TYPO3.Flow - Bug # 29405

Status:	New		Priority:	Must have
Author:	Sebastian Kurfuerst		Category:	Session
Created:	2011-09-01		Assigned To:	
Updated:	2013-04-04		Due date:	

PHP Version:	
Has patch:	
Complexity:	
Affected Flow version:	FLOW3 1.0.0

Subject:	When storing a new entity inside the session, it will be fully serialized instead of just the reference being stored

Description

When having the following code:

```
1 $newEntity = new MyEntity();
2 $newEntityRepository->add($newEntity);
3
4 $someObjectWithSessionScope->setEntity($newEntity);
```

Then, $newEntity is FULLY serialized; although it should only be stored as UUID reference in the session.

As an example, this happens when running the BlogExample setup; after creating an initial user -- this user is directly logged in, but the Account is fully stored in the session.

This leads to very weird bugs, in my case I wanted to store a reference to $account in another entity; and doctrine then told me that the object was not known yet.

The issue is in DependencyInjection\ProxyClassBuilder::buildSerializeRelatedEntitiesCode around line 239:

```
1 if (\$this->\$propertyName instanceof \\TYPO3\\FLOW3\\Persistence\\Aspect\\PersistenceMagicInterface && !\\TYPO3\\
  FLOW3\\Core\\Bootstrap::\$staticObjectManager->get('TYPO3\\FLOW3\\Persistence\\PersistenceManagerInterface'
  )->isNewObject(\$this->\$propertyName) || \$this->\$propertyName instanceof \\Doctrine\\ORM\\Proxy\\Proxy) {
2          if (!property_exists(\$this, 'FLOW3_Persistence_RelatedEntities') || !is_array(\$this->
  FLOW3_Persistence_RelatedEntities)) {
3
```

Basically this check somehow needs to be adjusted, such that:
1) New objects **which are scheduled for addition** are serialized as UUID
2) New objects **which are NOT scheduled for addition** are serialized completely.

@Karsten: Any hints on how we can check that? It seems that PersistenceManagerInterface.::isNewObject also returns TRUE when the object is not known at all to the unit of work. Maybe we need to extend PersistenceManagerInterface then?

History

#1 - 2011-09-09 15:17 - Robert Lemke

- Target version changed from 1.0 beta 2 to 1.0.0

#2 - 2011-10-20 01:47 - Karsten Dambekalns

- Target version deleted (1.0.0)

#3 - 2011-10-21 13:20 - Karsten Dambekalns

- Affected Flow version set to FLOW3 1.0.0

#4 - 2013-04-04 11:34 - Benno Weinzierl

Is that issue still relevant?

I am asking because we have an other issue with session scope objects getting another session scope object injected.

This works for one request, after that it results in a ugly apache seg-fault (i suspect because the injected session object is also stored and not

reinjected).

I fixed it with @Flow\Transient but it would be nice to make session scope objects work as expected auto of the box.