dkd_staticupload

Extension Key: **dkd_staticupload**

Copyright 2005, Andreas Otto, <andreas.otto@dkd.de>

Table of Contents

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

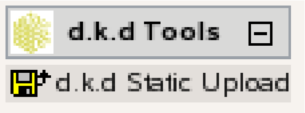
Introduction

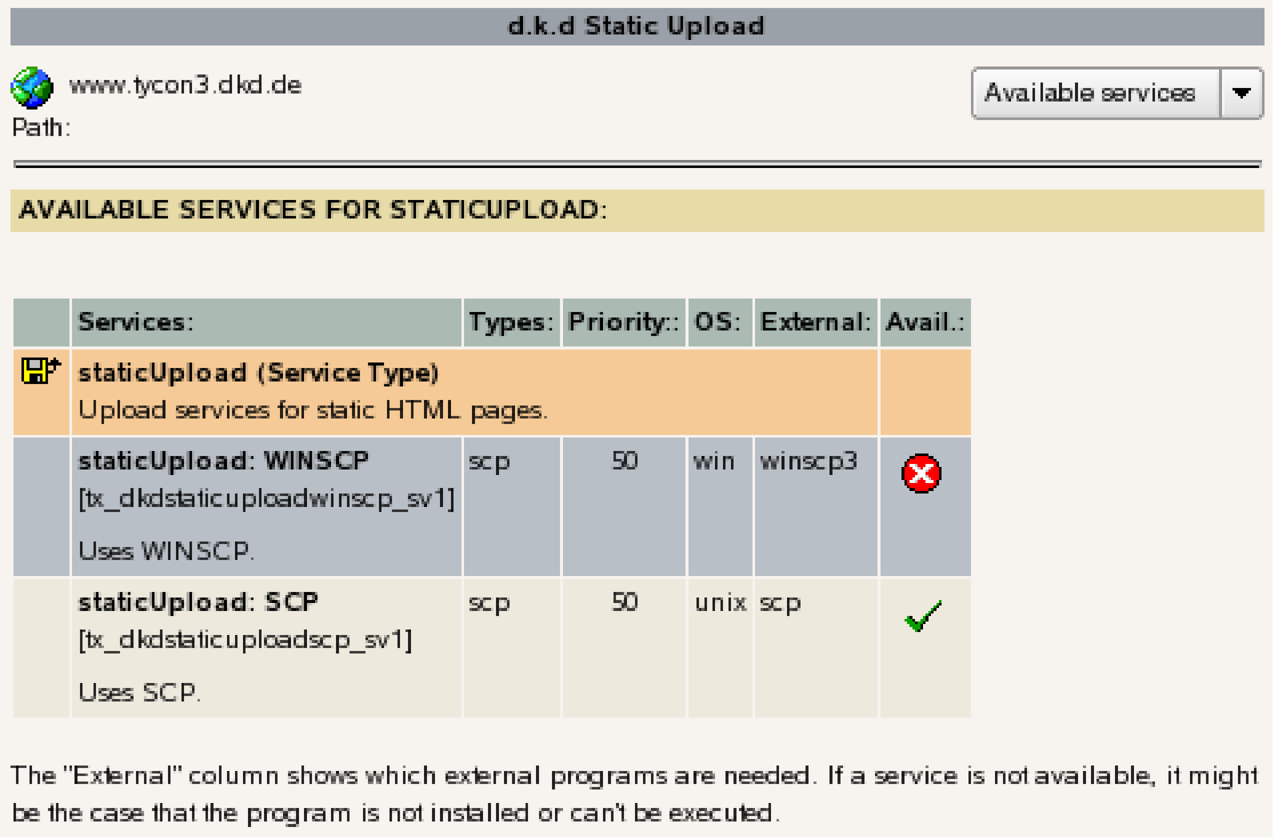
What does it do?

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.





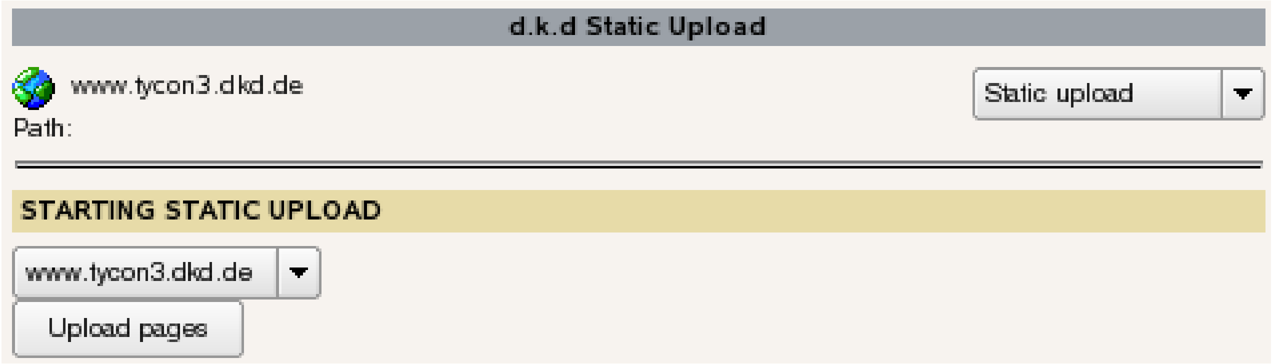
d.k.d Static Upload

www.tycon3.dkd.de
Path:

