For a new feature we are adding into 1.10.x, we need a place to store images, that should be out of courses (it's global). At the moment, when a user adds a watermark in the global configuration, we store the image in main/default_course_documents/images/, but this is less than ideal as, supposedly, this folder serves as a base to copy all the example content into a new course.

Badges images will be many (not just one watermark), so it is even less ideal.

Considering 1.10.0 should be a transition between 1.9.x and 2.0, I think it makes sense to start using the data directory progressively.

That's why I asked Angel to create the data/ directory and a "badges/" subdirectory.

Any objection?
We need to check if chash will not get confused about the version if there is a data/ directory.

The next (easy) step would be to move the archive/ directory into data/

Then main/upload/users/

Then main/lang/ and main/css/ ?

Then (brrrr) courses/

But all these steps are really optional and I would prefer to leave them for 2.0, except the archive/ directory: if we move it into data/, then we are not adding a step to the Chamilo installer, which would be nice.

Related issues:
Related to Chamilo LMS - Feature #5871: Make Chamilo debian-packageable

Associated revisions
Revision 9438dbd - 23/04/2015 15:51 - Julio Montoya
Add new file structure see #7522

Revision 7e580a87 - 23/04/2015 16:02 - Julio Montoya
Changing paths see #7522

Revision f6096b1b - 24/04/2015 12:53 - Julio Montoya
Updating paths see #7522

Revision 05321a4f - 29/04/2015 10:37 - Julio Montoya
Move main/css to app/Resources/public/css see #7522

Revision c73bbbc4 - 29/04/2015 11:39 - Julio Montoya
Replace css files from main/css to app/Resources/public/css see #7522

History
#1 - 12/02/2015 02:42 - Torkil Zachariassen
If /data, someday, is going to be the one and only directory that needs rw (0700 in *nix) permissions, configuring SELinux will become easy and clean. thumbs up ...torkil...
It's already the case in the (dev) 2.0 version. So this is the final destination, but we're still a long way from 2.0. 1.10.0 is a transitional version to let us hold until 2.0.

The idea of the "data" folder is that this information is not available online (for chamilo 2.0) To access any item inside that folder, you need to check if the user has the permission. For example to access:

data/course/ABC/document/image.png

A controller checks if the user has permission to access this file.

data/courses  (ex courses)

If you want to add "public" content this should be moved inside a new public folder called "web"

For example for the uploads of the user:

web/uploads/badges (badges images upload by user?)
web/uploads/user/* (profile images upload by user) (ex main/upload/users/)
web/uploads/groups/* (group images)
web/uploads/theme/css (ex main/css/mytheme specific upload files by user)

Specific Chamilo content (no upload by user is allowed)

web/default_course_document/etc
web/badges
web/etc ..

We could start creating a symfony2 data structure using this folders to deal with that archive folder:

app/cache
app/log

So the migration could be more smoothly.

For main/lang/ I don't have an answer yet. But the idea is to add a file inside a app/Resource/translations/es.po that will overwrite all Spanish terms.

My goal is double here:
#. Make Chamilo "packageable" for Linux distributions (which implies having only one variable content directory)
#. Smooth the migration and integration of Symfony components

For the second one, I am less worried. However, to comply with the first one, we need to have only one "data" (var) directory that can be moved around to another disk. As such, the idea of splitting content between web/ and data/ doesn't please me at all. We are trying to escape the current situation where we have to modify 6 different folders permissions to install Chamilo, remember?

As such, to me, this should all go into the data/ directory, and then maybe inside the data directory you could have web/ and some other subdirectory if you want, but I want only one directory of data that can be changed by the user in total.

Given this requirement, what do you suggest?

In symfony2 there's a way to build debian packages:


sf2debpkg:
This tool helps you build a native Debian package for your Symfony project.

https://github.com/liip/sf2debpkg

I think you can't truly have only one folder to change permissions that means that all "data" must be together, configuration files, logs, cache, course
data, user uploads, vendor files, assets, lang files, etc, etc. And is very dangerous to combine them all.

