tx_dkdstaticupload_module1.php' );
8:
9: // TimeTrack'ing is needed e.g. by t3lib_TSTemplate
10: require_once(PATH_t3lib.'class.t3lib_timetrack.php');
```

```

11:
12: $TT = new t3lib_timeTrack;
13: $TT->start();
14: $TT->push('', 'Script start (dkd_staticupload/mod1)');
15:
16: $GLOBALS['LANG']->includeLLFile('EXT:dkd_staticupload/mod1/locallang.php');
17:
18: // Make instance:
19: $SOBE = t3lib_div::makeInstance('tx_dkdstaticupload_module1');
20: $SOBE->init();
21: $SOBE->main();
22: $SOBE->printContent();
23:
24: $TT->pull();

```

In the file “class.tx_dkdstaticupload_module1.php” we included:

```

1: require_once ($BACK_PATH.'template.php');
2:
3: require_once (PATH_t3lib.'class.t3lib_sbase.php');
4: require_once (PATH_t3lib.'class.t3lib_tstemplate.php');
5: require_once (PATH_t3lib.'class.t3lib_page.php');

```

This works for us and enables us to use TS to configure a BE module with TS. However, this creates PHP warnings because of not serving the right variable type to array functions:

```

1: {main}()
2: tx_dkdstaticupload_module1->init()
3: t3lib_TStemplate->start()
4: t3lib_TStemplate->matching()
5: t3lib_matchCondition->match()
6: reset()

```

Basically this is all about the variable “\$this->altRootLine” which should always be an array. In our case this variable is set to an empty string on line 4 of the stack trace above. A solution would be to check on line 5 if “\$this->altRootLine” is an array or not and then set it as an empty array if needed:

```

1: if ( !is_array( $this->altRootLine ) ) {
2:     $this->altRootLine = array();
3: }

```

Starting with TYPO3 3.9.0 the above mentioned problem has been solved. The fix we have proposed in this document became part of the TYPO3core.

Overcoming “max_execution_time”

It was clear from the beginning that we had to find a way to overcome the limitations imposed by script timeouts, server timeouts or browser timeouts.

To solve these problems we thought about moving the upload process into the background and sending a notification eMail to the address specified in the TS configuration. Especially on Win32 systems moving the process into the background was a bit more difficult.

Linux:

```

1: $cmd= sprintf( '%sphp "%scli/exec_bg_cli.php" "%s" 2>&1 &', $conf['path_to_php'],
t3lib_extMgm::extPath( $this->extKey ), base64_encode( serialize($conf) ) );
2: exec( $cmd );

```

Win32:

```

1: $cmd= sprintf( '%sphp -q "%scli/exec_bg_cli.php" "%s"', $conf['path_to_php'],
t3lib_extMgm::extPath( $this->extKey ), base64_encode( serialize($conf) ) );
2: // Initialize COM object to run the script as background process
3: $wsh = new COM("WScript.Shell");
4: $wsh->Run( $cmd, 7, false);
5: $wsh = null;

```

Reference

The reference explains only settings which are available for the services “staticUpload: SCP” and “staticUpload: WinSCP”. The configuration of other services can differ.

Property:	Data type:	Description:	Default:
www_example_com	string	Domain name, top level of the configuration. Must match a TYPO3 domain record. See “Example TS configuration”.	
server	string	The name of the target server.	

Property:	Data type:	Description:	Default:
username	string	Username on the target server.	
path_to_php	string	The path to the PHP cli executable.	
path_to_scp	string	The path to the used scp executable.	
options	string	Options for the scp executable.	
local_dir	string	One or more local directories which should be transferred to the remote server.	
remote_dir	string	Target directory on the remote server.	
logfile_dir	string	Directory on the local server where logfiles should be stored.	
service	string	Name of the service which should be used, e. g. scp.	
email	string	Email address where the notification will be sent to when the upload process has been finished.	

[tsref:service.staticUpload]

Known problems

- PHP warnings because due to wrong array handling in the TYPO3core. Solved starting with TYPO3 3.9.0. See “Using TS to configure a BE module” in this document.

To-Do list

- Use records for configuration instead of TS

Changelog