AVAILABLE SERVICES FOR STATICUPLOAD:

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

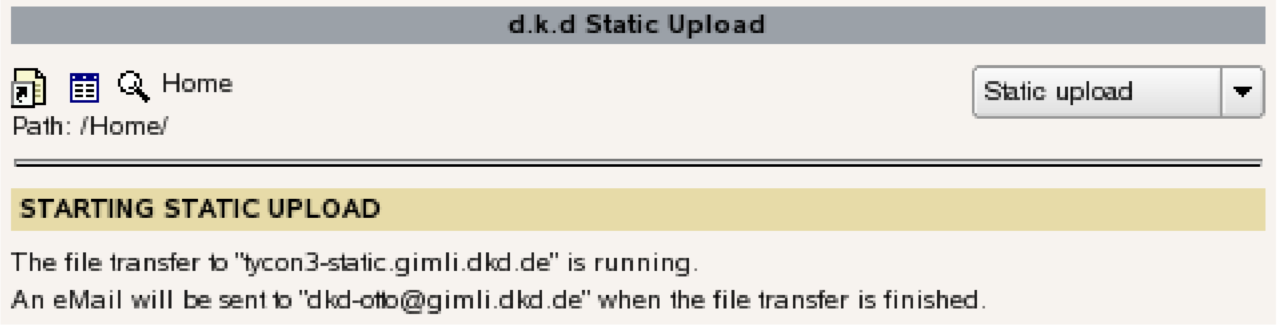
The "External" column shows which external programs are needed. If a service is not available, it might be the case that the program is not installed or can't be executed.



d.k.d Static Upload

www.tycon3.dkd.de
Path:

STARTING STATIC UPLOAD

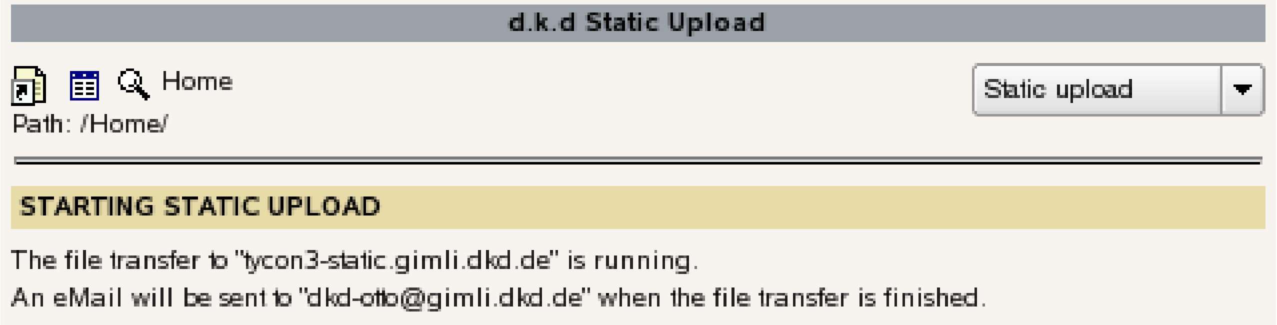


d.k.d Static Upload

Home
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
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

Verify that a BE user named “_cli_staticupload” exists. Only this BE user can run the CLI of TYPO3.

The extensions named “staticUpload: SCP” and “staticUpload: WINSCP” 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 for “staticUpload: SCP”

```
service.staticUpload {
  www_example_com {
    server = www.example.com
    username = root
    path_to_php = /usr/bin
    path_to_scp = /usr/bin
    options = -i /var/lib/wwwrun/.ssh/www.example.com_dsa
    local_dir = /srv/www.local.example.com/htdocs/publish/* /srv/www.local.example.com/htdocs/fileadmin /srv/www.local.example.com/htdocs/uploads /srv/www.local.example.com/htdocs/typo3temp
    remote_dir = /srv/www.example.com/htdocs
    logfile_dir = /tmp
    service = scp
    email = john.doe@example.com
  }
}
```

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:

```
// DEFAULT initialization of a module [BEGIN]
unset($MCONF);
require ('conf.php');
require ($BACK_PATH.'init.php');
// DEFAULT initialization of a module [END]

require_once( t3lib_extMgm::extPath('dkd_staticupload') . 'mod1/class.tx_dkdstaticupload_module1.php' );

```

```
// TimeTrack'ing is needed e.g. by t3lib_TStemplate
require_once(PATH_t3lib.'class.t3lib_timetrack.php');

$TT = new t3lib_timeTrack;
$TT->start();
$TT->push('','Script start (dkd_staticupload/mod1)');

$GLOBALS['LANG']->includeLLFile('EXT:dkd_staticupload/mod1/locallang.php');

// Make instance:
$SOBE = t3lib_div::makeInstance('tx_dkdstaticupload_module1');
$SOBE->init();
$SOBE->main();
$SOBE->printContent();

$TT->pull();
```

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

```
require_once ($BACK_PATH.'template.php');

require_once (PATH_t3lib.'class.t3lib_scbase.php');
require_once (PATH_t3lib.'class.t3lib_tstemplate.php');
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:

```
if ( !is_array( $this->altRootLine ) ) {
  $this->altRootLine = array();
}
```

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 limitiations 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:

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

Win32:

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

Reference

A reference of available settings can be found in the documentation of the upload services.

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