TYPO3.Flow - Feature # 43193

Status:	Rejected	**Priority:**	Should have
Author:	Christopher Hlubek	**Category:**	
Created:	2012-11-21	**Assigned To:**	Christopher Hlubek
Updated:	2012-11-21	**Due date:**	
PHP Version:			
Has patch:	No		
Complexity:			
Subject:	Optimize Classloader		

Description

The default FileBackend does an expiration check on any call to require, which costs quite some syscalls. For the classes cache we could generally disable expiration (maybe with a backend option). For frozen packages this is already the case, but in Production a package cannot be frozen.

History

#1 - 2012-11-21 12:18 - Robert Lemke

- Status changed from New to Rejected

That's what freezable cache backends are for. In Production Flow uses a freezable cache backend for code caches and freezes it as soon as proxy classes have been written.

#2 - 2012-11-21 12:23 - Christopher Hlubek

Not exactly (there is no frozen code cache), but I also didn't notice, that for code caches the SimpleFileBackend is used, which doesn't do any expiration checks.

Robert Lemke wrote:

> *That's what freezable cache backends are for. In Production Flow uses a freezable cache backend for code caches and freezes it as soon as proxy classes have been written.*

#3 - 2012-11-21 12:27 - Robert Lemke

Christopher Hlubek wrote:

> *Not exactly (there is no frozen code cache), but I also didn't notice, that for code caches the SimpleFileBackend is used, which doesn't do any expiration checks.*
>
> *Robert Lemke wrote:*
>
> > *That's what freezable cache backends are for. In Production Flow uses a freezable cache backend for code caches and freezes it as soon as proxy classes have been written.*

I was referring to "FreezableBackendInterface". In production the compiler will detect freezable cache backends and then freeze the code caches:

```
        if ($this->bootstrap->getContext()->isProduction() && $classesCacheBackend instanceof FreezableBackendInterface) {
            $classesCache->getBackend()->freeze();
        }
```

(see CoreCommandController)