

TYPO3: Autoloader explained

Since version 4.3, [TYPO3](#) has an own [autoloader](#). Currently, documentation for it is lacking and I'll try to explain it a bit.

The autoloader code is available in `t3lib/class.t3lib_autoloader.php`.

Table of contents
TYPO3 4.3
Extension name calculation
ext_autoload.php
TYPO3 4.6
Caching
Autoloading ext_autoload.php
Extbase class loading
Creating an ext_autoload.php file
No autoload file
Manually
extbase extension utility
Extension Development Evaluator
Links

TYPO3 4.3

The autoloader was introduced in version 4.3 and didn't really change in 4.4 and 4.5. Have a look at the [4.3 source](#).

Autoloading process:

1. Determine extension name from class name
2. Load extension's `ext_autoload.php` file
3. Load file for class as described in `ext_autoload.php`

Extension name calculation

In short: The lowercased second part of the class name needs to equal the extension name without `"_"`.

Example: Class `Tx_BlogExample_Controller_BlogController` belongs to extension `blog_example`, because a lowercased `BlogExample` equals `blog_example` without underscores.

Note that the mapping only works for activated extensions.

The code that does this is in [t3lib_extmgm::getExtensionKeyByPrefix\(\)](#) and

[t3lib_autoloader::attemptToLoadRegistryForGivenClassName\(\)](#) ↗

ext_autoload.php

This file defines a mapping of class names to file paths. Every single class needs to be listed in the mapping array; wildcards are not possible. Class names need to be **lowercase**.

An example:

```
<?php
$extPath = t3lib_extMgm::extPath('nr_semantic_annotations');
$srcPath = $extPath . 'src/NR/SemanticAnnotations/';
$prefix  = 'nr_semanticannotations_';

return array(
    $prefix . 'config'          => $srcPath . 'Config.php',
    $prefix . 'config_tables'   => $srcPath . 'Config/Tables.php',
    $prefix . 'exception'       => $srcPath . 'Exception.php',
    $prefix . 'helper_typo'     => $srcPath . 'Helper/Typo.php',
);
?>
```

Updating the file will have immediate effect, at least up to TYPO3 4.5.

The file does not need to be registered anywhere. TYPO3 just finds it by its name.

TYPO3 4.6

The behavior of [t3lib_autoloader](#) ↗ in version 4.6 changed quite a bit compared to the previous versions:

1. Class mapping information gets cached
2. `ext_autoload.php` for each active extension is loaded unconditionally on startup
3. Files for extbase-style class names are loaded automatically without an explicit mapping in `ext_autoload.php`.

Caching

TYPO3 will not notice if you change `ext_autoload.php`, because it caches the class-to-filename mapping now.

Either clear the cache in the TYPO3 backend, or just do the following in your shell:

```
$ rm typo3temp/Cache/Code/cache_phpcode/*
```

Autoloading `ext_autoload.php`

Previously, an extension's `ext_autoload.php` file was only loaded if a class was requested that got mapped to the extension. With 4.6, each active's extension autoload file gets loaded automatically when the autoloader is initialized.

This means that now you can use arbitrary class names that have no connection to the

extension name at all.

Extbase class loading

If your extension follows the [Extbase](#) *directory structure and class naming conventions*, you do not need an autoload file at all in TYPO3 4.6!

Quick reminder: A class `Tx_BlogExample_Controller_DefaultController` has its file in `blog_example/Classes/Controller/DefaultController.php`.

See the nitty-gritty details in the extbase documentation sources, chapter [Class naming conventions](#). Sorry for the docbook link; I did not find a HTML version of that file.

Creating an `ext_autoload.php` file

No autoload file

When you're on TYPO3 4.6 upwards and follow the extbase class naming conventions, you do not need an `ext_autoload.php` file. The autoloader's magic will find them.

Manually

If you have only a couple of classes, creating the autoload file manually is an option.

The file *returns* an array, which has *lowercased* class names as key and the full file paths as value. See the [example](#) earlier in this article.

extbase extension utility

Somewhere in your code add the following lines. Use them only if you need them, not permanently.

```
<?php
Tx_Extbase_UTILITY_Extension::createAutoloadRegistryForExtension(
    $extensionKey,
    t3lib_extMgm::extPath($extensionKey)
);
?>
```

Note that this works for all kinds of classes and filenames, not only extbase styled ones.

Extension Development Evaluator

The [extdeveeval extension](#) has a "Generating `ext_autoload.php`" mode to assist creating your autoload file.

It only registers classes whose name begins with `tx_extensionname`, `Tx_extensionname` or `user_extensionname` - all others are ignored.

Links

Other pages talking about the TYPO3 autoloader:

- [TYPO3 Autoloader verwenden](#)↗ (german)

Comments? Please [send an e-mail](#).