Zend_Config_Configurable

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

Zend Framework: Zend_Config_Configurable (interface) Component Proposal

Proposed Component Name	Zend_Config_Configurable (interface)
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Config_Configurable (interface)
Proposers	Lars Strojny
Zend Liaison	Matthew Weier O'Phinney
Revision	0.1 - 26 December 2007: Initial submission. (wiki revision: 14)

Table of Contents

1. Overview
2. References
3. Component Requirements, Constraints, and Acceptance Criteria
4. Dependencies on Other Framework Components
5. Theory of Operation
6. Milestones / Tasks
7. Class Index
8. Use Cases
9. Class Skeletons

1. Overview

Zend_Config_Configurable is an interface indicating configuration facilities of a component. It will unify all components which consume Zend_Config objects for their internal configuration.

2. References

- Inversion of Control Containers and the Dependency Injection pattern (used in examples)
- Repository pattern (used in the examples)

3. Component Requirements, Constraints, and Acceptance Criteria

- This interface **will** enforce the way components handle configuration
- This interface **will** unify the way components handle configuration
- This interface allows other Zend components/user defined classes to find out weither an object is configurable
- This interface **is implemented** by various current classes

- This interface **will** direct others how to implement configurability with Zend_Config

4. Dependencies on Other Framework Components

- Zend_Config (for type hinting)
- Zend_Db_Adapter_Abstract
- Zend_Layout
- Zend_Filter_Inflector
- Zend_Controller_Router_Rewrite
- Zend_Controller_Router_Route_Interface
- Zend_Controller_Router_Route_Regex
- Zend_Controller_Router_Route_Module
- Zend_Controller_Router_Route_Static

5. Theory of Operation

6. Milestones / Tasks

- Milestone 1: Get feedback
- Milestone 2: Implement interface Zend_Config_Configurable and commit to incubator
- Milestone 3: Edit implementors to handle configurations the way Zend_Config_Configurable enforces it (in incubator)
- Milestone 4: Edit unit test of the implementers
- Milestone 5: Document Zend_Config_Configurable with good examples

7. Class Index

- Zend_Config_Configurable

8. Use Cases

UC-01

```
/**
 * An example repository object which uses the interface to decide whether
 * to set an config object or not
 */
class Repository
{
    public function retrieveObject(Criteria $criteria)
    {
        $object = $this->_retrieve($criteria);
        if ($object instanceof Zend_Config_Configurable) {
            $object->setConfig($this->getConfig());
        }
        return $object;
    }

    [...]
}
```

9. Class Skeletons

```
interface Zend_Config_Configurable
{
    /**
     * Add configuration object
     *
     * Add a configuration object to the component. Optionally you can specify
     * a configuration section in which the basic configuration object resides.
     *
     * @param Zend_Config $config  Configuration object
     * @param string      $section Section of the configuration (optionally)
     */
    public function setConfig(Zend_Config $config, $section = null);

    /**
     * Get configuration object
     *
     * @return Zend_Config|void
     */
    public function getConfig();

    /**
     * Instanciate component from configuration object
     *
     * @param Zend_Config $config  Configuration object
     * @param string      $section Section of the configuration (optionally)
     * @return object     Instance of the configured component
     */
    public static function configure(Zend_Config $config, $section = null);
}
```

]]></ac:plain-text-body></ac:macro>
]]></ac:plain-text-body></ac:macro>