

## Core - Bug # 56442

<b>Status:</b>	Resolved	<b>Priority:</b>	Must have
<b>Author:</b>	Xavier Perseguers	<b>Category:</b>	Extbase
<b>Created:</b>	2014-02-28	<b>Assigned To:</b>	Xavier Perseguers
<b>Updated:</b>	2014-03-03	<b>Due date:</b>	
<b>TYPO3 Version:</b>	6.2		
<b>PHP Version:</b>	5.4		
<b>Complexity:</b>	medium		
<b>Is Regression:</b>	No		
<b>Sprint Focus:</b>			
<b>Subject:</b>	Invalid FileReference used		
<b>Description</b>			
<p>In my domain model (watches) I have this:</p> <pre>family (1:n) familyLine (1:n) watch (1:1) image</pre> <p>Thus a family has some familyLines which in turns have watches and each watch has an associated image (FileReference).</p> <p>As "image" is a FileReference, column "image" from table "watch" is a counter of how many references are associated, thus "1" in my case since there is only one allowed image to be linked.</p> <p>familyLines are lazy loaded, watches too and image reference as well.</p> <p>When I loop over my families in the view, I basically do this:</p> <pre>foreach families as family     foreach family.familyLines as familyline         show familyline.&lt;first-reference&gt;.image     endforeach endforeach</pre> <p>Now, the <b>first image</b> to be loaded will put the corresponding FileReference's uid into memory. In my case this happens to be sys_file_reference row with uid=1. When I loop over the second picture to be retrieved (from lazy loading), it has to map value "1" (which actually is not a uid but a counter) as a FileReference and as such it finds the first FileReference that was retrieved before and it does not load the actual FileReference but the previous one.</p> <p>In the end this means that it shows the first image instead of the real one.</p>			
<b>Related issues:</b>			
related to Core - Task # 53514: Extbase: DataMapper causes redundant queries		<b>Resolved</b>	<b>2013-11-11</b>

### Associated revisions

Revision 04f66dba - 2014-03-03 12:21 - Helmut Hummel

[BUGFIX] Always fetch related objects if relation set by child

If the DataMapper shall map a 1:N relation for  
a property of a domain object and the element

type of this property is another domain object and not a storage type, the data mapper must always fetch the related elements from persistence and cannot rely on the persistence session lookup.

This is the case because the relation to the parent is set on the child side and not the other way around. The column in the parent row is useless in that case and does not hold the id of the child.

We can also not rely on the persistence session because we do not know the identifier of the child until we fetched it.

Resolves: #56442

Releases: 6.2, 6.1

Change-Id: lcc3ebf9b825f6380691c60839621f01ca9875e4e

Reviewed-on: <https://review.typo3.org/27949>

Reviewed-by: Xavier Perseguers

Tested-by: Xavier Perseguers

Reviewed-by: Marc Bastian Heinrichs

Tested-by: Marc Bastian Heinrichs

## History

---

### #1 - 2014-02-28 15:01 - Xavier Perseguers

- Status changed from New to On Hold

~~The problem is actually not related to Extbase, but my domain model as I type hinted the domain model property as a FileReference whereas it's actually a ObjectStorage of FileReference's. Must see if a single FileReference is possible with FAL.~~

Extbase is properly seeing the relation as RELATION\_HAS\_ONE

### #2 - 2014-02-28 15:53 - Gerrit Code Review

- Status changed from On Hold to Under Review

Patch set 1 for branch **master** of project **Packages/TYPO3.CMS** has been pushed to the review server.

It is available at <https://review.typo3.org/27949>

### #3 - 2014-02-28 16:07 - Xavier Perseguers

- Assigned To set to Xavier Perseguers

### #4 - 2014-03-01 10:25 - Marc Bastian Heinrichs

I consider this not as a bug.

A relation to filereference is 1:n by design - also, if max in tca is set to one.

The image property should be defined:

```

/**
 * @var \TYPO3\CMS\Extbase\Persistence\ObjectStorage<\TYPO3\CMS\Extbase\Domain\Model\FileReference>
 * @lazy
 */
protected $image;

```

In the fluid template you have to iterate over the image to render the single one or you could use the iteratorFirstVH of vhs.

```

{namespace v=Tx_Vhs_ViewHelpers}
<v:var.set name="firstImage" value="{v:iterator.first(haystack: images)}" />
<f:image src="{firstImage.uid}" class="{cssClass}" alt="" maxWidth="{settings.imageMaxWidth}" height="{imageHeight}" width="{imageWidth}"
treatIdAsReference="TRUE"/>

```

#### #5 - 2014-03-01 22:19 - Helmut Hummel

Marc Bastian Heinrichs wrote:

```

| I consider this not as a bug.
| A relation to filereference is 1:n by design - also, if max in tca is set to one.

```

I tend to agree. It would be consistent.

However, the whole Extbase persistent can deal with the situation that the relation is one to one, but the relation is defined on the child (parentKeyFieldName is set).

Besides that, what makes it now break is the caching introduced in #53514

So my suggestion would be to let the method mapObjectToClassProperty() deal with that and act accordingly.

#### #6 - 2014-03-01 22:34 - Gerrit Code Review

Patch set 2 for branch **master** of project **Packages/TYPO3.CMS** has been pushed to the review server.

It is available at <https://review.typo3.org/27949>

#### #7 - 2014-03-03 12:25 - Marc Bastian Heinrichs

Sorry Xavier, misinterpreted the whole thing.  
Of course the example with a 1:1 relation with irre worked before introducing the cache in #53514.

Thanks for finding this bug! :-)

#### #8 - 2014-03-03 12:30 - Helmut Hummel

- Status changed from Under Review to Resolved

- % Done changed from 0 to 100

Applied in changeset commit:04f66dba2538cfc3c7de8578e4f2bd35900702af.