TYPO3.Flow - Bug # 46716

Status:	New	**Priority:**	Must have
Author:	Adrian Föder	**Category:**	Object
Created:	2013-03-27	**Assigned To:**	
Updated:	2013-05-21	**Due date:**	
PHP Version:			
Has patch:	No		
Complexity:			
Affected Flow version:	Git master		
Subject:	Empty class names in DependencyInjection proxy code when using Caches / Factory-created dependencies		

Description

I noticed a few cases where the $className argument of \TYPO3\Flow\Object\ObjectManager::createLazyDependency was an empty string; that is the case, for example, in

\TYPO3\Flow\Resource\ResourceManager::Flow_Proxy_injectProperties:

```
/**
 * Autogenerated Proxy Method
 */
 private function Flow_Proxy_injectProperties() {
    $statusCache_reference = &$this->statusCache;
    $this->statusCache = \TYPO3\Flow\Core\Bootstrap::$staticObjectManager->getLazyDependencyByHash('b3ca84fd627a5045e163e999a38877bf', $statusCache_reference);
    if ($this->statusCache === NULL) {
        $this->statusCache = \TYPO3\Flow\Core\Bootstrap::$staticObjectManager->createLazyDependency('b3ca84fd627a5045e163e999a38877bf',  $statusCache_reference, '', function() { return \TYPO3\Flow\Core\Bootstrap::$staticObjectManager->get('TYPO3\Flow\Cache\CacheManager')->getCache('Flow_Resource_Status'); });
    }
     // etc...
```

The complete generated class file can be seen here:
https://gist.github.com/afoeder/18b878d29c418b313773#file-typo3_flow_resource_resourcemanager-php-L480

Possible reason

So, looks like the reason is at \TYPO3\Flow\Object\DependencyInjection\ProxyClassBuilder::buildPropertyInjectionCodeByConfiguration. There is a check for

```
if ($propertyClassName === NULL) {
```

but at the end of the method, the $propertyClassName is still used without having been modified, hence still NULL:

```
protected function buildPropertyInjectionCodeByConfiguration(Configuration $objectConfiguration, $propertyName, Configuration $propertyConfiguration) {
    $className = $objectConfiguration->getClassName();
    $propertyClassName = $propertyConfiguration->getClassName();
    if ($propertyClassName === NULL) {
        $preparedSetterArgument = $this->buildCustomFactoryCall($propertyConfiguration->getFactoryObjectName(), $propertyConfiguration->getFactoryMethodName(), $propertyConfiguration->getArguments());
    } else {
        // ...
    }
    // ...
    return $this->buildLazyPropertyInjectionCode($propertyClassName, $propertyName, $preparedSetterArgument);
}
```

See, obviously the last line **can** be invoked if $propertyClassName is NULL; which is quite often the case when using configured factories.

History

#1 - 2013-03-27 12:25 - Adrian Föder

- Subject changed from Infinite recursive call in DependencyProxy to Empty class names in cache-related proxy classes

- Status changed from Accepted to New

- Priority changed from Must have to Should have

#2 - 2013-03-27 15:41 - Adrian Föder

- Subject changed from Empty class names in cache-related proxy classes to Empty class names in DependencyInjection proxy code when using Caches / Factory-created dependencies

#3 - 2013-03-27 15:42 - Adrian Föder

- Priority changed from Should have to Must have

#4 - 2013-05-21 13:22 - Robert Lemke

- Assigned To deleted (Robert Lemke)

- Target version set to 2.0.1