

## TYPO3 Core - Bug #78868

### Check for broken extensions should check ext\_emconf.php

2016-12-03 09:31 - Franz Holzinger

<b>Status:</b>	Closed	<b>Start date:</b>	2016-12-03
<b>Priority:</b>	Could have	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Install Tool	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Complexity:</b>	
<b>TYPO3 Version:</b>	7	<b>Is Regression:</b>	No
<b>PHP Version:</b>	5.6	<b>Sprint Focus:</b>	
<b>Tags:</b>			

#### Description

The Extension Manager throws exceptions after the upgrade from TYPO3 6.2.27 to 7.6.14. The Install Tool should have reported those problems before.

Those problematic extensions have never been installed. They are old and inside the typo3conf/ext folder. However they crash the EM:

Uncaught TYPO3 Exception

#1439552059: The extension "dd\_extensionframework" has invalid version constraints in conflicts section. Extension key is missing! (More information)

```
TYPO3\CMS\Core\Package\Exception\InvalidPackageManifestException thrown in file
/var/www/html/typo3_src-7.6.14/typo3/sysex/core/Classes/Package/PackageManager.php in line 1022.
```

```
22 TYPO3\CMS\Core\Package\PackageManager::mapExtensionManagerConfigurationToComposerManifest("dd_
extensionframework", array, stdClass)
```

```
/var/www/html/typo3_src-7.6.14/typo3/sysex/core/Classes/Package/PackageManager.php:
```

```
00955:         $extensionManagerConfiguration,
```

```
00956:         $composerManifest ?: new stdClass()
```

```
00957:     );
```

```
00958:
```

```
00959:     return $composerManifest;
```

```
21 TYPO3\CMS\Core\Package\PackageManager::getComposerManifest("/var/www/html/typo3conf/ext/dd_exte
nsionframework/")
```

```
/var/www/html/typo3_src-7.6.14/typo3/sysex/core/Classes/Package/PackageManager.php:
```

```
00347:     }
```

```
00348:     try {
```

```
00349:         $composerManifest = $this->getComposerManifest($packagePath);
```

```
00350:         $packageKey = $this->getPackageKeyFromManifest($composerManifest, $packageP
ath, $packagesBasePath);
```

```
00351:         $this->composerNameToPackageKeyMap[strtolower($composerManifest->name)] = $
packageKey;
```

```
20 TYPO3\CMS\Core\Package\PackageManager::scanAvailablePackages("PackageManagement::packagesMayHav
eChanged")
```

```
19 call_user_func_array(array, array)
```

```
/var/www/html/typo3_src-7.6.14/typo3/sysex/extbase/Classes/SignalSlot/Dispatcher.php:
```

```

00138:     }
00139:
00140:         $slotReturn = call_user_func_array([$object, $slotInformation['method']], $preparedSlotArguments);
00141:
00142:         if ($slotReturn) {
18 TYPO3\CMS\Extbase\SignalSlot\Dispatcher::dispatch("PackageManagement", "packagesMayHaveChanged"
)
/var/www/html/typo3_src-7.6.14/typo3/sysext/extensionmanager/Classes/Utility/ListUtility.php:
00150:     protected function emitPackagesMayHaveChangedSignal()
00151:     {
00152:         $this->signalSlotDispatcher->dispatch('PackageManagement', 'packagesMayHaveChanged'
);
00153:     }
00154:

```

That extension uses invalid constraints:  
(see attached file)

```

'constraints' => array(
    'depends' => array(
        'typo3' => '',
        'php' => '',
        'cms' => '',
    ),
    'conflicts' => array(
        '' => '',
    ),
),
'suggests' => array(
),

```

#### Related issues:

Is duplicate of TYPO3 Core - Feature #73015: Add ext\_emconf.php validator

New

2016-01-29

#### History

#1 - 2016-12-03 09:35 - Franz Holzinger

- File ext\_emconf.php added

sk\_fancybox:

```

array (
    'depends' =>
        array (
            'typo3' => '4.4.0-6.0.99',
            '' => '',
        ),
    'conflicts' =>
        array (
        ),
    'suggests' =>
        array (
        ),
),

