

## TYPO3.Flow - Feature #459

### Implement dependency injection via abstract factory methods

2008-04-16 10:59 - Robert Lemke

<b>Status:</b>	Closed	<b>Start date:</b>	2008-04-16
<b>Priority:</b>	Should have	<b>Due date:</b>	
<b>Assignee:</b>	Robert Lemke	<b>% Done:</b>	0%
<b>Category:</b>	Object	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Complexity:</b>	
<b>PHP Version:</b>			
<b>Has patch:</b>			

#### Description

Dependency injection works fine so far as long as components in a singleton scope are requested. Prototype components can't be injected as soon as more than one instance is needed or the requesting class / method needs to have control over the constructor arguments (which is very often the case).

Therefore consumer code currently calls `$this->componentManager->getComponent('NameOfPrototypeComponent')` in order to get a fresh instance.

This can be optimized by moving the instantiation into an (abstract) factory method:

```
abstract protected function createSomeComponent($argument1, $argument2);
```

This method should be detected by the component manager and automatically be implemented by creating a proxy class.

#### History

##### #1 - 2008-07-01 15:35 - Robert Lemke

- Target version deleted (45)

##### #2 - 2008-07-23 14:16 - Robert Lemke

- Status changed from Accepted to Closed

- Estimated time set to 0.00 h

Dropped the idea for now (see mailing list discussion about component factory).