Exception naming for 2.0 - Jurriën Stutterheim

<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: Exception naming for 2.0 Component Proposal

Proposed Component Name	Exception naming for 2.0
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Exception naming for 2.0
Proposers	Jurriën Stutterheim
Zend Liaison	TBD
Revision	1.0 - 03 June 2008: Initial Draft. (wiki revision: 5)

Table of Contents

1. Overview

This proposal describes an idea for naming conventions for exceptions. There is also a proposal about component naming, but it seemed that this would not fit in well with that proposal.

Currently there is an exception class per component and possibly per part of that component. E.g. Zend_Component_Exception and Zend_Component_Part_Exception. This approach poses a few problems.
The first is that it is currently not possible to catch specific exception in a catch block. This problem is partially tackled by the Exception codes in exceptions proposal. However, this still requires a switch block to sort out which exception was thrown. Please note that this proposal is not a replacement for the exception code proposal. The exception code proposal can be very beneficial for this proposal as well.
The second problem is that exception classes are scattered between the rest of the classes, where they don't really belong.

Instead a propose to follow a more conventional ways of naming exceptions. The name of the exception class would already provide some info of what the problem might be. Also, separate catch blocks could be made to separate different kinds of logic per exception. The way this proposal suggests naming these classes would also arrange for exception classes would be grouped together in a directory.

To maintain some BC and to make it possible to catch an unspecified exception for an entire component, a Zend_Component_Exception class could remain.

In case of Zend_Config, some possible exception classes could be the following:

```
Zend_Config_Exception_ZendConfigException // The main exception file for this
component
Zend_Config_Exception_ReadOnlyException // Thrown when trying to set a value to a
readonly config. Extends Zend_Config_Exception.
Zend_Config_Exception_FileNameNotSetException // Thrown when no filename is set.
Extends Zend_Config_Exception.
```

As you can see, all exceptions are collected in one folder. Now if I want to create a config and I want to be able to handle exceptions, I can write something like this:

```
try {
    $config = new Zend_Config_Xml($configFile);
} catch (Zend_Config_Exception_FileNameNotSetException $e) (
    // Do something
} catch (Zend_Config_Exception_ReadOnlyException $e) {
    // Do something
}
```

But if I wanted to just catch any exception thrown by Zend_Config, I could just do it the old-fashioned way:

```
try {
    $config = new Zend_Config_Xml($configFile);
} catch (Zend_Config_Exception_ZendConfigException $e) (
    // Do something
}
```

Usage with PHP 5.3 namespaces
The exception names proposed above are for the current way of using pseudo-namespaces in ZF. When ZF switches to use PHP 5.3's namespaces, there should be an exception namespace inside the component namespace.
The exception class could look something like this:

```
<?php
namespace Zend::Config::Exception;

class ReadOnlyException extends Zend::Exception {}
```

This would translate to Zend::Config::Exception::ReadOnlyException, or when using the "use" statement, just ReadOnlyException.

```
throw new Zend::Config::Exception::ReadOnlyException('File is readonly');
```

```
use Zend::Config::Exception;

throw new ReadOnlyException('File is readonly');
```

2. References

- Naming conventions for 2.0 - Matthew Ratzloff
- Exception codes in exceptions - Matthew Ratzloff & Thomas Weidner

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

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

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