```

Uncaught TYPO3 Exception

#1439552058: The extension "sk\_fancybox" has invalid version constraints in depends section. Extension key is missing! (More information)

TYPO3\CMS\Core\Package\Exception\InvalidPackageManifestException thrown in file

/var/www/html/typo3\_src-7.6.14/typo3/sysext/core/Classes/Package/PackageManager.php in line 1012.

```

22 TYPO3\CMS\Core\Package\PackageManager::mapExtensionManagerConfigurationToComposerManifest("sk_fancybox", array, stdClass)

/var/www/html/typo3_src-7.6.14/typo3/sysext/core/Classes/Package/PackageManager.php:
00955:         $extensionManagerConfiguration,
00956:         $composerManifest ?: new stdClass()

00957:     );

00958:
00959:     return $composerManifest;

21 TYPO3\CMS\Core\Package\PackageManager::getComposerManifest("/var/www/html/typo3conf/ext/sk_fancybox/")

/var/www/html/typo3_src-7.6.14/typo3/sysext/core/Classes/Package/PackageManager.php:
00347:     }
00348:     try {

00349:         $composerManifest = $this->getComposerManifest($packagePath);

00350:         $packageKey = $this->getPackageKeyFromManifest($composerManifest, $packagePath, $packagesBasePath);
00351:         $this->composerNameToPackageKeyMap[strtolower($composerManifest->name)] = $packageKey;

20 TYPO3\CMS\Core\Package\PackageManager::scanAvailablePackages("PackageManagement::packagesMayHaveChanged")

19 call_user_func_array(array, array)

/var/www/html/typo3_src-7.6.14/typo3/sysext/extbase/Classes/SignalSlot/Dispatcher.php:
00138:     }
00139:

00140:     $slotReturn = call_user_func_array([$object, $slotInformation['method']], $preparedSlotArguments);

00141:
00142:     if ($slotReturn) {

18 TYPO3\CMS\Extbase\SignalSlot\Dispatcher::dispatch("PackageManagement", "packagesMayHaveChanged")

/var/www/html/typo3_src-7.6.14/typo3/sysext/extensionmanager/Classes/Utility/ListUtility.php:
00150:     protected function emitPackagesMayHaveChangedSignal()
00151:     {

00152:         $this->signalSlotDispatcher->dispatch('PackageManagement', 'packagesMayHaveChanged');

00153:     }
00154:

17 TYPO3\CMS\Extbase\Utility>ListUtility::emitPackagesMayHaveChangedSignal()

/var/www/html/typo3_src-7.6.14/typo3/sysext/extensionmanager/Classes/Utility/ListUtility.php:
00110:     if ($this->availableExtensions === null) {
00111:         $this->availableExtensions = [];

00112:         $this->emitPackagesMayHaveChangedSignal();

00113:         foreach ($this->packageManager->getAvailablePackages() as $package) {
00114:             $installationType = $this->getInstallationTypeForPackage($package);

16 TYPO3\CMS\Extbase\Utility>ListUtility::getAvailableExtensions()

/var/www/html/typo3_src-7.6.14/typo3/sysext/extensionmanager/Classes/Utility/ListUtility.php:
00290:     public function getAvailableAndInstalledExtensionsWithAdditionalInformation()
00291:     {

00292:         $availableExtensions = $this->getAvailableExtensions();

