

# Installing and Upgrading

May 2002

Typo3 3.0

Copyright 2001-2002, Kasper Skårhøj, [kasper@typo3.com](mailto:kasper@typo3.com)

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

# Table of content

Installing Typo3.....	3	How bad is it?.....	20
<b>Quick Install.....</b>	<b>3</b>	Trivial updates.....	20
1-2-3.....	3	The filesystem.....	20
The Install Tool.....	3	Configuration files.....	20
Very Important: Changing Image Processing settings .....	6	TypoScript syntax change.....	21
Installing Typo3.....	7	Conformance with php.ini-optimized.....	21
<b>Verbose Install.....</b>	<b>7</b>	on-site.....	22
Which package and which fileformat?.....	7	PHP_SCRIPT include-scripts, libraries and backend modules.....	23
.zip distribution.....	7	<b>The Install Tool.....</b>	<b>27</b>
Typo3 distributed folders and documents.....	7	Aimed to end newbie agony.....	27
Site specific folders and documents.....	8	Install Tool Password.....	27
Custom folders?.....	8	<b>Installing a Typo3 webserver .....</b>	<b>28</b>
Duplicates in the .zip distribution.....	8	<b>on RedHat 6.1.....</b>	<b>28</b>
Why duplicates?.....	8	Disclaimer / NOTICE THIS!.....	28
.tgz distribution.....	10	Hardware.....	28
Internal symlinks.....	10	Software.....	28
The folders.....	10	Paths.....	29
Using the source distribution effectively.....	10	Installing.....	29
The upgrade advantage.....	11	Redhat 6.1.....	29
<b>Upgrade Typo3.....</b>	<b>12</b>	Joe (editor):.....	29
Philosophy.....	12	FreeType:.....	29
Previews and new versions?.....	12	ImageMagick.....	30
Upgrade.....	12	MySQL.....	30
How to update existing sites with "previews"? ..	18	PHP4 / Apache.....	31
<b>Upgrading from Typo3.1 to 3.2.....</b>	<b>20</b>	Apache user/group.....	31
Was Typo3.1 ever released?.....	20	Configure Apache (/www/conf/httpd.conf). ..	31
		Start Apache.....	32

# Installing Typo3

## Quick Install

**Note:** Quick Install allows you to configure the vital database information for Typo3 in three simple steps. It's meant for newbies and is automatically active if you download one of the zip-packages with complete sites. A "complete site" is a combination of the source-code and site-specific folders. Therefore this would *not* apply for the Typo3 source-code archives but only such downloads as the testsite or other out-of-the-box Typo3 applications.

### 1-2-3

1. Download **testsite-3.0\_FOR\_NEWBEEES.zip**
2. Upload the contents of this file to your PHP4 webserver. **Tip:** The zip-file contains thousands of files, so if you're able to unzip the file on the server, better do that!
3. Hit the site with your browser and follow the install tool. The install tool will help you to 1) setup MySQL username/password/database, 2) dump any default database content and 3) check the configuration of your PHP-installation for compatibility. This is outlined below.
4. After the install tool, **you should seriously consider to protect the folder typo3/install/ with a password or simply delete it.**

### The Install Tool

The Install Tool is located in 'typo3/install/' but should automatically start in "123"-mode with password 'joh316' if you have not set up any database for the site.

In other words, this is what you should see:

**1: Enter the username and password to your MySQL database:**

**Typo3.2a3 Install Wizard**

1: Basic Configuration  
2: Database Analyser

**Check database:**

**✘ Username, password or host not set**

You need to enter data for these values:  
Username:  
Password:  
Host:

Use the form below.

**Username:**

**Password:**

**Host:**

**Database:**  
(Can only select database if username/password/host is correctly set first)

2: Select the database (should be empty) or create a new by name:

**Typo3.2a3 Install Wizard**

1: Basic Configuration  
2: Database Analyser

**Check database:**

✓ **Connected to MySQL successfully**

Username: xxxxxx  
Password: xxxxxx  
Host: localhost

✗ **No database selected**

Currently you have no database selected.  
Please select one or create a new database.

Database: VCN  
Create database? (Enter name):

3: If everything is OK, continue to installation of the default dataset:

**Write config to localconf.php:**

✓ **Basic configuration completed**

You have no fatal errors in your basic configuration. You may have warnings though. Please pay attention to them! However you may continue and install the database.

**Step 2:** Click here to install the database.

4: If the database is empty (has zero tables), just click "Write to database" and the sql-file is dumped

The screenshot shows the 'Typo3.2a3 Install Wizard' interface. At the top, it lists two steps: '1: Basic Configuration' and '2: Database Analyser'. The current step is 'Database Analyser:'. It shows a green checkmark and the text 'Connected to MySQL successfully'. Below this, it displays 'Username: xxxxxx', 'Password: xxxxxx', and 'Host: localhost'. Another green checkmark is followed by 'Database', and a red box highlights the text 't31\_XXXXXXXX is selected as database. Has 0 tables.'. The next section is 'Import SQL dump:', which includes an information icon and the text 'Select tables to import'. It explains that this is an overview of CREATE TABLE definitions and that any table chosen for dumping will be dropped first. A checked checkbox is next to the text 'Import the whole file 't32\_testsite-32a1.sql' directly (ignores selections above):'. A red box highlights the 'Write to database' button at the bottom.

## Typo3.2a3 Install Wizard

1: Basic Configuration  
2: Database Analyser

### Database Analyser:

✓ Connected to MySQL successfully

Username: xxxxxx  
Password: xxxxxx  
Host: localhost

✓ Database

t31\_XXXXXXXX is selected as database.  
Has 0 tables.

### Import SQL dump:

**i** Select tables to import

This is an overview of the CREATE TABLE definitions in the SQL file. Select which tables you want to dump to the database. Any table you choose dump to the database is dropped from the database first, so you'll loose all data in existing tables.

Import the whole file 't32\_testsite-32a1.sql' directly (ignores selections above):

Write to database

**5: The mandatory setup is completed, but please remember to change the password of the Install Tool or take another appropriate action towards securing the tool.**

**Import SQL dump:**

**Imported ALL**  
Queries: 341

**Basic Installation Completed**

Apparently you have completed the basic setup of the Typo3 database.  
Now you can choose between these options:

- Continue to configure and verify your installation of Typo3 (recommended!).
- Go to the testsite pages.
- Go to the backend login. (username: *admin*, password: *password*)

**Security Risk!**

This script is a **great danger to the security of Typo3** if you don't use it somehow.  
We suggest one of the following options will improve security and is recommended highly.

Enter new password:

Enter again:

Set new password

Please consider to go into normal-mode with the Install Tool in order to verify all the details of the configuration. Especially the image generation software must be paid attention to!

See details in the next section of this document.

### **Very Important: Changing Image Processing settings**

When you change the settings for Image Processing (in normal mode) you *must* take into account that old images may still be in `typo3temp/` folder and prevent new files from being generated! This is especially important to know, if you're trying to set up image processing for the very first time.

The problem is solved by clearing the `typo3temp/` folder. Also make sure to clear the `cache_pages` table.

# Installing Typo3

## Verbose Install

### Which package and which fileformat?

Typo3 is available in various “packages” and within each package normally a .zip or .tgz distribution (targeted at Windows- and Unix-server users respectively).

- **typo3\_src**

This is the source files of the Typo3 CMA which is mandatory in all Typo3 systems. This is how the files might be named:

- |                   |   |
|-------------------|---|
| typo3_src-3.0.tgz | (Contains “symlinks”, so download only for Unix servers where you are able to create symlinks!)                       |
| typo3_src-3.0.zip | (This is the the safe-bet to download if the tgz-file is not your cup of tea. Works on both Unix and Windows servers) |

- **testsite**

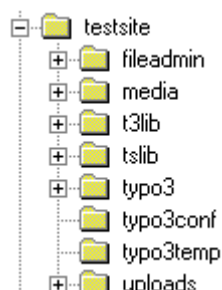
The testsite is a “complete site” with a lot of default content for testing and learning. **It's advisable to download the testsite to test Typo3.** This is how the filenames might be:

- testsite-3.0\_FOR\_NEWBEES.zip** (This is *both* the source *and* site-specific files and is ready to upload, unzip and run out-of-the-box! Please try this if you are a newbie!  
Can be used on both Unix and Windows servers)

testsite-3.0.tgz (This does not contain the source files but only symlinks to the source following the generally recommended practice of symlinks. This is for experienced Typo3 users only.)

### .zip distribution

This is a out-of-the-box package for ftp-upload. It is integrated with a testsite that will help you to get started with Typo3. The filestructure looks like this:



### Typo3 distributed folders and documents

**media/**, **t3lib/**, **tslib/**, **typo3/** contains the Typo3 source-code, files, images and scripts distributed by Typo3.

**index.php** is the main script for the website frontend (same as tslib/index\_ts.php).

**showpic.php** is a document used to display images on request (same as tslib/showpic.php).

**clear.gif** is a 1x1 clear-pixel gif-file. Used many places.

*Don't change any of these!*

## Site specific folders and documents

**fileadmin/** contains files of your choice. You can put external scripts, external html and images here. Also used to store uploaded files by default.

**typo3temp/** should be empty from the beginning. But gradually temporary files will appear here. The directory must be writeable.

**uploads/** contains folders used to store documents attached to database records - hence they should be writeable!

**typo3conf/** contains configuration for the local site, eg. localconf.php, extTables.php-files, user-defined modules, user-defined classes etc.

**typo3conf/localconf.php** contains local configuration for your site.

## Custom folders?

Yes just add whatever you like. Why not?

## Duplicates in the .zip distribution

The reason why the .zip distribution for ftp-upload is much larger than the .tgz version is that numerous things are stored twice or more in the zip file. That's not the case with the .tgz version.

