

# Zend\_Filter\_Localize - Thomas Weidner

<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\_Filter\_Localize / Zend\_Filter\_Normalize Component Proposal

<b>Proposed Component Name</b>	Zend_Filter_Localize / Zend_Filter_Normalize
<b>Developer Notes</b>	<a href="http://framework.zend.com/wiki/display/ZFDEV/Zend_Filter_Localize+Zend_Filter_Normalize">http://framework.zend.com/wiki/display/ZFDEV/Zend_Filter_Localize / Zend_Filter_Normalize</a>
<b>Proposers</b>	Thomas Weidner
<b>Zend Liaison</b>	Matthew Weier O'Phinney
<b>Revision</b>	1.0 - 24 January 2009: Initial Draft. (wiki revision: 7)

## 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\_Filter\_Localize and it's cousin Zend\_Filter\_Normalize are filters for localizing / normalizing input.

### 2. References

- [Zend\\_Locale](#)

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

- Zend\_Filter\_Localize **will** localize given normalized input to a user locale
- Zend\_Filter\_Normalize **will** normalize given localized user input to a normalized notation
- Both components **will ONLY** handle numbers, integer, float and dates
- This components **will not** automatically translate any content as f.e. cities, daynames and so on

## 4. Dependencies on Other Framework Components

- Zend\_Exception
- Zend\_Locale

## 5. Theory of Operation

Zend\_Filter\_Localize and Zend\_Filter\_Normalize are nearly related but will act reverse from each other. Zend\_Filter\_Normalize acts as normalization filter and converts any given localized input from a user into a normalized representation. So a user can for example give '1.234,50' and the filter will return the normalized representation '1234.50' with which the application can do further processing, like storing into a database. This allows multilingual applications to work internally with a fixed representation and still allows users to provide localized inputs.

Zend\_Filter\_Localize works the same way as Zend\_Filter\_Normalize but reversed. It converts a normalized value into a localized representation. So, for example, a normalized '1234.50' input from a database can be displayed to the user in a localized representation and he would get '1.234,50'.

Both components make use of Zend\_Locale\_Format and are able to work with:

- Numbers
- Integers
- Real Point Values (Floats without scientific notation)
- Dates
- Times
- Currencies

Both components WILL NOT do any translations like daynames, monthnames, cities and so on.

## 6. Milestones / Tasks

- Milestone 1: [DONE] Proposal finished
- Milestone 2: [DONE] Proposal accepted
- Milestone 3: [DONE] Working implementation
- Milestone 4: [DONE] Unit tests
- Milestone 5: [DONE] Documentation
- Milestone 6: [DONE] Moved to core

## 7. Class Index

- Zend\_Filter\_Localize
- Zend\_Filter\_Normalize

## 8. Use Cases

UC-01

Normalize given input to english notation  
Autodetect the given valuetype  
Uses the application wide locale as locale to use (de in this case)

```
$filter = new Zend_Filter_Normalize();  
$output = $filter->filter('1.234,50');  
// returns 1,234.50
```

#### UC-02

Normalize the given input to english notation  
Input is fixed to date, which does not autodetect  
Uses the locale 'de' as defined

```
$filter = new Zend_Filter_Normalize('date', 'de');  
$output = $filter->filter('12.April.2008');  
// returns ISO date notation '2008-04-12T00:00:00 +01:00' as no format is defined
```

of course also array notation is supported

```
$filter = new Zend_Filter_Normalize(array('type' => 'date', 'locale' => 'de'));  
$output = $filter->filter('12.April.2008');  
// returns ISO date notation '2008-04-12T00:00:00 +01:00' as no format is defined
```

#### UC-03

Localizes a normal input to a localized notation  
Autodetect the given valuetype  
Uses the application wide locale as locale to use (de in this case)

```
$filter = new Zend_Filter_Localize();  
$output = $filter->filter('1,234.50');  
// returns 1.234,50
```

## 9. Class Skeletons

```
class Zend_Filter_Localize
{
    public function __constructor($type = null, $locale = null)
    {
    }

    public function filter($value)
    {
    }
}

class Zend_Filter_Normalize
{
    public function __constructor($type = null, $locale = null)
    {}

    public function filter($value)
    {}
}
```

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