Core - Bug # 56317

Epic # 55070 (Accepted): Workpackages

Epic # 55065 (New): WP: Overall System Performance (Backend and Frontend)

Bug # 52949 (Resolved): Speed decrease since 4.5

Bug # 56307 (Accepted): Findings from Bootstrap/PackageManager/ClassLoader analyses

Status:	New	**Priority:**	Should have
Author:	Steffen Ritter	**Category:**	
Created:	2014-02-26	**Assigned To:**	
Updated:	2014-02-26	**Due date:**	

TYPO3 Version:	6.2
PHP Version:	
Complexity:	
Is Regression:	No
Sprint Focus:	

Subject:	Cache Local and Default Configuration to prevent duplicate File Access

Description

See configurationManager
```
    /**
     * @var array
     */
    protected static $defaultConfiguration = NULL;
    /**
     * @var array
     */
    protected static $localConfiguration = NULL;
    /**
     * @var string Absolute path to typo3conf directory
     */
    protected $pathTypo3Conf = PATH_typo3conf;
    /**
     * Writing to these configuration pathes is always allowed,
     * even if the requested sub path does not exist yet.
     *
     * @var array
     */
    protected $whiteListedLocalConfigurationPaths = array(
        'EXT/extConf',
        'EXTCONF',
        'INSTALL/wizardDone',
        'DB',
        'SYS/caching/cacheConfigurations',
    );
    /**
     * Return default configuration array
     *
     * @return array
     */
    public function getDefaultConfiguration() {
        if (static::$defaultConfiguration === NULL) {
            static::$defaultConfiguration = require_once $this->getDefaultConfigurationFileLocation();
```

```php
        }
        return static::$defaultConfiguration;
    }
/**
 * Get the file location of the default configuration file,
 * currently the path and filename.
 *
 * @return string
 * @access private
    */
    public function getDefaultConfigurationFileLocation() {
        return PATH_site . $this->defaultConfigurationFile;
    }
/**
 * Return local configuration array typo3conf/LocalConfiguration.php
 *
 * @return array Content array of local configuration file
    */
    public function getLocalConfiguration() {
        if (static::$localConfiguration === NULL) {
            static::$localConfiguration = require_once $this->getLocalConfigurationFileLocation();
        }
        return static::$localConfiguration;
    }
```

History