

Zend_Config Refactoring

<ac:macro ac:name="note"><ac:parameter ac:name="title">Under Construction</ac:parameter><ac:rich-text-body><p>This proposal is under construction and is not ready for review.</p></ac:rich-text-body></ac:macro>

<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 Refactoring Component Proposal

Proposed Component Name	Zend_Config Refactoring
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Config Refactoring
Proposers	Lars Strojny
Revision	0.1 - 26 December 2007: Initial version (wiki revision: 4)

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 suffers from various problems. The current implementation does not distinguish between the storage of the configuration (the configuration source) and the representation in the object. Every config class is representation and adapter at the same time. As the organisation in adapters cuts down the native possibilities the config formats already provide. XML already has xinclude, which is more mature than the Zend_Config extension mechanism. Additionally the value representation of the configuration is "just strings". This could be done much better. The actual targets of this proposal:

- Better differentiation between source adapters and front representation to unify provided output
- The built-in options of a format should tap the full potential
- Keep the simple usage which is currently possible by providing more features

2. References

- [Automatic primitive type detection for Zend_Config](#)

3. Component Requirements, Constraints, and Acceptance Criteria

- Zend_Config adapters **must no longer** extend Zend_Config
- Zend_Config adapters **must** only read the configuration source
- Zend_Config adapters **must** normalize the different input sources
- Zend_Config_Adapter_Xml **must** capable of XIncludes
- Zend_Config_Adapter_Ini **must not** use parse_ini_file() any longer
- Zend_Config **should** represent configuration in the
- Zend_Config **should** manage an arbitrary number of configuration sources at the same time by merging it

4. Dependencies on Other Framework Components

- Zend_Config

5. Theory of Operation

6. Milestones / Tasks

- Milestone 1: Finish proposal
- Milestone 2: Implement prototype in laboratory

7. Class Index

- Zend_Config
- Zend_Config_Adapter_Interface
- Zend_Config_Adapter_Xml
- Zend_Config_Adapter_Ini
- Zend_Config_Adapter_Yaml 
- Zend_Config_Adapter_Db 

8. Use Cases

UC-01

```
$config = new Zend_Config();  
$config->addAdapter(new Zend_Config_Adapter_Ini('config/parameters.xml', 'live'));  
$config->addAdapter(new Zend_Config_Adapter_Xml('config/routes.xml', 'live'));
```

9. Class Skeletons

None yet

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

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