

# Zend\_Acl Improvement & Zend\_Acl\_Adapter - Niek Slatius

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



## Under Construction

This proposal is under construction and is not ready for review.

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

## Zend Framework: Zend\_Acl & Zend\_Acl\_Adapter Component Proposal

<b>Proposed Component Name</b>	Zend_Acl & Zend_Acl_Adapter
<b>Developer Notes</b>	<a href="http://framework.zend.com/wiki/display/ZFDEV/Zend_Acl+&amp;+Zend_Acl_Adapter">http://framework.zend.com/wiki/display/ZFDEV/Zend_Acl &amp; Zend_Acl_Adapter</a>
<b>Proposers</b>	[Niek Slatius]
<b>Zend Liaison</b>	TBD
<b>Revision</b>	0.1 - 17 December 2008: Initial Draft. (wiki revision: 15)

## 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\_Acl is a component that manages ACL rules. Zend\_Acl currently has no (flexible) support for retrieving ACL rules from persistent sources and no support for storing ACL rules to persistent sources. This improvement will provide support for aforementioned functionality by adding support for adapters to the Zend\_Acl component.

## 2. References

- [Access control list Wikipedia Entry](#)

### 3. Component Requirements, Constraints, and Acceptance Criteria

- Zend\_Acl **will** add have support for accepting an adapter from which this component can read ACL rules
- Zend\_Acl **will** add have support for accepting an adapter to which this component can write ACL rules
- Zend\_Acl **will** add have support for accepting a default adapter for each new instance of Zend\_Acl
- Zend\_Acl **will** add have support for accepting a default readonly value for each new instance of Zend\_Acl

### 4. Dependencies on Other Framework Components

- Zend\_Acl\_Exception

### 5. Theory of Operation

The component will provide an interface to which an instance of a Zend\_Acl\_Adapter\_Interface can be provided.

This Zend\_Acl\_Adapter\_Interface will provide support for reading:

- roles
- resources
- privileges

This Zend\_Acl\_Adapter\_Interface will provide support for writing:

- save (roles, resources and privileges in one go, because of possible integrity constraint handling )

The Zend\_Acl\_Adapter\_Interface will also provide support for being set as readonly.

### 6. Milestones / Tasks

- Milestone 1: Requirement specifications
- Milestone 2: Receive comments from the community about draft / suggestions from community for further improvements
- Milestone 3: Create working prototype

### 7. Class Index

- Zend\_Acl
- Zend\_Acl\_Exception
- Zend\_Acl\_Adapter\_Exception
- Zend\_Acl\_Adapter\_Interface
- Zend\_Acl\_Adapter\_Abstract

### 8. Use Cases

config.ini:

```
database.adapter = pdo_mysql
database.params.host = localhost
database.params.dbname = zendtest
database.params.username = ****
database.params.password = ****

acl.resourcetable.name = aclresource
acl.resourcetable.resourceColumn = id
acl.resourcetable.resourceParentColumn = aclresourceId
acl.privilegetable.name = aclprivilege
acl.privilegetable.roleColumn = aclroleId
acl.privilegetable.resourceColumn = aclresourceId
acl.privilegetable.nameColumn = name
acl.privilegetable.allowColumn = allow
acl.roletable.name = aclrole
acl.roletable.primary = id
acl.role2roletable.name = aclrole_aclrole
acl.role2roletable.roleColumn = aclroleId
acl.role2roletable.roleParentColumn = aclroleParentId
acl.readonly = 1
```

## 9. Class Skeletons

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