Imagine you want to delete cache but you forget to add something and you delete all your data :) The idea is that the only accessible point is "web" and the rest of Chamilo is "hidden" (by apache).

Because one requirement only used for some very specific cases you want to change the all structure. It doesn't seem logic for me but I understand your point to simplify things.

Another solution (hack) is making a "link" for example there will be a folder "data/web" that link to "web". I think we should just stick with the default Symfony2 structure, other developers will understand that classic structure.

#6 - 18/02/2015 02:02 - Torkil Zachariassen
I am not really sure if these comments fit the bill. But we moved the configuration files to /etc with these rather restrictive permissions:

```
[root@example etc]# ls -ld /etc/chamilo
drwxr-x---. 2 root apache 4096 des 15 18:02 /etc/chamilo
```

```
[root@example]# ls -l /etc/chamilo
total 64
-rw-r-----. 1 root apache 10601 des 15 18:02 configuration.php
```

which works just fine.

In our line of thought the web server (group: apache) should have readonly accesses to some things (/var/www/chamilo), and readwrite (0770) access to a few other things, say /var/www/chamilo/data

( Still, on RH7 the home of apache is /usr/share/httpd ... )

...torkil...

#7 - 02/04/2015 22:50 - Yannick Warnier
Julio Montoya wrote:

Imagine you want to delete cache but you forget to add something and you delete all your data :)

Well, that's just a question of having a data/cache/ folder. It's the same risk as doing "rm -rf /" or, in your current Chamilo installation, if you delete the main Chamilo folder (remember, data/ will be a new folder, so no direct relation with archive/). If the documentation clearly states that the cache folder is data/cache/, I can't see this happening by mistake, unless if you deserve it :)

The idea is that the only accessible point is "web" and the rest of Chamilo is "hidden" (by apache).

I understand that. Some web projects (like Dolibarr) used to put their "web" files in an htdocs/ directory and ask the users to configure their virtualhost to point to that directory as a root directory. As we progress in time and versions, this is becoming less of an issue as more and more systems offers auto-installers that include Chamilo (Softaculous, Instalator, Docker, Juju, Debian packages, etc), which reduce the amount of work needed for any person wanting to install Chamilo.

Because one requirement only used for some very specific cases you want to change the all structure. It doesn't seem logic for me but I understand your point to simplify things.

I understand why you're worried, but in fact Debian only asks to respect the HFS (Hierarchical File System), which is valid for most Linux and BSD distributions, which are the 90% of all platforms on which Chamilo is installed. So this is not really specific. Debian packaging is strict because it requires the respect of standards, and we can all see the positive influence using PSR-1 and PSR-2 has had on Chamilo lately.

Another solution (hack) is making a "link" for example there will be a folder "data/web" that link to "web". I think we should just stick with the default Symfony2 structure, other developers will understand that classic structure.

Not really a solution. It also requires a higher level of skills from the Chamilo installer, because links are not transported into .zip and .tar.gz packages.

#8 - 02/04/2015 22:52 - Yannick Warnier
Torkil Zachariassen wrote:

I am not really sure if these comments fit the bill. But we moved the configuration files to /etc with these rather restrictive permissions:

[...]

18/03/2020
Thanks, but that isn’t really the point, no. The configuration files can easily be moved to /etc/chamilo manually. The issue is because some Chamilo folders are currently very difficult to move out of the main Chamilo folder and we want to make this easier. The config directory is not a difficult one to move.

#9 - 02/04/2015 23:18 - Yannick Warnier

Julio, in the case of the app/ structure of Symfony, is there a need for a "data/" directory?

Could everything go into app/? (given there is app/cache/). I’m not sure why data could not be app/data/ and then everything could go into app/ somehow, and so app/ would be the root folder and we wouldn’t need app/ anymore, and then we’re back to the start)

I understand we need a web/ folder for all public (non-code) resources, and that app/ should be where the code is (but then app/cache/ and app/log/ don’t make much sense to me).

