

Chamilo LMS - Support #6617

Next version number after Chamilo 1.9.8: 1.10? 2.0? 10.0? 14.0?

23/08/2013 14:27 - Julio Montoya

Status:	Bug resolved	Start date:	23/08/2013
Priority:	High	Due date:	
Assignee:	Yannick Warnier	% Done:	100%
Category:	Global / Others / Misc	Estimated time:	0.00 hour
Target version:	2.0	Spent time:	0.00 hour
Complexity:	Normal	SCRUM pts - complexity:	?

Description

I remember that we discuss about the possibilities to change the version number from new Chamilo LMS (master in github).

<http://semver.org/> says:

Given a version number MAJOR.MINOR.PATCH, increment the:

- MAJOR version when you make incompatible API changes,
- MINOR version when you add functionality in a backwards-compatible manner, and
- PATCH version when you make backwards-compatible bug fixes.

So for Chamilo LMS we should embrace this standard.

Here are the current possibilities (feel free to suggest more)

The current state:

1.9.8, 1.9.9 then **1.10.0, 1.10.1 ... new big version 1.11.0**

This is wrong, because we are still in version 1!! This gives an impression that we are not moving on.

The "natural" version order according to semver.org:

1.9.8, 1.9.9 then **2.0, 2.1 new big version 3.0**

We should do this, now that there are 2 different "products" in the Chamilo Association: Chamilo LMS and Chamilo LCMS. Chamilo LCMS has now version 3 (Chamilo LCMS Connect 3.1 Bradypodion Carpenteri) and version 4 (Chamilo LCMS Connect 4 Brokesia)!!

New convention (get rid of "1.x"):

1.9.8, 1.9.9 then **10.0.0, 10.1.0 new big version 11.0.0**

Not sure if you will agree with this.

~~New convention version number based in the year.~~

~~-1.9.8, 1.9.9, then **14.0, 14.1 ... new big version 15.0**~~

This force us to release at least 1 big version every year, but this was not the case in 2013.

The idea of 3 ~~and 4~~ is:

- To get rid of the "1.x" prefix so people could easily remember "Chamilo LMS" + a version number and suggest that the project is evolving fast (which is the case).

- to manage better minor versions: 14.0 and 14.1 could include DB changes. But not DB changes from 14.0.1 to 14.0.2

- Having version like 1.8.8.6 is now just ridiculous :) hopefully for version 1.9.x we stick to 3 numbers! :)

- By "big version" I mean database changes, folder restructure, new tools, etc.

What do you think??

Related issues:

Related to Chamilo LMS - Support #6145: Packaging 1.10

Feature implemented 19/05/2013

History**#1 - 23/08/2013 14:27 - Julio Montoya**

- Subject changed from Next version name after chamilo 1.9.x to Next version number after Chamilo 1.9.8

#2 - 23/08/2013 14:30 - Julio Montoya

- Target version set to 2.0

#3 - 02/09/2013 12:19 - Julio Montoya

- Subject changed from Next version number after Chamilo 1.9.8 to Next version number after Chamilo 1.9.8: 1.10? 10.0? 13.0?

#4 - 21/11/2013 11:18 - Julio Montoya

- Status changed from New to Needs more info

- Assignee set to Yannick Warnier

#5 - 28/02/2014 13:04 - Julio Montoya

- Description updated

- Priority changed from Normal to High

#6 - 28/02/2014 13:05 - Julio Montoya

- Subject changed from Next version number after Chamilo 1.9.8: 1.10? 10.0? 13.0? to Next version number after Chamilo 1.9.8: 1.10? 2.0? 10.0? 14.0?

- Description updated

#7 - 22/04/2014 16:14 - Yannick Warnier

- Category set to Global / Others / Misc

- Status changed from Needs more info to Bug resolved

- % Done changed from 0 to 100

First of all I think "Major version" is more representative than "Next big version" :-)

At the moment the "no changes in DB between minor versions" is **very** practical. Too practical, in fact, to really generate whatever protest that happened about the stability of upgrades that we had in the past (before I implemented this concept). Also, before that, I was the one single-handedly managing the migration procedure, and was a daunting task when it had to be done for each minor version.

Next, I just released a 1.9.6.1 version (so much for sticking to 3 digits in 1.9...) just to apply the latest security patch. It is obviously an exception to the rule and hasn't even be promoted at all (it's just replaced the 1.9.6 package on the website).

Obviously, the issue here is the "1" at the beginning, not the "1" at the end.

So I'm not against making a "jump" to either 10.x or 14.x (or 15.x because I'm not sure we're going to release a 1.10 this year) **but** I'm completely against making database changes between minor versions (like you suggest between 14.0 and 14.1). I think we can manage much better without these database changes **between** major versions, mostly thanks to the "extra fields" that we are adding pretty much everywhere.

Opening the possibility of making database changes between minor versions is, in fact, the single decision most likely to push us to have 4 digits versions in the future ("oh well, no hurry, we don't need to push for the release of 15.0 because we can add this and that db change in 14.1, so let's continue in 14.*...").

All this being said, the most **natural** order of progression from now on would, of course, be to move to version 2.0. This is something that our users would understand easily (apparently they haven't really been affected by Chamilo LCMS versioning so far - most people don't know there is a version 3 or 4 - they've just been confused at some point with version 2).

So at the moment I'm still not decided. I fear that moving to version 10.0 or 15.0 would cause some confusion between our users, so I think the jump to version 15.0 would be too big.

Considering **all** these aspects, I think the best decision would be for us to go from 1.9 to (1.)10.0 and remove the prefixing "1". This is easier to explain to anybody. We could even already start with version 1.9.8 and call it "(1.)9.8" in our marketing and stuff, so people get slowly used to the concept. If we do improve our release cycle, we could then "catch up" a bit later and get to version 16.0 while in 2016, or something of that order. Taking the risk of setting a version to the year number right away is too risky for the size of our development community (not sure we could reach that goal).

OK, so taking the decision here, let's call our next major version "Chamilo LMS 10.0" and the next minor version Chamilo LMS 9.8.

Let's slowly move everything to that concept, right after releasing the 1.9.8 RC, so we have time to do proper marketing.

#8 - 22/04/2014 18:07 - Julio Montoya

I'm agree with the choice, we will follow the semver.org standard.

To be clear I'm putting an example:

Chamilo LMS 10.0.0	MAJOR version when you make incompatible API changes, database changes
Chamilo LMS 10.1.0	MINOR version when you add functionality in a backwards-compatible manner
Chamilo LMS 10.0.1	PATCH version when you make backwards-compatible bug fixes , security updates

So, for MINOR and PATCH versions you could update chamilo just unzipping a file.

#9 - 22/04/2014 18:59 - Yannick Warnier

Agreed. Just to be complete about it, MAJOR includes everything you can do for MINOR, and MINOR includes everything you can do for PATCH

#10 - 22/04/2014 19:00 - Julio Montoya

Yannick Warnier wrote:

Agreed. Just to be complete about it, MAJOR includes everything you can do for MINOR, and MINOR includes everything you can do for PATCH

Yes, that's right.

#11 - 14/11/2014 21:43 - Alexander Hu

I would vote for 2.x as next one.

Was sitting for 3 days at a customer, had some nice talks.

They wouldn't understand the big jump.