**t3lib/** and **typo3/t3lib/** is the same.

This is a directory with class-files used by the front-end as well as the backend. Both the frontend, TSFE, (in the root) and backend, TBE, (in typo3/) require the directory to be found relative to their own path. Therefore it's listed twice. There are more than 400 files in the folder, most of them small image-files like icons.

**media/** and **tslib/media/** is the same.

The media/ folder is used by the frontend. It contains scripts and media-resources used by the frontend. Technically tslib/media/ folder is the 'original' folder, but on an actual site only the media/ folder is (and should always be) used. Therefore you may delete tslib/media/ (if it's not already done to save space in the distribution).

**typo3/gfx/** and **...t3lib/gfx/** is the same.

The gfx/ folder represents 300 icons and imagefiles. Those are used primarily in the backend from the folder typo3/gfx/. Some are used by the front-end as well (which is the reason why it's found in the t3lib/ folder anyway). Because the backend (TBE) uses the content of typo3/gfx/ and not typo3/t3lib/gfx/ you may delete typo3/t3lib/gfx/.

**index.php** and **tslib/index\_ts.php** is the same.

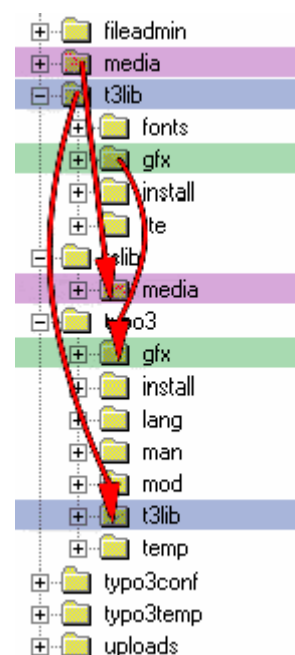
**showpic.php** and **tslib/showpic.php** is the same.

**typo3conf/tables.php** and **t3lib/install/tables.php** is the same (in most cases should be!).

## Why duplicates?

The reason is that the "real" source distributed in the .tgz file uses symbolic links (UNIX filesystem feature) to point to both folders and files, which are duplicates. For instance the folder typo3/gfx/ is a symbolic link (symlink) to typo3/t3lib/gfx/. That means that the content of the folders are actually the same files - if you edit, add or delete a file in typo3/t3lib/gfx/ it's reflected in typo3/gfx/ and vice versa.

So using symbolic links on UNIX is an extremely efficient way to share files and directories without copying them around and as you can see, there are no duplicates in the source-distribution - just symlinks to the





'original' folder. But when you make a zip-file of such a filestructure the linked files and directories are copied as indepent files. This provides the duplicates.

**So the .zip-files** include redundant code and graphics. The up-side to this is that you don't have to worry about "symlinks" etc - just upload the zip-file contents to the webserver and run it. The down-side is that the zip-distribution is much larger than the tgz-distribution (and secondly the zip-distribution - that is if you don't use the smart symlinks in the .tgz-distribution - upgrades are a little more work).

**On the other hand the .tgz-files** are also decompressed by applications like WinZip. **Notice**, that if you decompress such as the Typo3 source from a tgz-file (eg. "typo3\_src-3.0.tgz") with WinZip *the symlinks are simply ignored!* Thus you would have to manually copy the t3lib/ folder to typo3/t3lib and the t3lib/gfx/ folder to typo3/gfx/ in order to compensate for that loss.

Please see the section about the .tgz distribution for more details.

## .tgz distribution

The tgz distribution is the 'source code only' and is not accompanied with any testsite. It's also intended for UNIX filesystems because the .tgz-file contains relative symlinks.

The main point is - compared to the .zip distribution - that this directory is stored once in the file-structure (possibly writeprotected **except from typo3/temp/ folder**) on the UNIX server and then every site using this Typo3 version symlinks the proper folders to the source. Thus you save space because you avoid redundant files and get easy maintenance if you do it right.

### Internal symlinks

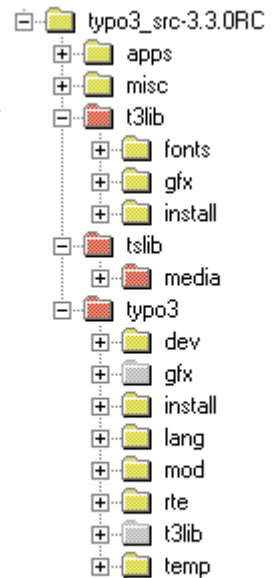
This is the internal symlinks in the tarball:

```
typo3/t3lib/ -> t3lib/
```

```
typo3/gfx/ -> typo3/t3lib/gfx/
```

```
typo3/thumbs.php -> t3lib/thumbs.php
```

These are shown with gray in the picture. Actually if you extract the .tgz-file with winzip, you'll not see the symlinked folders (gray) in typo3/. They are removed (see notice on previous page about this!)



### The folders

**t3lib/**, **tslib/**, **tslib/media**, and **typo3/** (red folders) are the main directories which should be symlinked to the actual website running Typo3.

**apps/** contains some 'projects' called applications which are modules or scripts extending Typo3 functionality. The applications are normally related to a certain database. You should use the applications in conjunction with a downloaded out-of-the-box application zip-file with everything like the database, documentation and configuration provided.

**misc/** contains information from previous Typo3 releases.

**files in the root:** Documents related to the current release, eg. changelog.

### Using the source distribution effectively

As I wrote the source distribution provides a minimum of space usage and maintenance if used intelligently. This is my suggestion:

1. Extract the .tgz file somewhere on your server
2. From the directory holding the root-directory of you site, create a symlink to the source folder. Call it 'typo3\_src':

```
drwxr-xr-x  7 httpd  httpd  4096 Aug  3 10:24 .
drwxr-xr-x 12 httpd  httpd  4096 Aug  1 13:52 ..
drwxr-xr-x  6 httpd  httpd  4096 Aug  1 14:09 testsite-3.0
lrwxrwxrwx  1 root   root   15 Aug  1 13:58 typo3_src -> typo3_src-3.0/
drwxr-xr-x  9 httpd  httpd  4096 Aug  7 00:37 typo3_src-3.0
```

(The green 'testsite-3.0' is the Typo3 website. The blue 'typo3\_src-3.0' is the source distribution as it was un-tarred. The red is the symlink to the source distribution. You create such a symlink from shell like this:

```
ln -s ./typo3_src-3.0 typo3_src
)
```

3. In the root of your site, create symlinks for t3lib/, tslib/ and typo3/ main-folders to the main-folders in the newly created 'typo3\_src' symlink in the parent folder.

```
drwxr-xr-x  6 httpd  httpd  4096 Aug  1 14:09 .
drwxr-xr-x  7 httpd  httpd  4096 Aug  3 10:24 ..
-rwxr-xr-x  1 httpd  httpd   46 Sep  7 1999 clear.gif
drwxr-xr-x  5 httpd  httpd  4096 Apr 13 00:30 fileadmin
```

```

lrwxrwxrwx 1 httpd httpd 18 Aug 1 13:54 index.php -> tslib/index_ts.php
lrwxrwxrwx 1 httpd httpd 12 Aug 1 13:54 media -> tslib/media/
-rwxr--r-- 1 httpd httpd 56 Sep 25 2000 mod_rewrite.htaccess
lrwxrwxrwx 1 httpd httpd 17 Aug 1 13:54 showpic.php -> tslib/showpic.php
lrwxrwxrwx 1 httpd httpd 19 Aug 1 13:54 t3lib -> ../typo3_src/t3lib/
lrwxrwxrwx 1 httpd httpd 19 Aug 1 13:54 tslib -> ../typo3_src/tslib/
lrwxrwxrwx 1 httpd httpd 19 Aug 1 13:54 typo3 -> ../typo3_src/typo3/
drwxr-xr-x 2 httpd httpd 4096 Aug 6 22:54 typo3conf
drwxr-xr-x 2 httpd httpd 4096 Aug 6 22:31 typo3temp
drwxr-xr-x 6 httpd httpd 4096 Nov 12 1999 uploads

```

4. In the root of your site, create symlinks to tslib/media/, tslib/showpic.php and tslib/index\_ts.php (call this one 'index.php').

```

drwxr-xr-x 6 httpd httpd 4096 Aug 1 14:09 .
drwxr-xr-x 7 httpd httpd 4096 Aug 3 10:24 ..
-rwxr-xr-x 1 httpd httpd 46 Sep 7 1999 clear.gif
drwxr-xr-x 5 httpd httpd 4096 Apr 13 00:30 fileadmin
lrwxrwxrwx 1 httpd httpd 18 Aug 1 13:54 index.php -> tslib/index_ts.php
lrwxrwxrwx 1 httpd httpd 12 Aug 1 13:54 media -> tslib/media/
-rwxr--r-- 1 httpd httpd 56 Sep 25 2000 mod_rewrite.htaccess
lrwxrwxrwx 1 httpd httpd 17 Aug 1 13:54 showpic.php -> tslib/showpic.php
lrwxrwxrwx 1 httpd httpd 19 Aug 1 13:54 t3lib -> ../typo3_src/t3lib/
lrwxrwxrwx 1 httpd httpd 19 Aug 1 13:54 tslib -> ../typo3_src/tslib/
lrwxrwxrwx 1 httpd httpd 19 Aug 1 13:54 typo3 -> ../typo3_src/typo3/
drwxr-xr-x 2 httpd httpd 4096 Aug 6 22:54 typo3conf
drwxr-xr-x 2 httpd httpd 4096 Aug 6 22:31 typo3temp
drwxr-xr-x 6 httpd httpd 4096 Nov 12 1999 uploads

```

## The upgrade advantage

The advantages of this approach is apparently that when you wish to upgrade to a new TYPO3 version, you simply change one symlink and every distributed TYPO3 file is upgraded instantly - on as many sites as you like!

