Core - Bug # 55517

Task # 49162 (Resolved): Rewrite install tool

Status:	Resolved	**Priority:**	Should have
Author:	Daniel Siepmann	**Category:**	
Created:	2014-01-31	**Assigned To:**	
Updated:	2014-06-11	**Due date:**	
TYPO3 Version:	6.2		
PHP Version:			
Complexity:			
Is Regression:	No		
Sprint Focus:			
Subject:	ClassLoader not working with NullBackend for legacy classes		

Description

Setting *cache_core* to *NullBackend*

```
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['cache_core']['backend'] =
'\TYPO3\CMS\Core\Cache\Backend\NullBackend';
```

creates Fatal Errors for calls to old class names like *t3lib_extMgm*.
You can reproduce it e.g. with a call to the *t3lib_extMgm* class inside *ext_tables.php* of an extension.

This error is not catched using the *Install ToolCheck for broken extensions* option.

Related issues:

related to Core - Epic # 47018: Implement Composer support and clean package ...	**Resolved**	**2013-08-28**

Associated revisions

Revision ebe55ee9 - 2014-01-31 22:03 - Daniel Siepmann

[BUGFIX] ClassLoader not working with NullBackend for legacy classes

Removed unused line that creates a fatal error.
Implemented eval in case NullBackend is set for “cache_core”
and no alias mapping is included.

Resolves: #55517
Releases: 6.2
Change-Id: Ie1d58985496c4561d8b20fedbd030ad98760becd
Reviewed-on: https://review.typo3.org/27218
Reviewed-by: Helmut Hummel
Tested-by: Helmut Hummel
Reviewed-by: Wouter Wolters
Reviewed-by: Christian Kuhn
Tested-by: Christian Kuhn

History

#1 - 2014-01-31 21:15 - Gerrit Code Review

- Status changed from New 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/27218

#2 - 2014-01-31 21:33 - Daniel Siepmann

Setup for disabling cache should be:

```
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['cache_core']['backend'] =
'\TYPO3\CMS\Core\Cache\Backend\NullBackend';
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['cache_classes']['backend'] =
'\TYPO3\CMS\Core\Cache\Backend\TransientMemoryBackend';
```

#3 - 2014-01-31 22:30 - Daniel Siepmann

- Status changed from Under Review to Resolved
- % Done changed from 0 to 100

Applied in changeset commit:ebe55ee9f48cea042fb45d15218f145669476eb3.

#4 - 2014-02-01 11:00 - Daniel Siepmann

To disable specific parts of *cache_core*, you can write your own Backend and ignore *set* for this.
Here is an example Backend to disable *ext_localconf* and *ext_tables* Cache:

```
<?php

namespace VENDOR\ExtName\Cache\Backend;

/**
 * A simple backend for "cache_core"
 *
 * @author Daniel Siepmann <daniel.siepmann@typo3.org>
 */
class CoreBackend extends \TYPO3\CMS\Core\Cache\Backend\SimpleFileBackend {

    protected $disabledCaches = array();

    /**
     * Saves data in a cache file.
     *
     * @param string $entryIdentifier An identifier for this specific cache entry
     * @param string $data The data to be stored
     * @param array $tags Tags to associate with this cache entry
     * @param integer $lifetime Lifetime of this cache entry in seconds. If NULL is specified, the default lifetime is used. "0" means unlimited
lifetime.
     * @return void
     * @throws \TYPO3\CMS\Core\Cache\Exception if the directory does not exist or is not writable or exceeds the maximum allowed path
```

```
length, or if no cache frontend has been set.
     * @throws \TYPO3\CMS\Core\Cache\Exception\InvalidDataException if the data to bes stored is not a string.
     * @throws \InvalidArgumentException
     */
    public function set($entryIdentifier, $data, array $tags = array(), $lifetime = NULL) {
        do {
            if (empty( $this->disabledCaches )) {
                continue;
            }
            foreach ($this->disabledCaches as $identifierPrefix) {
                if (strpos( $entryIdentifier, $identifierPrefix ) === 0) {
                    return;
                }
            }

        } while( false );
        parent::set($entryIdentifier, $data, $tags, $lifetime);
    }

    public function setDisableForIdentifierPrefixes( array $prefixesForDisablingCache ) {
        $this->disabledCaches = $prefixesForDisablingCache;
        return $this;
    }

}

?>
```

Add this Backend to your Extensions and configure it inside your *AdditionalConfiguration.php*:

```
    // Setup our custom Cache Backend for early bootstrap.
    TYPO3\CMS\Core\Core\Bootstrap::getInstance()->getEarlyInstance( 'TYPO3\CMS\Core\Core\ClassLoader' )
        ->setRuntimeClassLoadingInformationFromAutoloadRegistry(array(
            'VENDOR\ExtName\Cache\Backend\CoreBackend' => PATH_site .
'typo3conf/ext/ext_key/Classes/Cache/Backend/CoreBackend.php',
        )
    );

    $GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['cache_core']['backend'] =
'VENDOR\ExtName\Cache\Backend\CoreBackend';
    $GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['cache_core']['options'] = array(
        'disableForIdentifierPrefixes' => array(
            'ext_localconf',
            'ext_tables',
        )
    );
```

Replace ExtName and VENDOR.

You can configure all existing prefixes inside the options and disable all specific parts of *cache_core*.

#5 - 2014-02-01 15:46 - Christian Kuhn

- Parent task set to #49162

#6 - 2014-06-11 13:40 - Daniel Siepmann

A teammate points me to the solution using the lifetime. Just set it to 1. So each cache will be invalid as soon as he will be fetched again.

The only negative part is the write part. Because every cache entry will be written even as he will never be used.