Ideally, no PHP file should find it’s way into data/, so we can set data/ as non-executable (exclude the execution of PHP files in this folder in general).

Let’s start proposing general rules to work on. This is the list of problematic folders:

<table>
<thead>
<tr>
<th>Folder</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>courses/</td>
<td>currently stores courses data. Need permissions check</td>
</tr>
<tr>
<td>archive/</td>
<td>currently stores temporary files of all kinds. Some of these should be access-protected, but I don’t think they are</td>
</tr>
<tr>
<td>archive/twig/</td>
<td>stores the template cache files</td>
</tr>
<tr>
<td>home/</td>
<td>currently stores homepage files, uploaded/created by users, not necessarily public, but currently all public</td>
</tr>
<tr>
<td>searchdb/</td>
<td>stores the fulltext indexing databases for Xapian. No direct access should be allowed</td>
</tr>
<tr>
<td>documentation/</td>
<td>Public. Changes when code changes.</td>
</tr>
<tr>
<td>main/inc/conf/</td>
<td>Configuration. Should be moved to /etc/chamilo/ on a Linux system. Should not be public</td>
</tr>
<tr>
<td>main/upload/*</td>
<td>users, courses, sessions and badges folders, should not be public</td>
</tr>
<tr>
<td>main/default_course_document/</td>
<td>Default media resources for courses. Can be public. Also servers to store the new document templates, I think</td>
</tr>
<tr>
<td>main/lang/</td>
<td>Language files. Most of them only updated with the code, but some of them can be updated through the sublanguages interface. Can be public, no issue here. Currently in PHP format (i.e. dangerous if modifiable), they should be moved to gettext soon. Usually updated with code, some subfolders can be extended by the user</td>
</tr>
<tr>
<td>main/css/</td>
<td>Same as language files but the root and default/ subfolder come with code, while an extension is possible by the user</td>
</tr>
<tr>
<td>main/template/</td>
<td>Same as main/lang/ and main/css/ in the future (currently only editable through direct access to the filesystem)</td>
</tr>
<tr>
<td>main/</td>
<td>The main code folder</td>
</tr>
</tbody>
</table>

So, translating this into new folder (this is a work in progress) into 1.10.x (and taking your points into account, Julio), my ideal evolution would be:

<table>
<thead>
<tr>
<th>Folder in 1.9.x</th>
<th>Folder in 1.10.x</th>
<th>Folder in 2.x</th>
</tr>
</thead>
<tbody>
<tr>
<td>courses/</td>
<td>data/courses/</td>
<td>app/data/courses</td>
</tr>
<tr>
<td>home/ (default)</td>
<td>data/home/</td>
<td>app/data/home</td>
</tr>
<tr>
<td>searchdb/</td>
<td>data/plugin/searchdb/</td>
<td>app/uploads/plugins/xapian/searchdb/</td>
</tr>
<tr>
<td>main/default_course_document/</td>
<td>data/default_courses_documents/</td>
<td>app/data/default_courses_documents/</td>
</tr>
<tr>
<td>archive/</td>
<td>data/temp/</td>
<td>app/cache</td>
</tr>
<tr>
<td>archive/twig/</td>
<td>data/cache/twig/</td>
<td>app/cache</td>
</tr>
<tr>
<td>main/inc/conf/</td>
<td>/config</td>
<td>app/config/</td>
</tr>
<tr>
<td>documentation/</td>
<td>documentation/</td>
<td>documentation/</td>
</tr>
<tr>
<td>main/upload/*</td>
<td>data/upload/*</td>
<td>web/uploads/* (only public images)</td>
</tr>
<tr>
<td>main/lang/</td>
<td>data/lang/</td>
<td>app/Resources/translations (custom lang files) src/CoreBundle/Resources/translations (system lang files)</td>
</tr>
</tbody>
</table>
Doing this table, I realize that the main/template/ folder will need some additional work to be able to move it to data/ (if we want to exclude PHP execution from the data/ folder): it currently contains a few .php files (that can be included by other php files and their code executed)

The lang/ folder will be a temporary exception (until we move to gettext)

Julio, could you edit this last table and put what you suggest, so we can find a compromize (urgently needed).