### Example:

```

drwxr-xr-x 8 httpd httpd 4096 Aug 7 01:07 .
drwxr-xr-x 12 httpd httpd 4096 Aug 1 13:52 ..
drwxr-xr-x 6 httpd httpd 4096 Aug 1 14:09 testsite-32
lrwxrwxrwx 1 root root 15 Aug 1 13:58 typo3_src -> typo3_src-3.0/
drwxr-xr-x 9 httpd httpd 4096 Aug 1 00:37 typo3_src-3.0
drwxr-xr-x 9 httpd httpd 4096 Aug 7 00:37 typo3_src-3.1

```

Then do this from shell:

```

rm ./typo3_src
ln -s typo3_src-3.1 typo3_src

```

.. and this is what you get:

```

drwxr-xr-x 8 httpd httpd 4096 Aug 7 01:07 .
drwxr-xr-x 12 httpd httpd 4096 Aug 1 13:52 ..
drwxr-xr-x 6 httpd httpd 4096 Aug 1 14:09 testsite-32
lrwxrwxrwx 1 root root 15 Aug 1 13:58 typo3_src -> typo3_src-3.1/
drwxr-xr-x 9 httpd httpd 4096 Aug 1 00:37 typo3_src-3.0
drwxr-xr-x 9 httpd httpd 4096 Aug 7 00:37 typo3_src-3.1

```

# Upgrade Typo3

## Philosophy

The intension is to make upgrades backwards compatible. But sometimes you may experience changes that affect your website. Normally the change is small, but always backup the database and files before an upgrade and possibly also test the upgrade on another server first.

If you conform to the recommended database-structures and don't use custom tables.php-files, it should be quite easy to upgrade.

## Previews and new versions?

Normally months passes by between new Typo3 versions are out. But in between there are often a lot of previews, every week sometimes! So whats the difference and what can you expect from them?

- A "new version" is a new set of "official" packages including the source as tgz/zip, the testsite as tgz/zip, the dummy-tgz etc. published on sourceforge.net. You can expect a "new version" to be tested good enough for production sites. (Still, make sure to read the changelog for !!! marked lines and make a test upgrade on a separate server first to make sure no changes affects your site negatively!)
- A "preview" is simply a tgz-snapshot of todays work on Typo3! You can normally expect these for work perfectly, but their completeness is not guaranteed to the same degree as with the "new versions". The "previews" are recommended to download and test and give feedback from. Please see the section below for how to handle the installation of the previews!

## Upgrade

When new versions of Typo3 arrive, you should always follow any additional the upgrade information carefully. You might also need to skim the changelog to see if any features affects the way your site works (look for lines prepended with !!! - those are the really important!)

Basically these steps are involved in updating your Typo3 site:

### **1: Backup**

Make a backup of both uploads/ folder and database. Also you may prefer to upgrade a copy of your site first if there has been a lot of changes and some of them seems to interfere with your site. See the changelog.

### **2: Install the new source**

Upgrade the distributed source files to the new version.

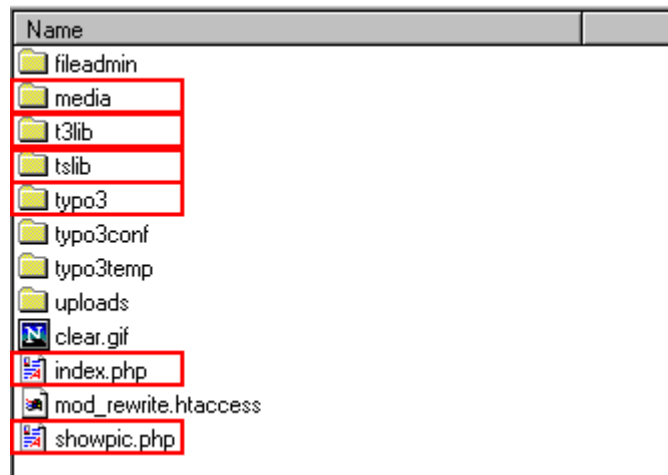
#### **.tgz:**

If you're using the source-distribution in the .tgz file, please see right above where details on symlinking is explained.

#### **.zip:**

If you're using the .zip distribution, then replace these files and folders:

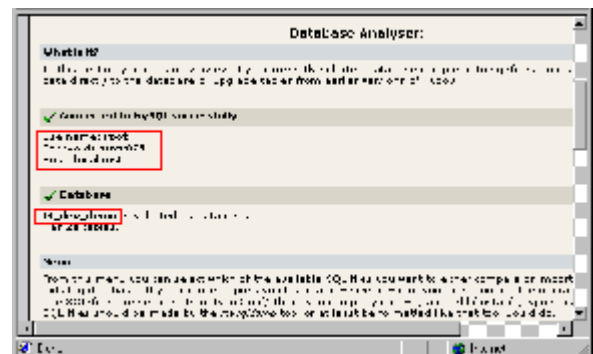
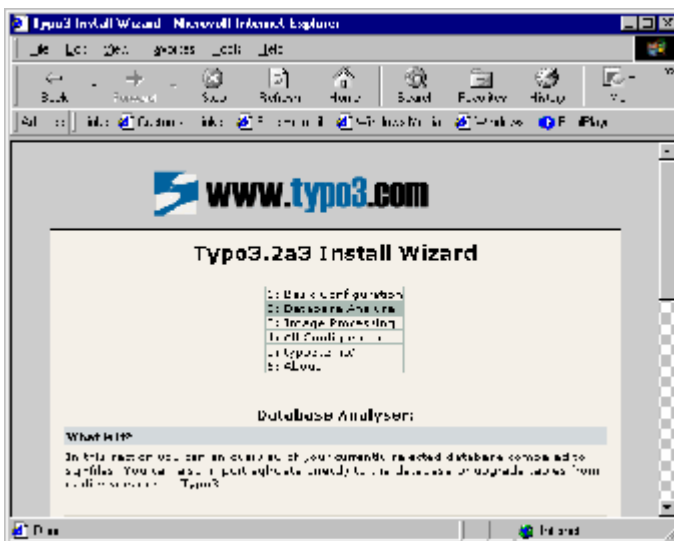
**media/, t3lib/, tslib/, typo3/, index.php and showpic.php**



### 3: Compare and Update the database

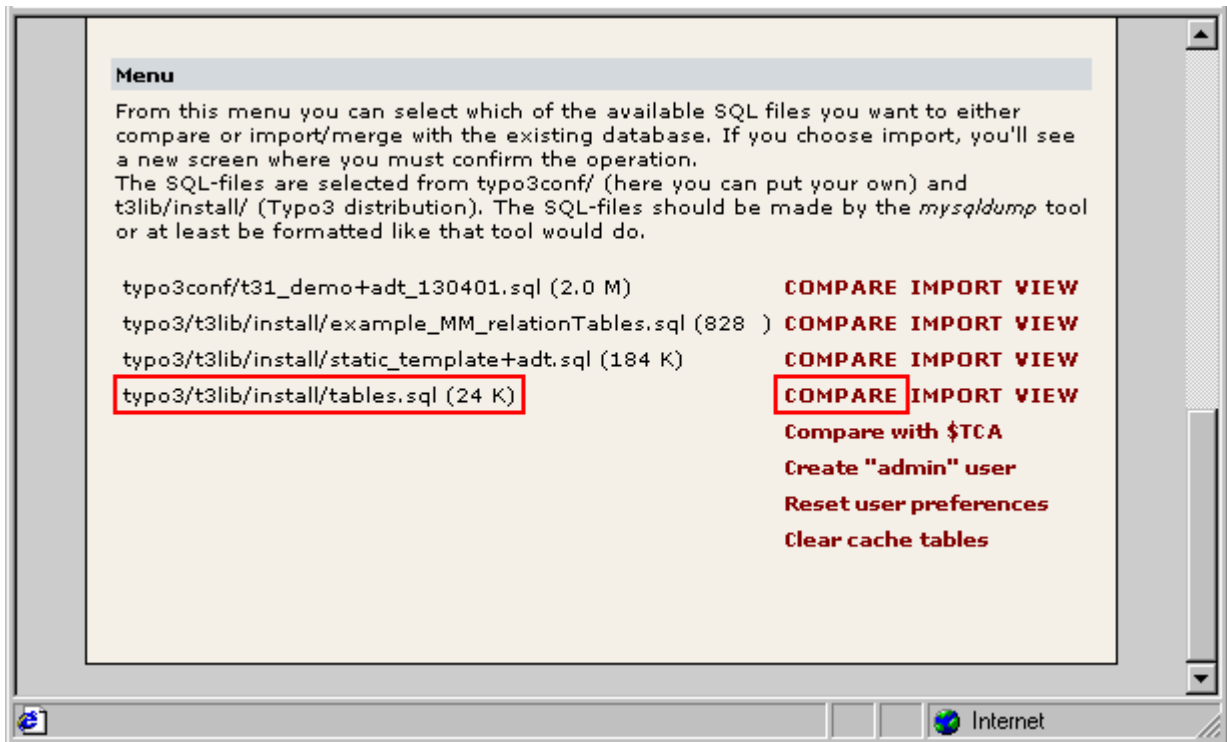
Enter the Install Tool at '.../typo3/install/' on your Typo Site.

(If you have password/access problems, please see the section about the Install Tool later in this document).



Go to the "Database Analyser" section, check that the database configuration is correct (above).

Then click "COMPARE" at the "typo3/t3lib/install/tables.sql" file:



Now the tables.sql file is read and compared to your current database tables and fields. Any discrepancies will be shown and you'll be able to execute queries sufficient to upgrade you database so it corresponds to the structure required for the new Typo3 version.

**Notice:** Be aware if you have deliberately added fields and/or tables to your Typo3 database for your own purposes! Those tables and fields are removed only if you mark them to be deleted of course, but please be alert that you don't overlook them!

