TYPO3.Flow - Feature #10

Register component types automatically from interfaces

2007-02-20 11:58 - Robert Lemke

Status:	Closed	**Start date:**	
Priority:	Should have	**Due date:**	
Assignee:	Robert Lemke	**% Done:**	0%
Category:	Object	**Estimated time:**	0.00 hour
Target version:	0.1		
PHP Version:		**Complexity:**	
Has patch:			

Description

From *The Guide*:

Component names and types

By default, the name of a component is identical to the PHP class which implements the component's functions. A class called T3P_MyPackage_MyImplementation will be automatically available as a component with the exact same name. Every part of the system which asks for a component instance with a certain name will therefore - by default - get an instance of the class of that name. It is possible to replace the original implementation of a component by another one. In that case the class name of the new implementation will naturally differ from the component name which stays the same at all times.

If the component name is the same as the name of a PHP interface, it is often referred to as a component type. An interface called T3P_MyPackage_MyInterface will be available as a component name as long as there exists at least one class in the same package implemententing that interface.

Associated revisions

Revision 130 - 2007-02-20 18:39 - Robert Lemke

The code result of my todays work:

- Reflection: T3P_Reflection_Class used the component manager to get an instance of the DocCommentParser which possibly is not registered as component itself when reflection is needed. Fixed that.
- Core: Renamed T3P_Core_ComponentManager->registerComponentClass() to T3P_Core_ComponentManager->registerComponent()
- Core: Added a new feature which automatically registers interfaces as component types and sets the class name to the default implementation if one is found. Fixes #10
- DataAccess: Just modified the configuration so it uses the component type T3P_Log_LoggerInterface instead of the specific component name
- Testing: Removed a semi-colon ;-)

History

#1 - 2007-02-20 11:58 - Robert Lemke

- Status changed from New to Accepted

#2 - 2007-02-20 12:00 - Robert Lemke

Component types

Any interface found in the Classes/ directory will equally be registered as a component if at least one class within the same package was found that implements this interface. The first class found in the package which implements the interface is considered to be the default implementation and the component's className option is set accordingly. Of course it is possible and wanted that the class name is defined explicitly in the Components.conf file or any other configuration source. This is especially important if more than one implementing class exists.

#3 - 2007-02-20 18:39 - Robert Lemke

- Status changed from Accepted to Closed

- Sprint set to fixed

(In r130) The code result of my todays work:

- Reflection: T3P_Reflection_Class used the component manager to get an instance of the DocCommentParser which possibly is not registered as component itself when reflection is needed. Fixed that.

- Core: Renamed T3P_Core_ComponentManager->registerComponentClass() to T3P_Core_ComponentManager->registerComponent()
- Core: Added a new feature which automatically registers interfaces as component types and sets the class name to the default implementation if one is found. Fixes #10
- DataAccess: Just modified the configuration so it uses the component type T3P_Log_LoggerInterface instead of the specific component name
- Testing: Removed a semi-colon ;-)