#10 - 02/04/2015 23:19 - Yannick Warnier
- Assignee changed from Yannick Warnier to Julio Montoya
- % Done changed from 0 to 20

#11 - 07/04/2015 09:50 - Julio Montoya
Yannick Warnier wrote:

Julio, in the case of the app/ structure of Symfony, is there a need for a "data/" directory?

We don't need the "data" it is going to be replaced with "app".

Could everything go into app/? (given there is app/cache/, I'm not sure why data could not be app/data/ and then everything could go into app/ somehow, and so app/ would be the root folder and we wouldn't need app/ anymore, and then we're back to the start)

Yes, actually that was one of my proposals a long time ago #5871 to create a "app/data"

I understand we need a web/ folder for all public (non-code) resources, and that app/ should be where the code is (but then app/cache/ and app/log/ don't make much sense to me).

app/cache and app/log don't make sense to you? why? app/cache is the same as our classic "archive" and app/logs it will be a new place instead of using "archive" for everything.

Ideally, no PHP file should find it's way into data/, so we can set data/ as non-executable (exclude the execution of PHP files in this folder in general).

Correct

I just edited the table.

#12 - 21/04/2015 12:39 - Julio Montoya
- Assignee changed from Julio Montoya to Yannick Warnier

#13 - 22/04/2015 06:40 - Yannick Warnier

I am rethinking this task from an additional point of view: storing on the cloud. There is a lot going on around the cloud and Chamilo, and I'd like to have a mechanism that allows us to store new installations of Chamilo on the cloud, using dynamic storage, which implies writing/reading the files through web services. If all files that can be modified are in the same 2-3 folders, it will be much easier for us to build a better scalable infrastructure with shared storage.

So let's say we go for app/data, app/cache, app/log, app/courses, app/home, app/upload/users, app/upload/courses, app/upload/sessions (the three last ones are for the extra fields), app/config', app/css', app/themes', app/plugins/ (for stuff that plugins need to store)

Would that be OK for you (to implement now)?

I think if we maintain some order into the app/ directory, we can still find a way to move folders from there easily for Linux packaging. Also, could you evaluate the time required (more or less) to implement a file access manager (from Symfony or whatever useful lib) that we could use to access normally through the filesystem but be able to reconfigure to use through web services? (for cloud storage like Amazon ECS, etc) (forget that one for now)
Yannick Warnier wrote:

I am rethinking this task from an additional point of view: storing on the cloud. There is a lot going on around the cloud and Chamilo, and I'd like to have a mechanism that allows us to store new installations of Chamilo on the cloud, using dynamic storage, which implies writing/reading the files through web services. If all files that can be modified are in the same 2-3 folders, it will be much easier for us to build a better scalable infrastructure with shared storage.

So let's say we go for app/data, app/cache, app/log, app/courses, app/home, app/upload/users, app/upload/courses, app/upload/sessions (the three last ones are for the extra fields), app/config/, app/css/, app/themes/, app/plugins/ (for stuff that plugins need to store)

Would that be OK for you (to implement now)?

I think if we maintain some order into the app/ directory, we can still find a way to move folders from there easily for Linux packaging. Also, could you evaluate the time required (more or less) to implement a file access manager (from Symfony or whatever useful lib) that we could use to access normally through the filesystem but be able to reconfigure to use through web services? (for cloud storage like Amazon ECS, etc) (forget that one for now)

Then "app/data" has no purpose in your description.

What's the difference between app/css/ and app/themes/?