## Update database tables and fields:

### ⚠ Table and field definitions should be updated

There seems to be a number of differences between the database and the selected SQL-file. Please statements you want to execute in order to update your database:

## Add fields

ALTER TABLE be\_groups ADD TSconfig blob NOT NULL;

ALTER

ALTER

ALTER

ALTER

ALTER

## Changing fields

ALTER TABLE be\_groups CHANGE non\_exclude\_fields non\_exclude\_fields blob NOT NULL;

Current value: *text NOT NULL*

ALTER TABLE be\_groups CHANGE pagetypes\_select pagetypes\_select tinyblob NOT NULL;

Current value: *tinytext NOT NULL*

ALTER TABLE be\_groups CHANGE tables\_select tables\_select tinyblob NOT NULL;

## Remove unused fields (rename with prefix)

ALTER TABLE cache\_pages CHANGE data zzz\_deleted\_data blob NOT NULL;

ALTER TABLE pages CHANGE nextUpdate zzz\_deleted\_nextUpdate int(10) unsigned NOT NULL;

ALTER TABLE sys\_dmail CHANGE url zzz\_deleted\_url varchar(120) NOT NULL default '';

ALTER TABLE sys\_dmail CHANGE preview\_recipient zzz\_deleted\_preview\_recipient varchar(80) NOT NULL;

ALTER TABLE sys\_dmail CHANGE recip\_table zzz\_deleted\_recip\_table varchar(40) NOT NULL default '';

ALTER

ALTER

ALTER

ALTER

ALTER

## Add tables

CREATE TABLE cache\_search (  
id int(11) unsigned NOT NULL auto\_increment,  
hash varchar(32) NOT NULL default '',  
page\_id int(11) unsigned NOT NULL default '0',  
HTMLfreeText mediumtext NOT NULL,  
tstamp int(11) unsigned NOT NULL default '0',  
tempFile\_data blob NOT NULL,  
cHashParams tinyblob NOT NULL,  
page\_type int(10) unsigned NOT NULL default '0',  
page\_gr\_list varchar(20) NOT NULL default '',  
PRIMARY KEY (id),  
KEY page\_id (page\_id),  
KEY sel (hash)  
) TYPE=MyISAM;

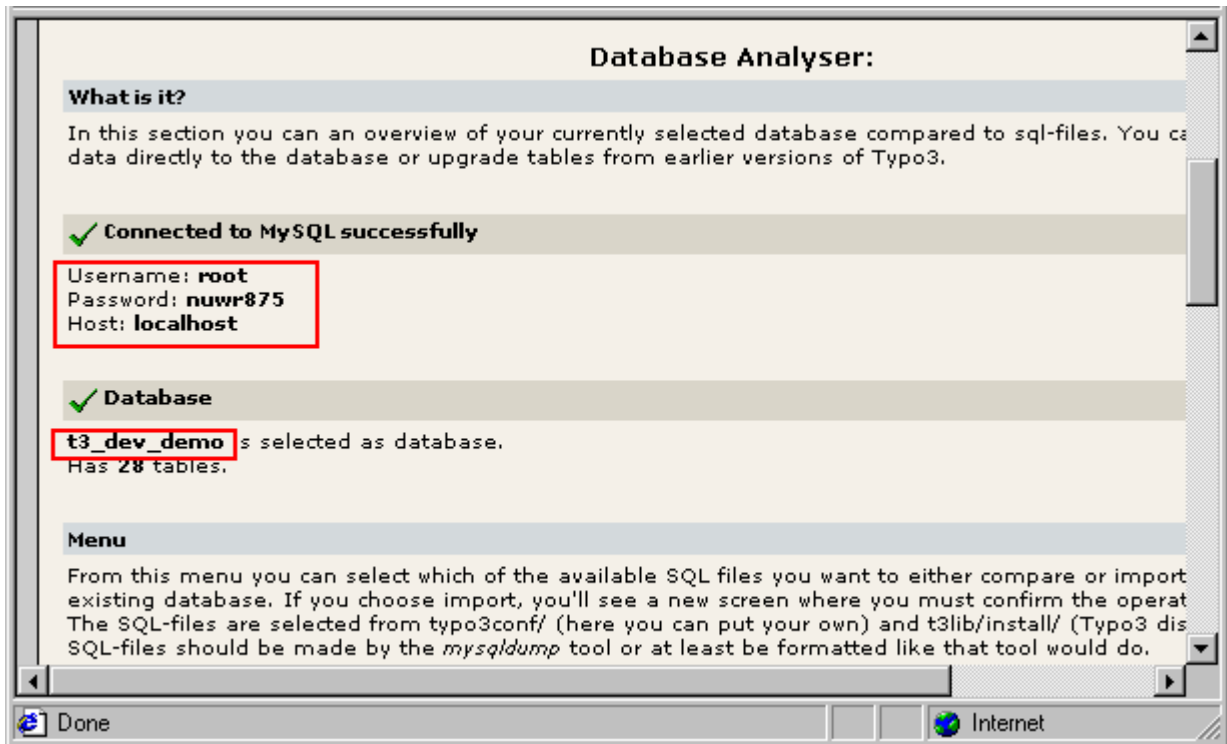
CREATE TABLE sys\_tracking (  
tracking\_code int(11) unsigned NOT NULL default '0',  
status int(11) unsigned NOT NULL default '0',  
KEY tracking (tracking\_code),  
KEY status (status)  
) TYPE=MyISAM;

CREATE TABLE sys\_products\_orders\_MM\_tt\_products (  
sys\_products\_orders\_uid int(11) unsigned NOT NULL default '0',  
sys\_products\_orders\_qty int(11) unsigned NOT NULL default '0',  
tt\_products\_uid int(11) unsigned NOT NULL default '0',  
KEY tt\_products\_uid (tt\_products\_uid),  
KEY sys\_products\_orders\_uid (sys\_products\_orders\_uid)  
) TYPE=MyISAM;

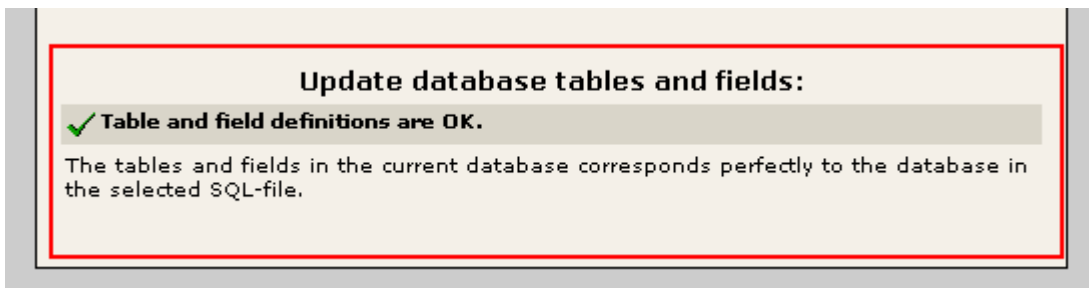
Write to database

Remove unused fields (and tables) if you wish and click "Write to database"

On the next screen you'll see that all add and changed fields are done properly since they don't show up again. However the fields and tables we chose to drop were renamed first. Now you can consider to let them be and delete them later on when you're sure you're not going to need them. Or you can just go on drop them finally.



Finally, you should see this:

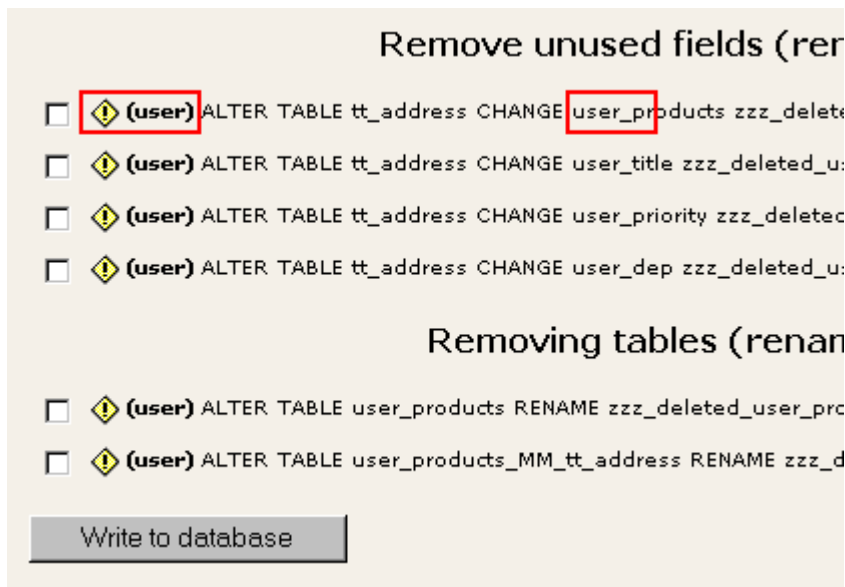


## CAUTION!

You may have added your own fields to the database tables or you may have added tables on your own. **Don't delete those (of course)!**

If you're smart, you name user defined tables and fields prefixed "user\_". Then they'll show up with a little warning icon, so you can easier distinguish them from fields and tables which should be dropped.