```

```

00293:         $availableAndInstalledExtensions = $this->getAvailableAndInstalledExtensions($availableExtensions);
00294:         $availableAndInstalledExtensions = $this->enrichExtensionsWithIconInformation($availableAndInstalledExtensions);

15 TYPO3\CMS\Extensionmanager\Utility>ListUtility::getAvailableAndInstalledExtensionsWithAdditionalInformation()

/var/www/html/typo3_src-7.6.14/typo3/sysex/extensionmanager/Classes/Controller/ListController.php:

00152:     {
00153:         $this->addComposerModeNotification();

00154:         $availableAndInstalledExtensions = $this->listUtility->getAvailableAndInstalledExtensionsWithAdditionalInformation();

00155:         ksort($availableAndInstalledExtensions);
00156:         $this->view->assign('extensions', $availableAndInstalledExtensions);

14 TYPO3\CMS\Extensionmanager\Controller>ListController::indexAction()

13 call_user_func_array(array, array)

/var/www/html/typo3_src-7.6.14/typo3/sysex/extbase/Classes/Mvc/Controller/ActionController.php:

00281:         if (!$validationResult->hasErrors()) {
00282:             $this->emitBeforeCallActionMethodSignal($preparedArguments);

00283:             $actionResult = call_user_func_array([$this, $this->actionMethodName], $preparedArguments);

00284:         } else {
00285:             $methodTagsValues = $this->reflectionService->getMethodTagsValues(get_class($this), $this->actionMethodName);

12 TYPO3\CMS\Extbase\Mvc\Controller\ActionController::callActionMethod()

/var/www/html/typo3_src-7.6.14/typo3/sysex/extbase/Classes/Mvc/Controller/ActionController.php:

00174:         $this->initializeView($this->view);
00175:     }

00176:     $this->callActionMethod();

00177: }
00178:

11 TYPO3\CMS\Extbase\Mvc\Controller\ActionController::processRequest(TYPO3\CMS\Extbase\Mvc\Web\Request, TYPO3\CMS\Extbase\Mvc\Web\Response)

/var/www/html/typo3_src-7.6.14/typo3/sysex/extbase/Classes/Mvc/Dispatcher.php:

00084:         $controller = $this->resolveController($request);
00085:         try {

00086:             $controller->processRequest($request, $response);

00087:         } catch (\TYPO3\CMS\Extbase\Mvc\Exception\StopActionException $ignoredException) {
00088:         }

10 TYPO3\CMS\Extbase\Mvc\Dispatcher::dispatch(TYPO3\CMS\Extbase\Mvc\Web\Request, TYPO3\CMS\Extbase\Mvc\Web\Response)

/var/www/html/typo3_src-7.6.14/typo3/sysex/extbase/Classes/Mvc/Web/BackendRequestHandler.php:

00030:         /** @var $response \TYPO3\CMS\Extbase\Mvc\ResponseInterface */
00031:         $response = $this->objectManager->get(\TYPO3\CMS\Extbase\Mvc\Web\Response::class);

00032:         $this->dispatcher->dispatch($request, $response);

00033:         return $response;
00034:     }

9 TYPO3\CMS\Extbase\Mvc\Web\BackendRequestHandler::handleRequest()

/var/www/html/typo3_src-7.6.14/typo3/sysex/extbase/Classes/Core/Bootstrap.php:

