TYPO3.Flow - Bug # 43587

Status:	Closed	**Priority:**	Should have
Author:	Adrian Föder	**Category:**	
Created:	2012-12-04	**Assigned To:**	
Updated:	2015-02-10	**Due date:**	
PHP Version:			
Has patch:	No		
Complexity:			
Affected Flow version:	Git master		
Subject:	Arrays in Yaml configuration should be interleaved rather than merge/overruled		

Description

consider the following unit test for the Arrays utility class:

```
<?php
/**
 * @test
 */
public function arrayMergeRecursiveOverruleAugmentsNonAssociativeArrayWithoutOverwritingKeys() {
    $actual = \TYPO3\Flow\Utility\Arrays::arrayMergeRecursiveOverrule(array('foo', 'bar', 'baz'), array('one', 'more', 'time'));
    $this->assertCount(6, $actual);
}
```

Currently this **fails** with $actual having only three items: 'one', 'more', 'time'.

This is troublemaking especially for the TYPO3.Flow.reflection.ignoredTags setting since this is a numeric array.

Having

ignoredTags: ['api', 'package', 'subpackage', 'license' ...

active and adding, for example,
ignoredTags: ['my', 'ignoredAnnotation'] ...

will result into removing "api" and "package" from the list eventually.

Related issues:

related to TYPO3.Flow - Feature # 39878: Utility function for determining if ...	**Closed**	**2012-08-17**

History

#1 - 2012-12-04 16:22 - Bastian Waidelich

Mh, not so sure.. The current behavior is what I would expect:

```
\TYPO3\Flow\Utility\Arrays::arrayMergeRecursiveOverrule(array(0 => 'foo', 1 => 'bar'), array(0 => 'overridden', 2 => 'baz'));
// array(0 => 'overridden', 1 => 'bar', 2 => 'baz'
```

The actual problem is probably, that we use non-associative arrays for the **ignoredTags** setting (another problem of this that you can't **unset** a previously defined tag).
In user code, this could easily worked around by using **array_flip** I guess

#2 - 2012-12-04 16:36 - Adrian Föder

yes, the current behavior is even documented behavior:

> *In case of identical keys, ie. keeping the values of the second.*

Ok, well, maybe the method "arrayMergeRecursiveOverrule" is not the right place for this, since also/even the naming of the method implies that numeric keys are overridden mercilessly.

OK, so the problem rather lies in how Settings are merged together.. maybe a distinction here is adequate.

This is really a PHP problem since "numeric" and "Associative" arrays are technically the same; where in other languages like JavaScript (and the Yaml representation), a numeric array is an array [] and an associative array is an object/hash {}...

For Settings, I really would expect arrays [] being interleaved, while hashes {} of course are merged and overruled on same index.

See my last comment @ https://review.typo3.org/#/c/13847/ additionally, please.

#3 - 2012-12-04 16:37 - Adrian Föder

- Subject changed from arrayMergeRecursiveOverrule should interleave nonassociative items rather than merge them to Arrays in Yaml configuration should be interleaved rather than merge/overruled

#4 - 2012-12-13 12:35 - Karsten Dambekalns

- Affected Flow version changed from Git 1.2 (master) to Git master

#5 - 2013-04-24 17:03 - Alexander Berl

The general problem stays - we cannot currently append tags to the ignoreTags configuration, which majorly limits configuration for otherwise broken 3rd party libraries (tcpdf, gedmo, ..).

The real problem, as Bastian noticed, is a syntax problem with the ignoreTags setting. Switching from array to dictionary would work, but make the declaration of tags ugly and cumbersome:

```
TYPO3:
  Flow:
    reflection:
      ignoredTags: {'api':'', 'package':'', 'subpackage':'', 'license':'', 'copyright':'', 'author':'', 'const':'', 'see':'', 'todo':'', 'scope':'', 'fixme':'', 'test':'', 'expectedException':'', 'depends':'', 'dataProvider':'', 'group':'', 'codeCoverageIgnore':''}
```

Extending YAML doesn't make sense, as it is a defined standard, so we're left with:
1) living with the ugly syntax
2) change the way we deal with numerically indexed arrays in configuration loading, meaning changing arrayMergeRecursiveOverrule or replacing it there

I'd go for number two and add a new arrayMergeRecursive function that only overwrites associative arrays.
Determining associativity with SchemaValidator->isNumericallyIndexedArray will not work, because it would also regard YAML dictionaries like

```
{0: '0', 1: '1'}
```

A function to determine associative arrays could look like this:

```
    /**
     * Determine whether the given php array is an associative array
     *
     * @param array $phpArray
     * @return boolean
     */
    protected function isAssociativeArray(array $phpArray) {
        $last = -1;
        foreach (array_keys($phpArray) as $key) {
            if (!is_integer($key) || $key != $last + 1) {
                return TRUE;
            }
            $last = $key;
        }

        return FALSE;
    }
```

So, only strictly monotonic integer indexed arrays with keys starting from 0 are regarded non-associative. The only loophole would be arrays that are declared like this:

```
$array = array( 0 => '0', 1 => '1' );
```

which would be regarded non-associative, even though they could be regarded associative as the key-value pairs are explicitly mapped. This does not apply to YAML though, since only [] arrays are parsed as integer indexed arrays.

This would mean, all array type configuration settings will not overwrite, while all dictionary type settings will overwrite.

#6 - 2013-04-25 00:05 - Alexander Berl

Hum, didn't know that PHP converts all numeric strings in array keys to their integer equivalent... i.e. there's no difference between array('a','b') and array('0' => 'a', '1' => 'b'). Bad bad PHP...
Well, that kind of messes everything up, because then also YAML dictionaries that are indexed 0...n will be regarded non-associative. Still better than not being able to append array options, but that behaviour must then be documented correctly.

#7 - 2013-05-14 12:54 - Gerrit Code Review

- Status changed from New to Under Review

Patch set 1 for branch **master** has been pushed to the review server.
It is available at https://review.typo3.org/20773

#8 - 2013-05-21 13:10 - Alexander Berl

Issue can be closed, the solution will be to change the configuration syntax to use dictionaries for configurations where merging without overwriting is wanted/needed.

#9 - 2015-02-10 16:43 - Bastian Waidelich

- Status changed from Under Review to Closed

see https://review.typo3.org/20773