

# Zend\_Log Dynamic - Art Hundiak

<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\_Log Component Proposal

<b>Proposed Component Name</b>	Zend_Log
<b>Developer Notes</b>	<a href="http://framework.zend.com/wiki/display/ZFDEV/Zend_Log">http://framework.zend.com/wiki/display/ZFDEV/Zend_Log</a>
<b>Proposers</b>	Art Hundiak
<b>Revision</b>	1.1 - 27 June 2006: Initial Proposal. (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

It is proposed that the existing static Zend\_Log class be changed into a fully dynamic logging class. This proposal is being submitted based on some of the traffic on the mailing list as well as some off line conversations.

### 2. References

- [PEAR Log](#)

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

For many applications the need for a logging class is self evident. The existing static Zend\_Log class covers this need rather nicely but suffers from the usual static class limitations i.e. it is difficult to add functionality and can be difficult to use in a testing environment.

### 4. Dependencies on Other Framework Components

Pretty much the same as the existing Zend\_Log class.

## 5. Theory of Operation

```
$log = new Zend_Log();  
$log->registerLogger(new Zend_Log_Adapeter_File('log_file.txt');  
$log->log('A Message');
```

In most cases the log class would be accessed via the registry. I would suggest having static getInstance(),setInstance(),hasInstance() methods to allow low level code access to one instance of the logging class without the need to go through the registry.

## 6. Milestones / Tasks

zone: Missing {zone-data:milestones}

## 7. Class Index

- Zend\_Log
- Zend\_Log\_Interface
- Zend\_Log\_Exception
- Zend\_Log\_Adapter (existing adapter classes)

## 8. Use Cases

## 9. Class Skeletons

I'm not going to propose skeleton details until I see how the proposal is received. I think the existing class covers most of the standard required functionality so it's basically a question of refactoring the code to remove the statics and maybe adding a params array to the constructor. If someone else has strong opinions on what the class needs then I'm more than willing to either work with them or to turn the entire proposal over to them.

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

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