

TYPO3.Flow - Bug # 3534

Status:	Resolved	Priority:	Must have
Author:	Tim Eilers	Category:	Core
Created:	2009-06-02	Assigned To:	Robert Lemke
Updated:	2010-10-20	Due date:	
PHP Version:			
Has patch:			
Complexity:			
Affected Flow version:			
Subject:	File permission issues		
Description			
<p>There are still some file permission issues, which won't kill you, but are not perfectly clean in sense of security and beautiness:</p> <ul style="list-style-type: none">- package manager creates 777-ed folders- setfilepermissions.sh gives group write permission to some folders. this maybe fine on MacOS but not on some Linux environments. i think this was done for the logfile problem, but it isnt needed anymore- setting folder and file permissions should be mentioned in documentation, because it is not everywhere possible to run the setfilepermissions.sh script. it should also cover situations where it is not possible to "chown" to webserver user (then set the needed folders to write permission for others aka 777 or 757) or special suexec/fastcgi environments (then changing anything isn't needed at all!).- Public/Resources needs webserver's write permission (perhaps that is already fixed)- there are rare cases where the CLI user is not root, that should be no problem at the moment, but always kept in mind...- Data/Temporary is 777-ed at the moment, too. it has to, since both CLI and Web requests create some hash-named folders. perhaps this could be changed to CLI/Web subfolders like it was done for the Logfiles?			
Related issues:			
related to TYPO3.Flow - Bug # 3513: Created logfiles are read-only for either...		Resolved	2009-05-29
related to TYPO3.Flow - Bug # 3569: setfilepermissions.sh: with non-writable ...		Resolved	2009-06-04

History

#1 - 2009-06-04 14:49 - Robert Lemke

- Target version changed from 283 to 1.0 alpha 2

Re: Data/Temporary is 777-ed at the moment, too. it has to, since both CLI and Web requests create some hash-named folders. perhaps this could be changed to CLI/Web subfolders like it was done for the Logfiles?

The temporary directories are already unique to SAPI type and process user name (if supported). The information is contained in the hashed directory name.

#2 - 2009-06-04 21:29 - Tim Eilers

Robert Lemke wrote:

Data/Temporary is 777-ed at the moment, too. it has to, since both CLI and Web requests create some hash-named folders. perhaps this could be changed to CLI/Web subfolders like it was done for the Logfiles?

The temporary directories are already unique to SAPI type and process user name (if supported). The information is contained in the hashed directory name.

yep, i know. The hashed directory names are a problem. It is also a problem that you didn't put the "Web" and "CLI" subfolders of "Logs" into svn / release. Of course in many situations it is no problem and with 777ed folders it wont be in the future. But i would like to have a clean filepermission structure without 777ed folders. I like the clean and solid development of FLOW3 so this little file permission thing should be also clean and good. Perhaps i create a little screencast with the problem i am trying to express :)

#3 - 2009-06-25 08:49 - Robert Lemke

- Target version changed from 1.0 alpha 2 to 283

#4 - 2009-07-06 14:52 - Robert Lemke

- Priority changed from Should have to Must have

#5 - 2009-07-06 14:58 - Robert Lemke

Random additional notes:

- The Public/* directories must be owned by the webserver's user because "touch" seems only to work if the file is owned by the current user
Warning: touch() [function.touch]: Utime failed: Operation not permitted in
/Users/Shared/Sites/dev/flow3/dist/Packages/Global/FLOW3/Classes/Resource/Publisher.php line 198
- Depending on how we decide about the general concept of having a GUI or CLI for the main admin tasks we possibly don't even have to cope with ownership at all (WP model)

#6 - 2009-07-06 18:35 - Tim Eilers

Robert Lemke wrote:

- Depending on how we decide about the general concept of having a GUI or CLI for the main admin tasks we possibly don't even have to cope with ownership at all (WP model)

Yep, the self-updating WP is really nice, but you still need to deal with ownership, cause WP can only do his work if it all belongs to webserver's user. (Or is at least writeable for him...). And please don't depend on a CLI. Most webspaces won't have one (think of the TYPO3 future of FLOW3 ;D)

I thought about writing a little documentation part telling how to set which ownership in different situations (root-access, shell-access, ftp-access, mod_php, FCGI/suexec, etc. pp.). How about that?

#7 - 2009-07-13 10:51 - Robert Lemke

- Target version changed from 283 to 1.0 alpha 3

#8 - 2009-07-21 11:20 - Robert Lemke

- Category set to Core
- Status changed from New to Resolved
- Assigned To set to Robert Lemke
- % Done changed from 0 to 100

After playing with the new setfilepermissions.sh script introduced in r2790 I am confident that we solved the basic file permission issues for now.