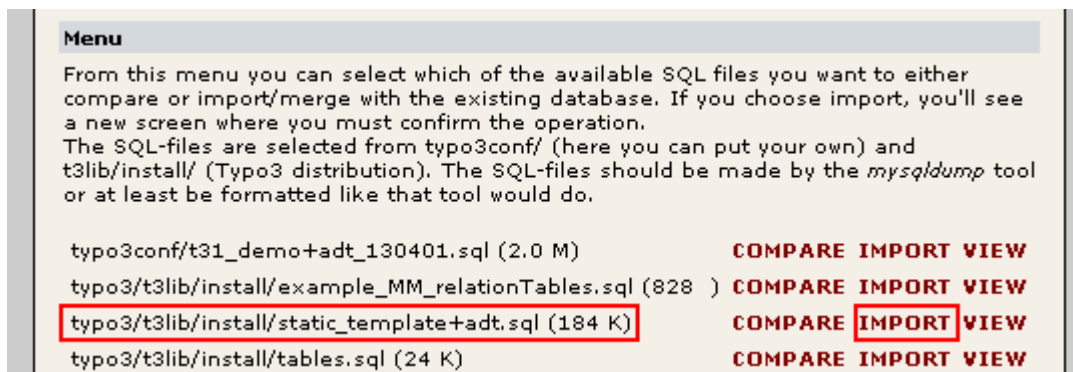




#### 4: Install the new static tables and description tables

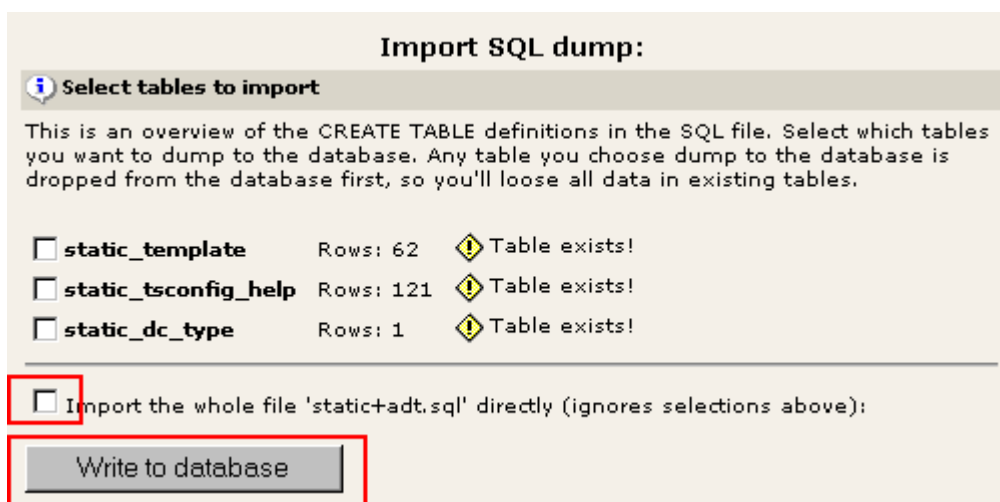
The table 'static\_template' (and some other "static\_" tables) contains information that you should never alter yourself. The point is that this tables contains static information distributed with Typo3 releases. Follow these steps to upgrade the table with the newest information.

Go to the "Database Analyser" section.

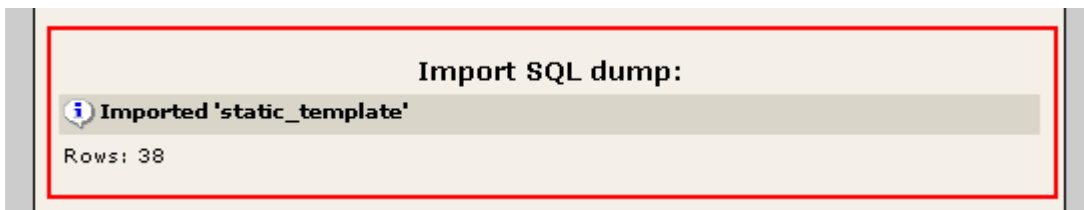


Then click "IMPORT" at the "typo3/t3lib/install/static+adt.sql" file:

Then tick the checkbox "Import the whole file..." and click "Write to database":



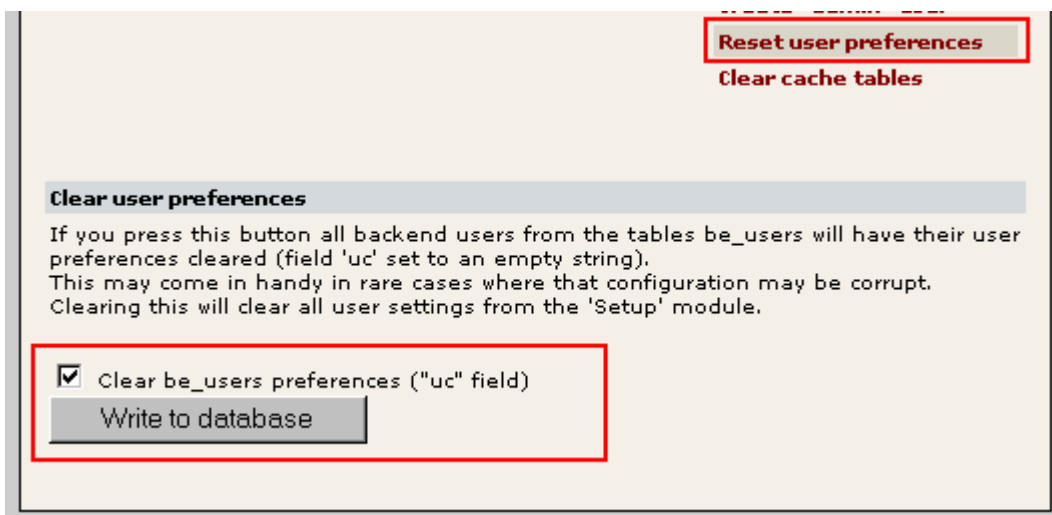
... and you should see (something like) this without errors:



You should also install the tables "typo3/t3lib/install/sys\_tabledescr\_X+adt.sql" (contains system help). Do it exactly as with the static+adt.sql file:

- Click "IMPORT" at the "typo3/t3lib/install/sys\_tabledescr\_X+adt.sql" file
- Then tick the checkbox "Import the whole file..." and click "Write to database"

### 5: Clear cache-tables, typo3temp/ if needed, clear user-settings if needed.



You should always clear the cache-tables. You might also consider clearing the backend user preferences if something in the upgrade requires this. Everything can be done from the Install Tool.

(This examples shows how to clear the 'uc' field of all users in be\_users table.)

### 6: Check the changelog, changes.txt

Look through the changelog. It's a good place to get an idea about new functions you might need or be interested in.

Especially pay attention to the lines starting with "!!!". They are important because they are about changes that may affect some websites.

The changelog is found in the root of the source distribution or it is for download with the source-archives.

## How to update existing sites with "previews"?

When a new preview is out, they are available only as tgz-files with a simple compressed snapshot of "todays code".

In order to upgrade an existing site with a preview there are these scenarios to think about BEFORE the 6 steps above is carried out:

- If you are using symlinks on Unix, unzip the source, remove old symlink, create new symlink.
- Otherwise **follow these guidelines for correctly swapping the old source with the new:**

1. Unzip the "preview" tgz-file with WinZip to some directory on your harddisc.
2. Compensate for the ignored symlinks in the tgz-file by:
 

```
Copy  t3lib/          ->  typo3/t3lib/
Copy  t3lib/gfx/      ->  typo3/gfx/
```
3. Substitute the old source with the new by removing old dirs and copying new ones:
 

```
Copy  typo3/          ->  [your_site]/typo3/
Copy  t3lib/          ->  [your_site]/t3lib/
Copy  tslib/          ->  [your_site]/tslib/
Copy  tslib/media     ->  [your_site]/media/
Copy  tslib/index_ts.php ->  [your_site]/index.php
Copy  tslib/showpic.php ->  [your_site]/showpic.php
```

**It's very important to make sure all the old source-code is substituted correctly! We don't want bug-reports because and old t3lib/ was not removed and substituted with the new or something like that.**

After making sure all the source files are in place, you basically need to follow the steps 1-6 as outlined in details above. Here I'll quickly list them:

1. (Make backup if you please)
2. (Install source - which you just did by the above steps....)
3. **Update the database structure** by COMPARE-ing the "**tables.sql**" file in Install Tools/"2: Database analyser"
4. **Import the static tables** by clicking IMPORT at "**static+adt.sql**" in Install Tools/"2: Database analyser". You might also import the description tables by clicking IMPORT at "sys\_descrtables....sql" file, but this is not so urgent.
5. **Clear all cache!**

Of course always check the changelog.txt file in the root of the source distribution. In particular pay attention to lines prepended with !!! - those may have important information for you!

**Notice:**

The same procedure takes place when you upgrade "new versions" of Typo3 - not only previews. But with the new versions you may skip step 2 above ("Compensate for the ignored symlinks...") because that is already done inside the .zip-distribution of the source.

# Upgrading from Typo3.1 to 3.2

## Was Typo3.1 ever released?

No.

... well, for a year it worked stable as a "beta" and finally "rc" version, but the "final release" never happend because ambitions for the final product still remained unfulfilled.

However jumping to Typo3.2 is done because a lot of basic stuff has changed and everything is cleaned up and redefined. That's change enough to denote it with a version jump.

Those changes are outlined in the change-log, 'changes.txt' and the log concerning the update of naming in the code found in 'changelog\_conversion\_to\_3.2.txt'

The pratical upgrade consequenses are outlined here.

## How bad is it?

Typo3 3.2 has been changed in numerous ways and that requires you to take extraordinary actions in the process of upgrading. We're sorry, but it's all done in order to streamline the product and clear out the concepts of inconsistent functionality. So this is why it all comes at one time - and hopefully not for the next many updates.

## Trivial updates

Before you take these steps into upgrading, you should walk through the regular upgrade procedure as outlined above in this document.

One thing to note here is that the Install Tool referer to in the update section is "disabled" because your localconf.php file must contain this line of code:

```
$TYPO3_CONF_VARS["BE"]["installToolPassword"] = "bacb98acf97e0b6112b1d1b650b84971";
```

So start out putting this line in localconf.php and hit the Install Tool. The password is "joh316" with this line.

## The filesystem

- remove the stdlib/ directory.

