

## TYPO3.Flow - Bug #59357

### Using the PackageManager directly instead of the Interface results in unexpected behavior

2014-06-05 11:25 - Marc Neuhaus

<b>Status:</b> New	<b>Start date:</b> 2014-06-05
<b>Priority:</b> Should have	<b>Due date:</b>
<b>Assignee:</b>	<b>% Done:</b> 0%
<b>Category:</b>	<b>Estimated time:</b> 0.00 hour
<b>Target version:</b>	<b>Complexity:</b>
<b>PHP Version:</b>	
<b>Has patch:</b> No	
<b>Description</b> If you try to inject the PackageManager into your class using the Inject Annotation the "getAvailablePackages" returns no Packages because the PackageManager doesn't seem to be initialized. Using the alternative injection method through an injectPackageManager method works without a Problem.	
<b>Related issues:</b> Has duplicate TYPO3 Flow Base Distribution - Bug #59358: PackageManager does ... <b>Closed</b> <b>2014-06-05</b>	

#### History

##### #1 - 2014-06-05 14:03 - Bastian Waidelich

- Status changed from New to Needs Feedback
- Assignee set to Bastian Waidelich

Marc, I can't reproduce this. Can you please add an example?

For me it doesn't make a difference which injection method is used (that would be a weird bug) but whether it's referring to the PackageManager or its interface (which is registered as "earlyInstance" in the bootstrap):

```
use TYPO3\FLOW\Annotations as Flow;
use TYPO3\FLOW\Cli\CommandController;
use TYPO3\FLOW\Package\PackageManager;
use TYPO3\FLOW\Package\PackageManagerInterface;

/**
 * @Flow\Scope("singleton")
 */
class TestCommandController extends CommandController {

    /**
     * @var PackageManager
     */
    protected $packageManager1;

    /**
     * @Flow\Inject
     * @var PackageManager
     */
    protected $packageManager2;

    /**
     * @var PackageManagerInterface
     */
    protected $packageManager3;

    /**
     * @Flow\Inject
     * @var PackageManagerInterface
     */
    protected $packageManager4;

    /**
     * @param PackageManager $packageManager1
     * @return void
     */
}
```

```

*/
public function injectPackageManager1(PackageManager $packageManager1) {
    $this->packageManager1 = $packageManager1;
}

/**
 * @param PackageManagerInterface $packageManager3
 * @return void
 */
public function injectPackageManager(PackageManagerInterface $packageManager3) {
    $this->packageManager3 = $packageManager3;
}

/**
 * @return void
 */
public function testCommand() {
    $this->outputLine('1: %d packages', array(count($this->packageManager1->getAvailablePackages())));
    $this->outputLine('2: %d packages', array(count($this->packageManager2->getAvailablePackages())));
    $this->outputLine('3: %d packages', array(count($this->packageManager3->getAvailablePackages())));
    $this->outputLine('4: %d packages', array(count($this->packageManager4->getAvailablePackages())));
}
}

```

Output:

```

1: 0 packages
2: 0 packages
3: 60 packages
4: 60 packages

```

## #2 - 2014-06-05 14:10 - Christian Look

Bastian Waidelich wrote:

Marc, I can't reproduce this. Can you please add an example?

For me it doesn't make a difference which injection method is used (that would be a weird bug) but whether it's referring to the PackageManager or its interface (which is registered as "earlyInstance" in the bootstrap):

[...]

Output:

[...]

For my case (which i stated in the other Ticket) I used the PackageManager directly.

I do now know what to do, but I think it is very confusing. There should be some sort of notification or error when you try to use it directly instead of the interface. It is just not the behaviour you would expect at all. Or it should be fixed so that you can use both. (I dint see the point of an interface face here anyways, but that might just be me)

Cheers,

Christian

## #3 - 2014-06-05 14:20 - Marc Neuhaus

lol, just realized, that i tried to inject the PackageManager itself through Annotation, which didn't work and then added an injectMethod with the PackageManagerInterface in the method which then works of course ;D

still, i would either expect a helpful exception that you should use the Interface instead, or it should work that way as well, imho

## #4 - 2014-06-05 14:31 - Bastian Waidelich

- Status changed from Needs Feedback to New

- Assignee deleted (Bastian Waidelich)

I do not know what to do, but I think it is very confusing [...]

still, I would either expect a helpful exception that you should use the Interface instead, or it should work that way as well, imho [...]

I agree, it's just a different issue. Would you update the subject accordingly?

**#5 - 2014-06-05 14:34 - Marc Neuhaus**

*- Subject changed from PackageManager isnt initialized when injected through annotation to Using the PackageManager directly instead of the Interface results in unexpected behavior*