

Core - Feature # 25337

Status:	Resolved	Priority:	Should have
Author:	Xavier Perseguers	Category:	
Created:	2011-03-16	Assigned To:	
Updated:	2012-08-26	Due date:	
PHP Version:	5.3		
Complexity:			
Sprint Focus:			
Subject:	Add PHP 5.3 namespace support		
Description	<p>As we target PHP 5.3 for TYPO3 4.6, Core should support namespaces:</p> <ul style="list-style-type: none">- t3lib_div::makeInstance- various autoloaders <p>(issue imported from #M17964)</p>		
Related issues:			
related to TYPO3.Flow - Feature # 27490: Vendor Namespaces Support	Resolved	2011-06-16	
related to Core - Feature # 21144: Remove ext_autoload.php files for extensio...	Closed	2009-09-25	
related to Core - Task # 40095: Move core to namespaces	Resolved	2012-08-23	2012-09-18

History

#1 - 2011-06-29 06:44 - Xavier Perseguers

Once support has been added, we should be clear about the new naming guideline for extension classes and support it in the various places (I basically think of user_ thingy for USER/USER_INT).

#2 - 2011-06-29 07:04 - Xavier Perseguers

Extract from post on v4 mailing list:

We now have a dependency to PHP 5.3 for TYPO3 4.6 but still, one of the basic features of PHP 5.3 used by FLOW3 is [namespaces](#).

FLOW3 is currently reviewing a change to support vendor namespaces [#27490] thus making the "F3" namespace obsolete:

```
\RobertLemke\AutoMakeTemplate\Controller\FooController
\Zend\...
\TYPO3\FLOW3\Package\PackageManagerInterface
\TYPO3\TYPO3\Domain\Model\Site
```

Currently we have pseudo namespaces with our class naming prefix (I take the Extbase variant with camel case):

```
"Tx_" extension_key "_" directory [ "_" directory ... ] "_" class_name
```

Example:

```
extbase/Classes/MVC/Controller/ActionController.php gives
```

Tx_Extbase_MVC_Controller_ActionController

Without discussing here how it should be named CGL-wised, the purpose of adding namespace support for 4.6 is to be able to name it e.g.,

\TYPO3\Extbase\MVC\Controller\ActionController

and instantiate it with

```
/** @var ActionController \TYPO3\Extbase\MVC\Controller\ActionController */
$actionController = t3lib_div::makeInstance('\TYPO3\Extbase\MVC\Controller\ActionController');
```

If new to namespaces, the [FAQ entry](#) on how a name is resolved when using a backslash at the beginning or not is worth reading.

#3 - 2011-06-29 14:27 - Robert Lemke

I can't really oversee the implications for TYPO3 v4 in detail but here's my guess:

Refactoring the core to **use** namespaces is probably not realistic, especially not for TYPO3 4.6. But that is, as far as I understood, also not Xavier's point.

Refactoring the core to **support** extension code using namespace is far more realistic. But you'll certainly have to implement a dual-support giving 100% backwards compatibility. While I refactored FLOW3 to support vendor namespaces, I stumbled over a lot of parts having something to do with it which I did not expect. Although FLOW3 was using namespaces already, quite a big amount of code had to be changed. The class loader was the smallest problem.

In FLOW3 we have a lot of code somehow processing class names and / or package keys. Fluid for example uses the package key to determine template paths. The MVC framework was parsing class names to figure out where to find the controllers, which prefixes to use for GET / POST arguments and so on. So you better be prepared to discover some oddities while adding namespaces support. And I didn't even start thinking about weird things extensions do.

Anyway, if I can give advice on that at all, it is to yes, try to implement that, but better don't do it for TYPO3 4.6.

#4 - 2011-06-29 16:12 - Xavier Perseguers

- File 25337_draft.diff added

Basic implementation added as draft patch...

How to test:

```
# file yourextension/Classes/Test/Foo.php
```

```
namespace T3\yourextension\Test;
```

```
class Foo {
    public function sayHello() {
        return 'Hello World!';
    }
}
```

```
# in another extension class:
```

```
$foo = t3lib_div::makeInstance('\T3\yourextension\Test\Foo');
t3lib_utility_Debug::debug($foo->sayHello());
```

if using same namespace in other class, relative class name is supported too:

```
namespace T3\yourextension\Test;

class FooBar {

    ...

    $foo = \t3lib_div::makeInstance('Foo');
    \t3lib_utility_Debug::debug($foo->sayHello());

}
```

and register it in your ext_autoload.php file "as usual":

```
return array(
    ...
    't3\yourextension\test\foo' => $extensionPath . 'Classes/Test/Foo.php',
    't3\yourextension\test\foobar' => $extensionPath . 'Classes/Test/FooBar.php',
);
```

#5 - 2011-07-15 13:20 - Stefan Neufeind

Status by Steffen Ritter on the mailinglist:

We indeed got very close on the dev days - the automatic zend conversion tool needed much less adaption then we thought - but it is not perfect yet. TYPO3 is so far away from Zend naming conventions and directory structure that it does not completely work without manual work.

Even if I think it would be doable until FF (as we currently already have a very good state) it would be such a big rebasement, I don't want to rush with that one.

As we have thought of forward and backwards compatibility layers and so on, we currently really are optimistic that it would work out quite nicely, but give it time to test it out in all thinkable variants.

#6 - 2011-08-02 17:56 - Xavier Perseguers

- Target version deleted (4.6.0-beta1)

#7 - 2012-07-19 14:04 - Stefano Kowalke

Are there any news here? How is the status? I have a suggestion to this topic.

#8 - 2012-07-19 14:14 - Ernesto Baschny

Stefano, please take a look at the typo3.teams.core list / newsgroup. In the thread "Minutes of the 6.0 release team meeting" Thomas Maroschik just gave us a status update of namespace support in TYPO3.

You might want to continue the discussion there. Thanks!

#9 - 2012-08-26 02:34 - Christian Kuhn

- Status changed from New to Resolved
- % Done changed from 0 to 100
- TYPO3 Version changed from 4.5 to 6.0

This is solved with #40095

Files

25337_draft.diff	2.5 kB	2011-06-29	Xavier Perseguers
------------------	--------	------------	-------------------