*UNIX / symlinks only:*

- Relink the media-directory

```
rm media
ln -s tslib/media/
```

- Relink the tables.php file which is now stored in t3lib/install/tables.php

```
cd typo3conf/
rm ./tables.php
ln -s ../t3lib/install/tables.php
```

## Configuration files

- Update your "typo\_db\_extTableDef\_script" script, if you have one
- The old \$tc array is renamed to \$TCA
- \$pagesTypes is renamed to \$PAGES\_TYPES

- \$modules is renamed to \$TBE\_MODULES

All internal structures are preserved.

## TypoScript syntax change

The TypoScript syntax has changed a little bit. See the changelog.

There are two important things to know:

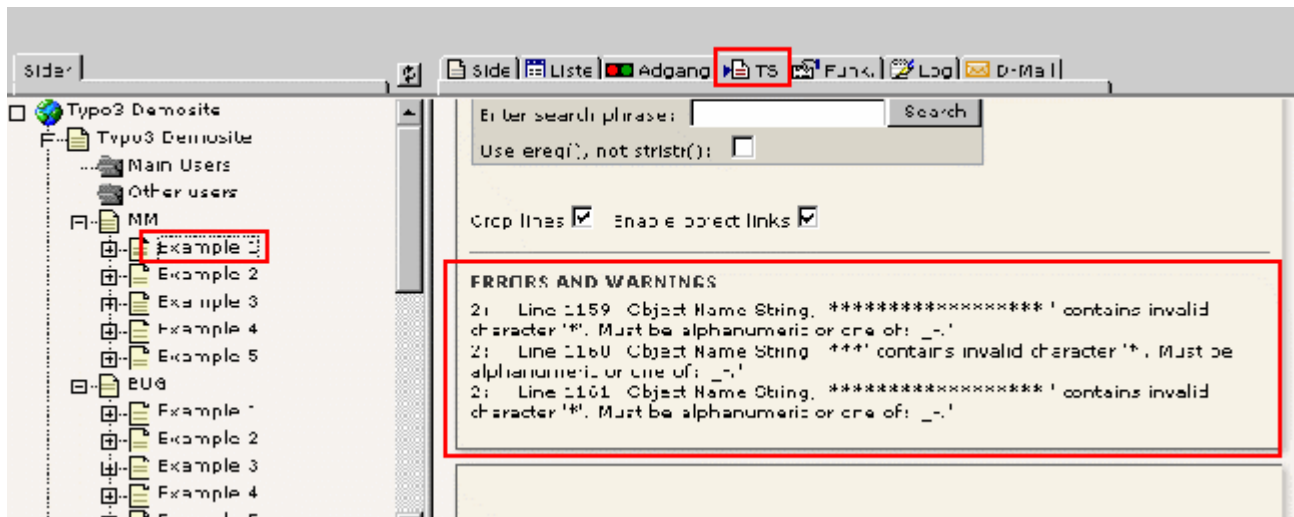
1. You cannot use characters other than a-z, A-Z, 0-9, - and \_ in TypoScript object names. So 'die\_tür.ist.rot' will not work!
2. Copying objects wipes out both old value and properties. By example:

```
1.ACT = 1
1.ACT < .1.NO
```

... will not work, because the value '1.ACT = 1' is removed by the object copy afterwards. Fix such problems simply by reversing the statements:

```
1.ACT < .1.NO
1.ACT = 1
```

You can check for these problems by the TS module:



You can also search through all your template code by dumping your sys\_templates table to a simple text file. Do it by this document:

typo3/dev\_dump\_template\_tables.php

When you've got the text, you can search for all occurrences of NO, ACT, RO copying by this regex:

```
[ ]*[[[:alnum:]]\.\ ]+[A-Z]+[ ]*<
```

## Conformance with php.ini-optimized

Conforming to the php.ini-optimized demands is primarily a question of getting incoming data from the HTTP\_XXX\_VARS arrays and not expect them to be in the global scope.

Also some global vars previously accessible globally are now put into objects in the front end.

For instance, you should not rely on the id of the current page to be accessible as \$GLOBALS["id"]. Now it's moved into the global object, \$TSFE. So get it from \$GLOBALS["TSFE"]->id

Please see "Inside Typo" in the Code Guidelines section for details on all the new naming conventions. Also see the changelogs

In TypoScript you may refer to global vars in two ways:

1. Getting it by the "data" datatype:

```
something = TEXT
something.data = global : id
```

Search for this regex in the dumped sys\_template table:

```
=[:space:]]*global
```

2. Using globals in conditions:

```
[globalString= todo=email]
```

For instance this should be converted to:

```
[globalString= GP:todo=email]
```

.. which means the "todo" variable is expected input from HTTP\_POST\_VARS or HTTP\_GET\_VARS

## on-site

1. FE-users must be in one folder!

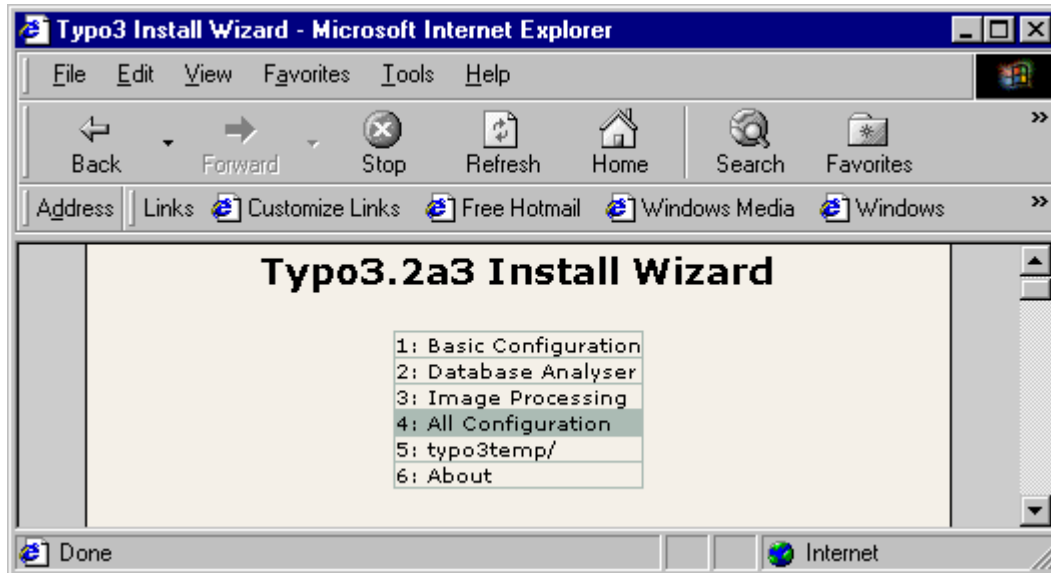
It's now possible to have more FE-users (fe\_users) in the same physical database with the same usernames. In order for the login to find the correct user, you must send the uid-number of the sysFolder containing the users of your site.

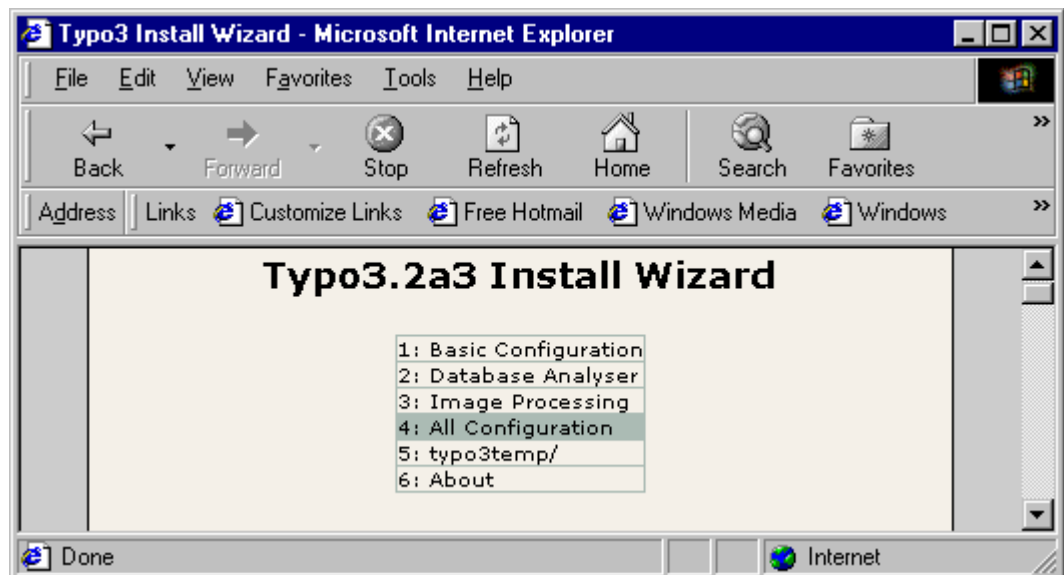
- You can do it per-template by a TypoScript *constant*:

```
styles.content.loginform.pid = (insert the pid here!)
```

- You can define it by adding a second page to the "Send to page" field in the content element "Login" (the first element is the page the login is 'sent to').

- or you can disable the feature in localconf.php:





2. Reset user settings!

You should reset the user settings of the backend users.

- Go to the "Database Analyser"
- Click "Reset user preferences" and go through the action.

## PHP\_SCRIPT include-scripts, libraries and backend modules.

Updating your self-written PHP-script may be the biggest adventure of the new version.

Basically it's all about changed names of functions, classes, variables, filenames- and locations. Therefore the changelog for conversion shows a lot of regex's which are an almost complete list of what was changed and which regular expression was used to search and replace.

This is a few of the most useful regex's which you may/may not find helpful to either search/replace or just check your converted scripts.

Before you update your scripts, do:

1. Read the "Project Code Guidelines" in "Inside Typo3" and consider adhering to them. They also reflect which concept is the main goal of the whole cleaning process up to version 3.2
2. Make sure you have at least briefly consulted the changelog, conversion-log and this list below:
3. Consider using the new USER cObject instead of PHP\_SCRIPT. Or better yet, the USER\_INT object (if you please). Please read TSref\_xxx.pdf for details.

