

## TYPO3.Flow - Bug # 3513

<b>Status:</b>	Resolved	<b>Priority:</b>	Must have
<b>Author:</b>	Robert Lemke	<b>Category:</b>	Log
<b>Created:</b>	2009-05-29	<b>Assigned To:</b>	Robert Lemke
<b>Updated:</b>	2010-10-20	<b>Due date:</b>	
<b>PHP Version:</b>			
<b>Has patch:</b>			
<b>Complexity:</b>			
<b>Affected Flow version:</b>			
<b>Subject:</b>	Created logfiles are read-only for either command line user or web user		
<b>Description</b>			
<p>The logfiles created by the FileBackend automatically are owned by the current user running the script - that may be the webserver's user if FLOW3 is accessed from the web or another user if called from the command line. The problem is that whoever runs FLOW3 first, owns the log file and the default access rights are something like <code>-rw-r--r-</code>. If the other user tries to run FLOW3, it will fail because it can't write to the log file.</p> <p>Therefore we should set the access rights to <code>-rw-rw---</code> upon creating the file. This should happen in the <code>open()</code> method of the FileBackend.</p>			
<b>Related issues:</b>			
related to TYPO3.Flow - Bug # 3534: File permission issues		<b>Resolved</b>	<b>2009-06-02</b>

### Associated revisions

#### Revision 6dccb801 - 2009-05-29 11:26 - Robert Lemke

- FLOW3: (Log) The FileBackend now chmods the newly created log file with 0660. Resolves #3513

#### Revision b3431ea7 - 2009-05-29 12:23 - Robert Lemke

- FLOW3: (Log) Okay, my last commit contained some plain stupid code ... Well, this fixes it. The FileBackend now checks if the logfile URL is a plain file or some userspace stream handler (such as the ones we use in unit tests). If it's a plain file, the newly created file is chmodded 0666 because we currently don't have a better solution at hand. Addresses #3513

#### Revision 3eb99520 - 2009-06-02 10:57 - Robert Lemke

- FLOW3: (Log) The FileBackend now supports a new option "createParentDirectories". When turned on (default = off), the logger creates the full path to the log file if the directories don't yet exist. Addresses #3513
- FLOW3: (Environment) Added a new function `getSAPIType()` which returns the *type* of PHP's SAPI - i.e. "Web" or "CLI". Addresses #3513
- FLOW3: Changed the settings for the log file URL. Now log files for CLI and Web reside in dedicated sub directories. Resolves #3513
- FLOW3: (MVC) Whyever, the AbstractController only supported Web Requests. Fixed that.

#### Revision 14240e94 - 2009-06-02 10:57 - Robert Lemke

- FLOW3: (Log) The FileBackend now supports a new option "createParentDirectories". When turned on (default = off), the logger creates the full path to the log file if the directories don't yet exist. Addresses #3513
- FLOW3: (Environment) Added a new function `getSAPIType()` which returns the *type* of PHP's SAPI - i.e. "Web" or "CLI". Addresses #3513
- FLOW3: Changed the settings for the log file URL. Now log files for CLI and Web reside in dedicated sub directories. Resolves #3513

- FLOW3: (MVC) Whyever, the AbstractController only supported Web Requests. Fixed that.

#### Revision ea52469b - 2009-06-02 11:00 - Robert Lemke

\*Changed the settings for the log file URL. Now log files for CLI and Web reside in dedicated sub directories. Refs #3513

#### Revision 41b79d1b - 2009-06-02 11:21 - Robert Lemke

- FLOW3: An updated version of the setfilepermission.sh. It surely is a bit too permissive but it works for now and can be optimized in later releases. Addresses #3513

### History

---

#### #1 - 2009-05-29 12:00 - Robert Lemke

- Status changed from *New* to *Resolved*
- % Done changed from 0 to 100

Applied in changeset r2459.

#### #2 - 2009-05-29 14:17 - Tim Eilers

- Status changed from *Resolved* to *New*

Setting the access right to

```
-rw-rw----
```

doesn't solve the problem.

## Imaginary environment

---

Imagine the following situation:

Webserver User = www-data

CLI User = johndoe (Group users)

Assuming the setfilepermissions.sh script has been run.

### Web call first

---

Now FLOW3 will be called by Web (f.e. the Default View) at first. The Logfile is created for the used context (lets assume Production). Then the permission will look like this (not a full ls -al, i am just typing the needed data):

```
-rw-rw---- www-data www-data FLOW3_Production.log
```

Now FLOW3 is called by CLI by johndoe. You will get an exception, cause johndoe is not allowed to write to the logfile.

### CLI call first

---

In this case FLOW is called by CLI first.

The permission will look like this:

```
-rw-rw---- johndoe users FLOW3_Production.log
```

Now a FLOW3 web call will fail, cause it can't write to the logfile.

## special case: CLI user is root, not johndoe

---

Then CLI will never have a problem, cause root is allowed to write to the logfile even if it belongs to www-data. But Web will have a problem, if FLOW3 was called by CLI first, cause the log belongs to root and will not be writeable by www-data.

## solution ideas

---

- chmod the Log Directory to 777 -> from a security view very bad!
- set SGUID for Log Directory -> if CLI is called first the logfile will look like this:

```
-rw-rw---- johndoe www-data FLOW3_Production.log
```

But the problem still persists if the Web call was first, then CLI under johndoe will fail. (But not when CLI user is root. So this could be a good solution if the CLI user is root everytime. But i think the solution at the bottom is more "universal"

- precreate the Logs with the setfilepermissions.sh script -> bad, cause there could be other contextes or other logfiles later

## final solution suggestion

---

- predefined seperate log directories for Web and CLI
- both dirs will be set to webserver user by the setfilepermissions.sh script -> change the script to set the owner/group of CLI log directory to the user running that script (because he will mostly be the user running CLI calls later)
- change the Logging Framework to use the seperate log directories depending on the "context" (Web or CLI)
- remove the chmod 660 again or change it to 640 or similar (at least remove the group write permission)

There will still be a problem if the user who uploads / creates / extracts the FLOW3 distribution + runs the setfilepermissions.sh script and the user who runs CLI requests later will differ. (f.e. script user = root, CLI user = johndoe).

I think this should be covered in the documentation where CLI requests are explained. In my opinion there are not most cases where that happens. Often CLI's are used at root and if it is another user it is often the one who uploads the distribution and runs the script.

### #3 - 2009-06-01 17:57 - Tim Eilers

- File *patch3513\_distribution.diff* added

- File *patch3513\_flow3.diff* added

Please check the attached patches. I don't want to commit them by myself (and i can't, cause i only could do it for the package but not for distribution). I also didn't know how to show directory diffs. So you have to create Data/Logs/Web and Data/Logs/CLI in the distribution.

The idea behind it:

I did not put the SAPI name into the Log File path or something else, because it could be something like "apache2handler" or "cgi" (see <http://de.php.net/manual/en/function.php-sapi-name.php> and i think that is not very intuitive. So all requests coming from the web are collected into the "Web" Folder.

Perhaps this is not the final solution but an idea how to solve it...

### #4 - 2009-06-02 09:40 - Robert Lemke

- Status changed from New to Accepted

- Assigned To set to Robert Lemke

- % Done changed from 100 to 50

### #5 - 2009-06-02 11:00 - Robert Lemke

- Status changed from Accepted to Resolved

- % Done changed from 50 to 100

Applied in changeset r2517.

## Files

---

patch3513_distribution.diff	1.7 kB	2009-06-01	Tim Eilers
patch3513_flow3.diff	4.2 kB	2009-06-01	Tim Eilers