Zend_Config_Db - Nick Daugherty

<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_Db Component Proposal

Proposed Component Name	Zend_Config_Db
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Config_Db
Proposers	Nick Daugherty
Zend Liaison	TBD
Revision	1.0 - 7 May 2010: Initial Draft. (wiki revision: 5)

Table of Contents

1. Overview

Zend_Config_Db is a component for loading Zend_Config objects that have been stored as key/value pairs in a database

2. References

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** extend Zend_Config
- This component **will** use a database table for loading Zend_Config objects

4. Dependencies on Other Framework Components

- Zend_Config_Exception

- Zend_Config
- Zend_Db

5. Theory of Operation

Zend_Config_Db will load a stored Zend_Config object from a database, using the standard Zend_Config workflow. The idea is to store configuration key/value pairs as rows in a database, the main benefit being that configurations of this type can be more easily accessed by multiple servers...such as in a cluster environment, where a change to an Ini or XML configuration file would need to be replicated across all machines using it. A matching Zend_Config_Writer_Db component will exist for updating configurations.

The component should be as flexible as possible, being able to load simple key/value pairs from the database, as well as being able to support environments such as 'production', 'staging', etc. The component will more or less do a SELECT * on the table, and format the rows into a Zend_Config object. Arrays in the Zend_Config object can be represented simply by using Ini 'dot seperator' conventions in key names, such as 'site.emails.admin' for the key column. The component then splits on the separator, breaking it down into the appropriate array.

6. Milestones / Tasks

- Milestone 1: design notes will be published here
- Milestone 2: Working prototype checked into the incubator supporting use cases #1, #2, ...
- Milestone 3: Working prototype checked into the incubator supporting use cases #3 and #4.
- Milestone 4: Unit tests exist, work, and are checked into SVN.
- Milestone 5: Initial documentation exists.

7. Class Index

- Zend_Config_Db

8. Use Cases

Instantiating with a Zend_Db_Table object directly

$config = new Zend_Config_Db($dbTable);

Instantiating passing a table name and adapter as configuration options

$config = new Zend_Config_Db(array('adapter' => $dbAdapter, 'table' => 'my_config_table_name'));

9. Class Skeletons

```
class Zend_Config_Db {

}
```

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