

TYPO3.Flow - Bug # 45272

Status:	New	Priority:	Should have
Author:	Peter Rauber	Category:	
Created:	2013-02-08	Assigned To:	
Updated:	2013-02-08	Due date:	
PHP Version:			
Has patch:	No		
Complexity:			
Affected Flow version: (any)			
Subject:	Related Value Objects get deleted by default cascading		
Description			
Description by example:			
An entity that relates to a value object:			
<pre>/** * A Building * * @FLOW3\Entity */ class Building { /** * @ORM\ManyToOne * @var \MyNamespace\Domain\Model\BuildingType */ protected \$buildingType; }</pre>			
Two entities point to the same BuildingType:			
<pre>\$building1 = new \MyNamespace\Domain\Model\Building(); \$building1->setBuildingType(new \MyNamespace\Domain\Model\BuildingType('Type 1')); \$this->buildingRepository->add(\$building1); \$building2 = new \MyNamespace\Domain\Model\Building(); \$building2->setBuildingType(new \MyNamespace\Domain\Model\BuildingType('Type 1')); \$this->buildingRepository->add(\$building2);</pre>			
When you try to delete building1, FLOW/doctrine will cascade the removal to the value object and that will cause an exception, because building2 still needs the value object, as it points to it's identifier.			
I solved that by adding an empty cascade-attribute to that relations: @ORM\ManyToOne(cascade={}).			
Suggestion:			
The cascading was added by Flow3AnnotationDriver as this driver adds \$mapping['cascade'] = array('all') to all properties that are not aggregate roots. I suggest that in the case of value objects \$mapping['cascade'] = array('persist') will be added instead of 'all'.			

