

---

# **Totem Documentation**

*Release 1.1.0*

**Baptiste "Talus" Clavié**

July 21, 2014



<b>1 Reference Guide</b>	<b>3</b>
1.1 Totem : Basic Usage . . . . .	3



Changeset calculator and handler for any type of complex data

**Requires** at least *PHP 5.4*. Compatible PHP 5.5.



---

## Reference Guide

---

### 1.1 Totem : Basic Usage

Totem is a PHP 5.4 library helping you to calculate changes brought to you data. Sure, that does help much to know what it does... Let me clarify then.

Let's say you have a simple array, which looks like this

```
['foo' => 'bar',  
 'baz' => ['fubar', 'fubaz']]
```

And then, for some reasons, you need to alter the `foo` key, and instead of having the “bar” value, it should have the “qux” value. So you are stuck with

```
['foo' => 'qux',  
 'baz' => ['fubar', 'fubaz']]
```

... And then, let's say you want to say to the end-user “hey, something has changed !”. How do you know which key has changed, without actually knowing it (think data edition through a form) ? The answer is simple, it's Totem !

#### 1.1.1 First step : Taking a bunch of Snapshots

Before you modify the data, you'll first need to take a Snapshot, which will freeze its data a a given time (all of it). To do that, you just need to instantiate a `Totem\Snapshot` object... Or, to be more precise, it depends on the root type of your data. Currently, here are the built-in types :

- `Totem\Snapshot\ArraySnapshot`, if your root data is an array
- `Totem\Snapshot\ObjectSnapshot`, if your root data is an object.

#### 1.1.2 Second Step : Calculate the diff between your two snapshots

Once you have at least two snapshots (one before your modification, the other one after that modification), you may calculate the diff

```
// let's consider you have two snapshots : one "before" and one "after" the  
// modifications  
$set = $snapshot['before']->diff($snapshot['after']);
```

You have then a `Totem\Set` object, in which, from its constructor, the changeset will be computed. It is in fact a sort of a container, in which will be stored all the modifications that happened since the “before” snapshot until the “after”

snapshot. Each items of this snapshot is a Totem\ChangeInterface, which is basically either a Totem\Set object if the key was an object or an array that was modified between the “before” changeset and the “after” changeset, or a Totem\Change object if your data was completely changed (if it is a whole different array, object, or whatever else – string, integer, boolean, you name it))

### 1.1.3 Third and final step : manipulate your diff

Once you have your Totem\ChangeInterface object, you may have access to the “old” value (what it was in the old snapshot), and the new value (what is is now)

```
// let's consider you stored the result set in a '$set' variable
if ($set->hasChanged('foo')) {
    $change = $set->getChange('foo');
    var_dump($change->getOld(), $change->getNew()); // should dump "bar", "qux"
}
```

There, you have a fully fonctionnal changeset !