

## Core - Bug # 61670

<b>Status:</b>	Rejected	<b>Priority:</b>	Should have
<b>Author:</b>	Philipp Wrann	<b>Category:</b>	Caching
<b>Created:</b>	2014-09-17	<b>Assigned To:</b>	
<b>Updated:</b>	2015-05-05	<b>Due date:</b>	
<b>TYPO3 Version:</b>	6.2		
<b>PHP Version:</b>	5.3		
<b>Complexity:</b>			
<b>Is Regression:</b>	No		
<b>Sprint Focus:</b>			
<b>Subject:</b>	cleared cache_classes causes fatal error		
<b>Description</b>			
<p>When configuring the cache_classes to apc backend it may happen (apache reload for example) that the cache gets flushed, thus the cached code will not work anymore, fatals are thrown from bootstrap.</p> <p>A on-the-fly class cache is not possible, i am aware of that (performance)</p> <p>An additional check in the autoload function would also be a performance loss</p> <p>But it might be a solution to ensure those caches are stored in the same backend.</p> <p>So if you make the cache_classes cache implement the PhpCapeable Interface it would effectively make those caches to be stored in typo3temp/Cache/Code</p> <p>That would make the bootstrap more robust.</p> <p>Additionally you should ensure all Classes used in the bootstrap are accessible via convention, so event if the cache_classes is gone the bootstrap does not throw a fatal error. If i remove all t3lib_div from my localconf files (some older extension) there is still a fatal because of some logger from the core, that is not accessible via TYPO3 Naming Convention.</p> <p>Fatal error: Interface 'Psr\Log\LoggerInterface' not found in .../typo3_src/typo3/sysex/core/Classes/Log/Logger.php on line 24</p> <p>Alternatively you could also check the cache_classes cache very early in the bootstrap, if its empty recreate it...</p>			
<b>Related issues:</b>			
duplicates Core - Bug # 60760: Race condition during system cache flush			<b>Needs Feedback</b> 2014-08-04

### History

#### #1 - 2014-09-18 09:53 - Philipp Wrann

Why a single cache entry for each class, wouldnt it be simpler to cache the complete cache\_classes as associative array and load that array in the autoloader on bootstrap?

If the cache is empty create it... That would be robust and maybe also faster, wouldnt it?

#### #2 - 2014-10-16 09:39 - Philipp Wrann

- File apc.png added

Please have a look:

[http://forum.typo3.org/index.php/m/721058/#msg\\_721058](http://forum.typo3.org/index.php/m/721058/#msg_721058)

Either my configuration is wrong, i dont get something or the caching behaviour of some caches does not qualify for APC Backend.

###

My apc config:

Runtime Settings

```
apc.cache_by_default 1
apc.canonicalize 1
apc.coredump_unmap 0
apc.enable_cli 1
apc.enabled 1
apc.file_md5 0
apc.file_update_protection 0
apc.filters
apc.gc_ttl 0
apc.include_once_override 0
apc.lazy_classes 0
apc.lazy_functions 0
apc.max_file_size 7M
apc.mmap_file_mask /tmp/apc.tEOnNw
apc.num_files_hint 0
apc.preload_path
apc.report_autofilter 0
apc.rfc1867 1
apc.rfc1867_freq 0
apc.rfc1867_name APC_UPLOAD_PROGRESS
apc.rfc1867_prefix upload_
apc.rfc1867_ttl 0
apc.serializer default
apc.shm_segments 1
apc.shm_size 2048M
apc.slam_defense 1
apc.stat 0
apc.stat_ctime 0
apc.ttl 0
apc.use_request_time 1
apc.user_entries_hint 0
apc.user_ttl 0
apc.write_lock 1
```

###

My Cacheconfig is as follows (located in AdditionalConfiguration.php):

```
$pagecache = $systemcache = array(
'backend' => 'TYPO3\CMS\Core\Cache\Backend\ApcBackend',
'frontend' => 'TYPO3\CMS\Core\Cache\Frontend\VariableFrontend',
'options' => array('defaultLifetime' => 0)
);

$pagecache['options']['defaultLifetime'] = 21600;
$pagecache['groups'] = array('pages','all');
$systemcache['groups'] = array('system');
$systemcache['options']['defaultLifetime'] = 0;
```

```

$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['cache_pages'] = $pagecache;
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['cache_pagesection'] = $pagecache;
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['cache_hash'] = $pagecache;
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['cache_rootline'] = $pagecache;
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['fluidchunks'] = $pagecache;
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['pidlist'] = $pagecache;
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['subtypes'] = $pagecache;
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['news_categorycache'] = $pagecache;

$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['extbase_object'] = $systemcache;
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['extbase_reflection'] = $systemcache;
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['extbase_datamapfactory_datamap'] = $systemcache;
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['extbase_typo3dbbackend_tablecolumns'] = $systemcache;
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['generatedflex'] = $systemcache;
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['l10n'] = $systemcache;
$GLOBALS['TYPO3_CONF_VARS']['SYS']['caching']['cacheConfigurations']['extbase_typo3dbbackend_queries'] = $systemcache;

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

```

###

TSConfig for users:

options.clearCache.pages = 1

options.clearCache.all = 1

### #3 - 2015-05-05 16:20 - Christian Kuhn

- Status changed from New to Rejected

Putting all caches into APC backend is a bad idea for various reasons. Especially for big caches like pages, and those that use tagging. The "logger not found issue" part of this report is a duplicate of #60760 - I'll now close this one here as duplicate of the other one.

## Files

apc.png	41.7 kB	2014-10-16	Philipp Wrann
---------	---------	------------	---------------