(Searching: Unless stated, the search/replace strings are a regular expression. These work with Allaire Homesite 4.52. Also the substitution is done in .php and .inc files. Search is case-insensitive by default. You should always commit a search without replace first to check the instances found. Also make a backup of your scripts before you do these things.)

Classes has changed names, check any objects, you create and change the names of the class they are created from:

```
--search
(class|extends|new)[[:space:]]+[[:alnum:]]_*
--action
replace name if necessary
```

Functions from the old stdlib/ are put into a class, t3lib\_div. You might use some of these, so search for that and replace if needed.

```
--search
--search
([^\A-Za-z0-9_>:])(milliseconds|gif_compress|png_to_gif_by_imagemagick|checkbrowser|fixed_lgd|fixed_lgd
```

```

_pre|hardspace|break_text|unbreak_text|inList|inArray|rmFromList|intInRange|intval_positive
|intExplode|trimExplode|extParseUrl|split_fileref|modifyHTMLColor|rm_endcomma|danish_strtou
pper|split_attributes|splitCalc|uniqueArray|uniqueList|shortMD5|testInt|isFirstPartOfStr|fo
rmatSize|stripslashes_array|get_tag_attributes|split_tag_attributes|imagefinder|linkfinder|
getURL|getStrippedURL|getMimeType|writeFile|get_dirs|debug_white_space|debug_ordvalue|debug
_code|parsetime|view_array|print_array,GPvar)[[:space:]]*\(\
--Replace with:
<lt3lib_div::\2(

```

```

replace $GLOBALS["typo_db"] with the new constant, TYPO3_db
--search, case-sensitive:
(\$GLOBALS\[[[:space:]]*"|\$)typo_db([']?[[:space:]]*\)\)
--replace
TYPO3_db
--check it (regex): This will find all mysql()-function calls that still use a variable as
database. This might be an error unless it's a database name of your own.
mysql[[:space:]]*\([[:space:]]*\$

```

```

check if you use any old global vars, which are now constants:
--search
stdlib_path|typo3lib_path|config_path|sitePath|TSlib_path|modPath|typo3Path|sitePath|tempPa
th
--action
Replace them with the proper constants, see changelog_Typo3.2

```

```

GLOBAL Vars, most of which moved into the global object $TSFE
--search, case sensitive
\$GLOBALS[[:space:]]*\([[:space:]]*"?(register|all|BE_USER_LOGIN|TYPO3_CONF_VARS|id|root
Line|page|sys_page|no_cache|fe_user|loginUser|gr_list|type|BE_USER_LOGIN|additionalHeaderDa
ta|JSCode|JSImgCode|typo3_DIV_SECTION|tmpl|show_hidden_page|FE_PREVIEW|cache_timeout_defaul
t|content|config|cacheContent_flag|sPre|pSetup|newHash|jumpurl|intTarget|extTarget|absRefPr
efix|compensateFieldWidth|lockFilePath|ATagParams|config|SWORD_REGEX|sword_list|linkVars)['
]?[[:space:]]*\)
--replace
$GLOBALS["TSFE"]->\1
--check it (regex): This checks if you use any "global ..." definitions in functions.
Manually check if some of the former global vars were defined there
global[[:space:]]*\$

```

```

--NOTICE: In addition to being moved into the TSFE object, some vars has changed names as
well. For the authoritative guide, see inside the file class.tslib_fe.php. Here's a few:
BE_USER_LOGIN -> beUserLogin
typo3_DIV_SECTION -> divSection
show_hidden_page -> showHiddenPage
FE_PREVIEW -> fePreview
cache_timeout_default -> cacheTimeOutDefault
cacheContent_flag -> cacheContentFlag
SWORD_REGEX -> sWordRegEx
sword_list -> sWordList
(There's a check of this later anyway...)

```

```

Remove/redefine timetracking.
--search
\$GLOBALS[[:space:]]*\([[:space:]]*"?(tt)[']?[[:space:]]*\)
--action
The timetracking utility works a bit different now, so check that out or just remove your
entry

```

```

cObj goes into TSFE:
--search
(\$GLOBALS\[[[:space:]]*"?(cObj)[']?[[:space:]]*\)([^\A-Za-z0-9_])
--replace
$GLOBALS["TSFE"]->cObj\3

```

```

$be_user converted to $BE_USER;
--search
(\$GLOBALS\[[[:space:]]*"?(be_user)[']?[[:space:]]*\)([^\A-Za-z0-9_])
--replace
$GLOBALS["BE_USER"]\3

```



Find instances of use of \$tc array and replace with \$TCA

```
--search
(\\$GLOBALS\\[[[:space:]]*["']?)tc(["']?[[[:space:]]*\\])([^A-Za-z0-9_])
--replace
$GLOBALS["TCA"]\\3
--check, search case-sensitive:
\\$tc([^A-Za-z0-9_])
```

--search, check:

```
(BE_USER_LOGIN|FE_PREVIEW|GET_URL_ID_TOKEN|show_hidden_page|cache_timeout_default|cacheContent_flag|no_cache_beforePageGen|typo3_DIV_SECTION|SWORD_REGEX|sword_list|cobj_depth_counter|ACCESSKEY|IMAGES_ON_PAGE|LAST_IMAGE_INFO|parsetime_start)
--action
If any of these are use, notice that they are moved into the global object TSFE and they are renamed as well. Look in the class tslib_fe to see their new name (or check the changelog-Typo3.2)
```

--search, checking if any of these are used by your script.

```
convertPathFromWindows|PHPVER|WHICH_OS|contentTable|sitename|DDMMYY|HHMM|typo_db|sim_exec_time|exec_time|cobj_depth_counter|recordRegister|currentRecord|TSPath|IMAGES_ON_PAGE|LAST_IMAGE_INFO|currentRecordTotal|currentRecordNumber|stype|scols|sword|scount|spointer|ACCESSKEY
--action
If any matches, check out the code and find out (see the changelog-Typo3.2) what has changed and what you need to do about that.
```

Converting your use of global vars from GET or POST method into using tslib\_div::GPvar() to fetch it.

This search finds all \$GLOBALS[.. instances with alphanumeric names that does NOT start with an upper-case letter. This excludes HTTP\_POST\_VARS/GET\_VARS and other system globals + TSFE, the new front-end global. But if you use any globals in the url which are upper-case they are not found either!

```
--search, case SENSITIVE
\\$GLOBALS\\[[["']?([^A-Z"'])[[:alnum:]]_]*["']?\\]
--replace
tslib_div::GPvar("\\1")
--check (see all globals which are probably not TSFE) search, case sensitive
\\$GLOBALS\\[[["']?[^T"']
```

--search

```
set_no_cache|set_cache_timeout_default
--action
```

if these are found, prepend \$GLOBALS["TSFE"]->, BUT maybe you should consider using the new PHP\_SCRIPT\_INC instead which enables you to combine cached page-content with dynamic content.

Recommended use of php is that the split() function is replaced with explode() (which now has a third parameter!) and mysql\_fetch\_array() is converted to mysql\_fetch\_assoc() (which does not return num keys in the array). You are not forced to do this substitution.

```
split()
--search
([^A-Za-z0-9_>:|)]split([[[:space:]]*\\)([[[:space:]]*["']
--action
Manually replace if possible
```

mysql\_fetch\_array() -> mysql\_fetch\_assoc()

```
--search
([^A-Za-z0-9_>:|)]mysql_fetch_array([[[:space:]]*\\)(
--replace
\\mysql_fetch_assoc\\2(
```

--search (environment variable that should go into getenv())

```
(\\$GLOBALS\\[[[:space:]]*["']?|\\$)(QUERY_STRING|HTTP_ACCEPT|HTTP_ACCEPT_ENCODING|HTTP_ACCEPT_LANGUAGE|HTTP_CONNECTION|HTTP_COOKIE|HTTP_HOST|HTTP_USER_AGENT|REMOTE_ADDR|REMOTE_HOST|REMOTE_PORT|SERVER_ADDR|SERVER_ADMIN|SERVER_NAME|SERVER_PORT|SERVER_SIGNATURE|SERVER_SOFTWARE|GATEWAY_INTERFACE|SERVER_PROTOCOL|REQUEST_METHOD|SCRIPT_NAME|PATH_TRANSLATED)(["']?[[[:space:]]*\\]|)([^A-Za-z0-9_])
--replace
getenv("\\2")\\4
```

**Backend modules:**

This is more tricky and includes more changes to be aware of. This is a summary of a few things to check:

conf.php file:

```
$TYPO3_MOD_PATH -> define("TYPO3_MOD_PATH", "mod/file/images/");
$backPath -> $BACK_PATH
$conf -> $MCONF
$lang -> $MLANG
```

locallang.php file:

```
$localLang -> $LOCAL_LANG
```

The module files:

```
'$BE_func->' is not an object anymore. Use 't3lib_BEfunc::'
$be_user -> $BE_USER
$lang -> $LANG
$template -> $TBE_TEMPLATE
$browser has changed
typo3lib() is not available. Use include_once().
Use the "PATH_" constants
$conf -> $MCONF
$tc -> $TCA
$backPath -> $BACK_PATH
Incoming data should be registeret global with t3lib_div::setGPvars()
```

# The Install Tool

## Aimed to end newbie agony...

The Install Tool is provided in order to simplify installation and bug-tracking.



## Install Tool Password

You must enter a password to access the install-tool. The password is set in localconf.php:

```
$TYPO3_CONF_VARS["BE"]["installToolPassword"] = "bacb98acf97e0b6112b1d1b650b84971";
```

If you insert this line in localconf.php, the password to the Install Tool is 'joh316'. Changing the password can be done from inside the tool.

```
$TYPO3_CONF_VARS["BE"]["installToolPassword"] = "";
```

Setting the install tool password to empty (default) the tool will exit immediately with the message "Install Tool deactivated".

