Zend_Config_Writer_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_Writer_Db Component Proposal

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

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_Writer_Db is a component for saving Zend_Config objects to a database table, rather than a flat file.

2. References

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **will** extend Zend_Config_Writer
- This component **will** save a Zend_Config object to a database table in a format that can be read by Zend_Config_Db

4. Dependencies on Other Framework Components

- Zend_Config_Exception

- Zend_Config_Writer
- Zend_Db_Table

5. Theory of Operation

Zend_Config_Writer_Db is the complement to Zend_Config_Db. It will write Zend_Config objects to a database as key/value pairs, where each table row is a key/value pair. Storing configurations in this way can have many benefits, perhaps the largest being that the Zend_Config object is more easily accessible from multiple places.

For example, a service that uses 3 application servers and 1 database server could store the configuration in the database and the 3 app servers read the configuration from the db. This has the immediate benefit of simpler updates, as the configuration only needs to be updated in one place, rather than in the saved Ini or Xml files on each app server.

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_Writer_Db

8. Use Cases

9. Class Skeletons

```
class Zend_Config_Writer_Db {

}
```

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