```

```

00204:     $requestHandler = $requestHandlerResolver->resolveRequestHandler();
00205:

00206:     $response = $requestHandler->handleRequest();

00207:     // If response is NULL after handling the request we need to stop
00208:     // This happens for instance, when a USER object was converted to a USER_INT

8 TYPO3\CMS\Extbase\Core\Bootstrap::handleRequest()

/var/www/html/typo3_src-7.6.14/typo3/sysext/extbase/Classes/Core/Bootstrap.php:

00191:     {
00192:         $this->initialize($configuration);

00193:         return $this->handleRequest();

00194:     }
00195:

7 TYPO3\CMS\Extbase\Core\Bootstrap::run("", array)

/var/www/html/typo3_src-7.6.14/typo3/sysext/backend/Classes/Http/BackendModuleRequestHandler.php:

00212:         // Run Extbase
00213:         $bootstrap = GeneralUtility::makeInstance(\TYPO3\CMS\Extbase\Core\Bootstrap::class);

00214:         $content = $bootstrap->run('', $configuration);

00215:
00216:         $response->getBody()->write($content);

6 TYPO3\CMS\Backend\Http\BackendModuleRequestHandler::dispatchModule("tools_ExtensionmanagerExtensionmanager")

/var/www/html/typo3_src-7.6.14/typo3/sysext/backend/Classes/Http/BackendModuleRequestHandler.php:

00090:         $moduleName = (string)$this->request->getQueryParams()['M'];
00091:         if ($this->isDispatchedModule($moduleName)) {

00092:             return $this->dispatchModule($moduleName);

00093:         } else {
00094:             // @deprecated: This else path is deprecated and throws deprecations logs at registration t
ime. Can be removed with TYPO3 CMS 8.

5 TYPO3\CMS\Backend\Http\BackendModuleRequestHandler::handleRequest(TYPO3\CMS\Core\Http\ServerRequest)

/var/www/html/typo3_src-7.6.14/typo3/sysext/core/Classes/Core/Bootstrap.php:

00300:
00301:     // Execute the command which returns a Response object or NULL

00302:     $this->response = $requestHandler->handleRequest($request);

00303:     return $this;
00304: }

4 TYPO3\CMS\Core\Core\Bootstrap::handleRequest(TYPO3\CMS\Core\Http\ServerRequest)

/var/www/html/typo3_src-7.6.14/typo3/sysext/backend/Classes/Http/Application.php:

00092:     }
00093:

00094:     $this->bootstrap->handleRequest($this->request);

00095:
00096:     if ($execute !== null) {

3 TYPO3\CMS\Backend\Http\Application::run()

/var/www/html/typo3_src-7.6.14/typo3/index.php:

00018: call_user_func(function () {

```

```

00019:     $classLoader = require __DIR__ . '/../vendor/autoload.php';
00020:     (new \TYPO3\CMS\Backend\Http\Application($classLoader)->run());
00021: });
2 {closure}()
1 call_user_func(Closure)
/var/www/html/typo3_src-7.6.14/typo3/index.php:
00019:     $classLoader = require __DIR__ . '/../vendor/autoload.php';
00020:     (new \TYPO3\CMS\Backend\Http\Application($classLoader)->run());
00021: });

```

## #2 - 2016-12-07 12:50 - Wouter Wolters

- Status changed from New to Closed

There was an issue within the TER extension which is fixed already. All extensions which were uploaded while that issue was there have that problem. there is nothing you can do except uploading a new version.

The version constraint contains a " => " which causes this problem. See also <https://forge.typo3.org/issues/77259#note-4>

A ext\_emconf.php validator is planned but not on a short notice. See <https://forge.typo3.org/issues/73015>

## #3 - 2017-04-27 22:49 - Thomas Mayer

I just updated an instance from 6.2.31 (non-composer install) to 7.6.18 and ran into this issue:

```
#1439552059: The extension "simulatestatic" has invalid version constraints in conflicts section. Extension key is missing! (More information)
```

```
TYPO3\CMS\Core\Package\Exception\InvalidPackageManifestException thrown in file
/path/to/typo3/sysex/core/Classes/Package/PackageManager.php in line 1022.
```

Note that I never used extension simulatestatic.

Also note that I updated the TER list in 6.2.31 shortly before upgrading.

I get this error every time I call the extension manager.

Everything else still seems to run (backend/frontend), thereby the install tool tells me that all extensions can be loaded.

I worked around it with a

```
cd typo3conf/ext
rm -R simulatestatic
```

Now the extension manager runs again.

## #4 - 2017-04-27 23:09 - Riccardo De Contardi

would it help to clean the table tx\_extensionmanager\_domain\_model\_extension ?

### Files

ext_emconf.php	1.97 KB	2016-12-03	Franz Holzinger
ext_emconf.php	1.07 KB	2016-12-03	Franz Holzinger