Please read the security warning in the Install Tool and take proper action. If you're in doubt, simply remove the script typo3/install/index.php

# Installing a Typo3 webserver on RedHat 6.1

(S1)

## Disclaimer / NOTICE THIS!

This is some relatively old notes on installing Apache/PHP/MySQL/GDlib/ImageMagick on RedHat 6.1 and provided 'as is' and not updated anymore!

If you're trying to prepare another server, you might find this document insufficient. Please respect that our main target is Typo3 and NOT installation of PHP. However there are lots of information in the mailing list archive. Please search that for more help!

I will not take responsibility for anything that results from your use of this installation guide. Like with everything else, it's on your own risk.

But especially I would like to point out, that this is my own way of configuring a Typo3 Webserver on LINUX. And this document may help you or it may not. If you have any comments - because your webserver is somehow different - please post them on the Typo3.com website (<http://www.typo3.com/Hints.1169.1.html>).

**I don't intend to support webserver installation** - I just wanted to share this information with you in case you might find it useful.

## Hardware

You need a hardware setup in the neighbourhood of this:

Pentium "as-fast-as-possible" (III, min: 200 MHz)  
Min 128 MB RAM  
Enough Harddisk (few GB)

## Software

Redhat 6.x

Apache 1.3.+

MySQL 3.2+

PHP4.0.3pl1+

joe editor                   joe-2.8-22.i386.rpm

**RedHat note:** You shouldn't use RedHats older than 6.1! Not even 6.0! This is due to the success of compiling freetype / installing ImageMagick. RedHat 6.1 works. You can possibly use RedHat 6.1+

**The ImageMagick problem:** ImageMagick is VERY sensitive regarding versions! You MUST be sure to use the right versions and probably also the RPM's distributed from [www.typo3.com](http://www.typo3.com). This is due to the fact that ImageMagick is filled with bugs some of which are fixed in new versions and other introduced by new versions! So use an "authorized" RPM.

ImageMagick is installed as two RPM's. The reason why is, that version 4.2.9 works for almost all image manipulation. BUT the RPM is compiled without LZW-compression for gif-files (due to the copyright issues with the Unisys patent). Should anyone have a 4.2.9-rpm with LZW enabled, I would gladly receive it!

As Typo3 is primarily based on gif-files we can't just skip them yet (PNG is supported now). Gif files are produced by ImageMagick 4.2.9 without LZW and in PHP with GDlib (installed in version 1.3) which uses RLE-compression.

The solution to the problem is, that a newer version (5.1.0) of ImageMagick with LZW enabled(!) is also installed and any generated GIF-file is finally compressed with version 5.1.0.

An alternate solution could be to compile PHP4.x with GD1.2 (which uses LZW) and clear the variable \$PATH\_ImageMagick\_LWZ found in "typo3conf/localconf.php". If this variable is cleared, the function

gif\_compress(\$theFile, \$type) in stdlib will accept GD to compress GIF-files properly.

ImageMagick version 5+ can be used alone but is not recommended due to a decrease in speed and the lack of fast blurring and sharpening functions (which makes dropshadows impossible)

ImageMagick is used by Typo3 by calling "convert", "combine" and "identify" with an exec()-call from PHP.

These executables is (after installing the RPM's) located in:

Version 4.2.9:                /usr/X11R6/bin/

Version 5.1.0:                /usr/bin/

Typo is by default setup to use ImageMagick from these locations (else change typo3conf/localconf.php)

## Paths

I use "/usr/local/" to install and compile the software. This is pretty much standard.

The MySQL databases will be installed in "/usr/local/var/" as default

The webserver, Apache, will be installed in "/www/"

## Installing

### Redhat 6.1

- Choose "Custom Install"

- Define your partitions as you like. The webroot is in "/www/" and the MySQL-databases are by default located in "/usr/local/var/". You might also consider a separate partion for "/tmp/"

- Install at least these packages:

- Graphics Manipulation
- Networked Workstation
- Anonymous FTP
- Development
- Extra Documentation
- Utilities

Note: You could probably skip some packages, but you run the risk that ImageMagick or freetype will not install right-away if some packages are missing. To be sure, install everything and disable useless services afterwards.

### Joe (editor):

I like to use the editor, joe. Install it:

```
rpm -i joe-2.8-22.i386.rpm
```

Now set joe as the default editor (usefull if you edit crontabs later on by "crontab -e")

```
joe /etc/profile
```

... and insert the following lines:

```
EDITOR=/usr/bin/joe
export EDITOR
```

### FreeType:

I have had a lot of troubles with the freetype library (used to create stuff like text on images). I have found that installing these two RPM's helps a lot (the first is for the sake of ImageMagick, while the last is for PHP).

As I'm not an expert into this, I might be unprecise in my conclusions here, but I send a little prayer each time I configure PHP, hoping that freetype will be recognized, and these RPM's seems to do the trick together with a compiled version.

```
rpm -i freetype-1.2-7.i386.rpm
rpm -i freetype-utils-1.2-7.i386.rpm
```

An now the compiled version: Last time I did an installation (RH 6.2) I also had to compile freetype, so if you have trouble with php-configure to recognize freetype, do this:

```
cd /usr/local
tar xzf freetype-1_2_tar.tar
cd freetype-1.2
./configure
make
make install
```

There has been reports that a "devel"-rpm is also needed in some cases. Basically the freetype library is recognized correctly by PHP-configure if you untar it, and compile it like stated here. Then point to the path with the ./configure of PHP later. Example:

```
./configure --with-gd --with-ttf=../freetype-1.3 --with-mysql --with-apa...
```

**If you're having trouble** with the compilation, be sure that these packages are installed:

```
compat-glibc-5.2-2.0.7.1.i386.rpm
glib-1.2.5-1.i386.rpm
glib-devel-1.2.5-1.i386.rpm
glib10-1.0.6-6.i386.rpm
glibc-2.1.2-11.i386.rpm
glibc-devel-2.1.2-11.i386.rpm
glibc-profile-2.1.2-11.i386.rpm
```

Another source of trouble is, if XFree86 / X11 is not installed. This is a major problem when freetype is about to be compiled!

## ImageMagick

ImageMagick seems to install without problems if you installed RedHat 6.1 and chose the recommended packages (or all).

```
rpm -i ImageMagick-4.2.9-1.i386.rpm
rpm -i ImageMagick-5.1.0-1.i386.rpm
```

If you have any trouble with these RPM's, try to install this (necessary with RedHat 6.0) first:

```
rpm -U bzip2-0.9.5c-1.i386.rpm
```

## MySQL

Install MySQL. This is the most time-consuming task. Not that you have to do a lot of typing, but the "make"-step might take up to halv an hour.

Unpack:

```
cd /usr/local
tar xzf mysql-3.22.27.tar.gz
cd mysql-3.22.27/
```

Configure / Compile / Install:

```
./configure
make
make install
```

Let this script install the databases  
scripts/mysql\_install\_db

Start MySQL

```
/usr/local/bin/safe_mysqld &
```

Installs the root-user with password "mysql\_password" in the database.

```
/usr/local/bin/mysqladmin -u root password mysql_password
```

Optional:

Copy the mysql-startscript. Edit this script!!

```
cp support-files/mysql.server.sh etc/rc.d/init.d/mysql
```

Note: "mysql\_password" is a password you choose. IT SHOULD NOT BE THE REAL ROOT-password on the machine.

## PHP4 / Apache

```
cd /usr/local
tar xzf apache_1.3.x.tar.gz
tar xzf php-4.0.x.tar.gz
```

Read the instructions for PHP4 ("less php-4.0.0/INSTALL"). This is what it says:

```
Apache initialized:
cd apache_1.3.x/
./configure
cd ..
```

```
PHP:
cd php-4.0.x/
./configure --with-gd --with-ttf --with-mysql --with-apache=../apache_1.3.x --enable-track-vars
make
make install
cd ..
```

```
Apache
cd apache_1.3.x/
./configure --prefix=/www --enable-module=rewrite --activate-module=src/modules/php4/libphp4.a
make
make install
cd ..
```

```
PHP finish
cd php-4.0.x/
cp php.ini-dist /usr/local/lib/php.ini
```

( You should set the path of sendmail in php.ini: Normally "/usr/sbin/sendmail")

NOTE: after the step with "./configure --with-gd ....." scroll back in the text and check if "ttf support" is enabled!! You could also pipe it into grep by adding this to the line: " | grep ttf"  
After a while a line should be output saying that ttf-support is enabled!

## Apache user/group

I personally use the user "httpd" for Apache. Do this if you like:  
adduser httpd

## Configure Apache (/www/conf/httpd.conf)

Change "/www/conf/httpd.conf":

```
joe /www/conf/httpd.conf
```

Add:

```
AddType application/x-httpd-php .php .php3
```

Set user / group to httpd (if you want this!)

You should out-comment "UserDir public\_html" (then users cannot access the webroot. Security!)

Change:

```
DirectoryIndex index.html index.php index.php3
```

Change

```
IndexIgnore * ( No directory browsing... Security!!)
```

Change:

(enabled "HostnameLookup On" - used to associate IP-numbers with servers in Typo3. Optional, but nice to enable!)

## Start Apache

```
/www/bin/httpd
```

You should now get a nice message that tells you, Apache is running.

Now, create a file in /www/htdocs/index.php:

```
joe /www/htdocs/index.php
```

Add:

```
<? phpinfo(); ?>
```

Hit the server with a browser. You should see a PHP-info page.

On this page, check:

- that GDlib is activated (Ver 1.3) with truetype support
- that MySQL is active
- that the sendmail-path is set (if you changed that...)

Congratulations!

Now read the readme.txt file that followed the Typo3-tarball. This will instruct you to install the Typo3-scripts.