The css, js, fonts should be here:

  app/Resources/assets/css
  app/Resources/assets/css/themes/julio_theme
  app/Resources/assets/js
  app/Resources/assets/fonts

etc

This files then are copy inside "web" the public part of Chamilo (when using symfony)

Julio Montoya wrote:

Yannick Warnier wrote:

I am rethinking this task from an additional point of view: storing on the cloud. There is a lot going on around the cloud and Chamilo, and I'd like to have a mechanism that allows us to store new installations of Chamilo on the cloud, using dynamic storage, which implies writing/reading the files through web services. If all files that can be modified are in the same 2-3 folders, it will be much easier for us to build a better scalable infrastructure with shared storage.

So let's say we go for app/data, app/cache, app/log, app/courses, app/home, app/upload/users, app/upload/courses, app/upload/sessions (the three last ones are for the extra fields), app/config/, app/css/, app/themes/, app/plugins/ (for stuff that plugins need to store)

Would that be OK for you (to implement now)?

I think if we maintain some order into the app/ directory, we can still find a way to move folders from there easily for Linux packaging. Also, could you evaluate the time required (more or less) to implement a file access manager (from Symfony or whatever useful lib) that we could use to access normally through the filesystem but be able to reconfigure to use through web services? (for cloud storage like Amazon ECS, etc) (forget that one for now)

Then "app/data" has no purpose in your description.

OK.

What's the difference between app/css/ and app/themes/?

CSS theme and Twig theme.

The css, js, fonts should be here:

  app/Resources/assets/css
  app/Resources/assets/css/themes/julio_theme
  app/Resources/assets/js
  app/Resources/assets/fonts
OK, no problem. Where does the tpl themes go?

These files then are copied inside "web" the public part of Chamilo (when using symfony)

What is the logic behing copying them if they are already there? (just asking)

#16 - 22/04/2015 16:51 - Yannick Warnier
- Assignee changed from Yannick Warnier to Julio Montoya

#17 - 23/04/2015 08:35 - Julio Montoya
- Assignee changed from Julio Montoya to Yannick Warnier

Yannick Warnier wrote:

What's the difference between app/css/ and app/themes/ ?

CSS theme andTwig theme.

Ahhh ok.

The css, js, fonts should be here:

    app/Resources/assets/css
    app/Resources/assets/css/themes/julio_theme
    app/Resources/assets/js
    app/Resources/assets/fonts

OK, no problem. Where does the tpl themes go?

tpl themes (twig templates) should go here:

    app/Resources/views

These files then are copied inside "web" the public part of Chamilo (when using symfony)

What is the logic behing copying them if they are already there? (just asking)

All the information inside app/ is not public. The only public folder is "web" (when using symfony).
You can't access to for example: http://chamilo/app/Resources/assets/css/default.css

#18 - 23/04/2015 14:17 - Julio Montoya
I found that in the skill you upload files inside data/badges...
This should be moved to app/upload/badges right?

#19 - 23/04/2015 15:21 - Yannick Warnier
Yes

#20 - 28/04/2015 14:53 - Julio Montoya
I won't implement the "app/Resources".
"Themes" will be in main/css/my_theme.
"templates" will be as always in main/templates/default.
The reason is that we don't have a mechanism to copy them in a "public" folder such as in Symfony2.

#21 - 28/04/2015 14:59 - Julio Montoya
I won't change also "main/home" because is a big old piece of code. That main/home should be replaced by a "page manager" (Doctrine CRUD). This page manager could create html files as cache if the DB is down.
#22 - 28/04/2015 15:08 - Yannick Warnier
- Assignee changed from Yannick Warnier to Julio Montoya

OK for main/templates (because there is no mechanism for the user to upload there, and as such no need to change the permissions to writeable for www-data), but I would like for home/ and main/css/ to be moved to app/ (or web/ directly) to avoid having to change the permissions for several folders at install time.

Anyway, a few is better than nothing, so if it's too complicated, let's skip it for now, but main/css/ and home/ are important.

#23 - 22/06/2015 22:51 - Yannick Warnier
- Status changed from Needs more info to Feature implemented
- % Done changed from 20 to 100

I'm closing this task and moving the rest to #7577.

At this point, we moved a series of things to the data/ directory and made considerable progress towards moving all changeable parts of Chamilo there. Well done!