

TYPO3.TYPO3CR - Bug # 3835

Status:	Resolved	Priority:	Must have
Author:	Robert Lemke	Category:	FLOW3 Persistence
Created:	2009-07-06	Assigned To:	Karsten Dambekalns
Updated:	2011-05-05	Due date:	

Subject: Nested array containing objects as leafs are not handled correctly

Description

Consider this class:

```
/**
 * @scope prototype
 * @entity
 */
class FooNode {

    /**
     * @var array
     */
    protected $objects = array();

    public function bar($foo, $bar, $object) {
        $this->objects[$foo][$bar] = $object;
    }
}
```

bar() adds the given object to \$this->objects – but indexed by \$foo and \$bar.

What you'll get during persistence time is:

```
#1246892972: Object expected, array given. (More information)
```

```
InvalidArgumentException thrown in file
```

```
/Users/Shared/Sites/dev/flow3/dist/Packages/Global/TYPO3CR/Classes/FLOW3/Persistence/IdentityMap.php in line 102.
```

```
12 F3\TYPO3CR\FLOW3\Persistence\IdentityMap::getUIDByObject(array)
```

```
/Users/Shared/Sites/dev/flow3/dist/Packages/Global/TYPO3CR/Classes/FLOW3/Persistence/Backend.php:
```

```
00347: protected function persistObject($object) {
```

```
00348: $queue = array();
```

```
00349: $node = $this->session->getNodeByIdentifier($this->identityMap->getUIDByObject($object));
```

```
00350:
```

```
00351: $classSchema = $this->classSchemata[$object->FLOW3_AOP_Proxy_getProxyTargetClassName()];
```

This should be handled correctly - or if that is not possible, at least a meaningful error should be shown.

Associated revisions

Revision a24e3d98 - 2009-07-06 17:37 - Robert Lemke

- [~BUGFIX] TYPO3CR (FLOW3 Persistence): Added a safeguard for getUUIDByObject() which checks the parameter for validity. Relates to #3835

Revision 298cca39 - 2009-07-07 12:16 - Karsten Dambekalns

[~BUGFIX] TYPO3CR (FLOW3 Persistence): (Nested) arrays containing objects are now stored correctly, even if the leafs are objects. Fixes #3835.

History

#1 - 2009-07-06 17:49 - Robert Lemke

Some idea: The keys could be stored serialized because we only (should) allow objects as leafs anyway:

In userland we have:

```
array('foo' => array('bar' => array('baz' => $object)))
```

and in the CR we would serialize `array('foo' => array('bar' => 'baz'))` which would be the key while `$object` is the value.

Alternatively you could use some special character like `chr(0)` if that is allowed in your storage backend:

```
$key = 'foo' . chr(0) . 'bar' . chr(0) . 'baz';
```

#2 - 2009-07-07 09:56 - Karsten Dambekalns

- *Status changed from New to Accepted*

#3 - 2009-07-07 13:00 - Karsten Dambekalns

- *Status changed from Accepted to Resolved*

- *% Done changed from 0 to 100*

Applied in changeset r2741.

#4 - 2011-05-05 17:14 - Robert Lemke

- *Target version deleted (0.5.0 alpha 3)*