

## Re: PHP Classes suggestions (was Re: Recommendations for PHP Form Validation Script)

*Source:* <http://coding.derkeiler.com/Archive/PHP/comp.lang.php/2003-11/1471.html>

---

*From:* Manuel Lemos ([mlemos\\_at\\_acm.org](mailto:mlemos_at_acm.org))

*Date:* 11/20/03

Date: Thu, 20 Nov 2003 00:02:30 -0200

Hello,

On 11/11/2003 09:27 AM, Zurab Davitiani wrote:

>>> *Well, if you think they are not going to do versioning, then it's not  
>>> worth going through the hassle, I guess. Just send out the  
>>> notices/requests to confirm compatibility. But, I would think, as a side  
>>> note, it would be great to have an optional version field for a package  
>>> anyway, so (for authors/packages that would use them) users know if they  
>>> are up-to-date or not.*

>>

>> *Yes, that is what I think is what is viable.*

>>

>> *What about the version format? Shall I let it make it arbitrary or  
>> enforce a format like M.m.a (Major.minor.alpha like 2.7.1)?*

>

>

> *I don't see this as a big deal either way; but I don't think enforcing any  
> particular format is going to add anything meaningful. It's obvious but*

If you have versions in a specific format, the site can always validate the new versions to make sure that next version number is more than the previous.

> *just make sure you keep the date/time of each release.*

The problem is which date should be taken as release? The date/time of the last updated file or the date and time of when the author updated the version number?

>>> *Yeah, that's a good idea. A simple comment field should suffice.*

>>> *Otherwise, you would have to get involved in multiple dependency trees*

>>> *involving combinations of factors such as:*

>>>- *PHP major/minor versions*

>>>- *platforms and their major/minor versions*

>>>- *compiled in or loadable modules and their major/minor versions*

>>>- *etc.*

>>  
>>*Yes, I think I will have something like:*  
>>  
>>*– Required: dependency package is necessary to make dependent class work.*  
>>  
>>*– Base: Required package has actually a base class*  
>>  
>>*– Required alternative: dependency package provides a functionality that*  
>>*otherwise needs to be provided by another required alternative*  
>>  
>>*– Conditional: Dependency package provides functionality that is*  
>>*necessary to make the dependent package work in some conditions.*  
>>  
>>*– Optional: Dependency package provides a functionality that is not*  
>>*absolutely necessary in any conditions.*  
>>  
>>*Am I forgetting any other case?*  
>  
>  
>*In my opinion, the simpler the better. I would simply suggest a single*  
>*option – worded either "Required" or "Dependencies" or something similar;*  
>*If specified, the author would have to enter a comment as to the extent of*  
>*each dependency.*  
>  
>*Otherwise, you run the risk of confusing some developers with all those*  
>*options and, thereby, ending up with incorrect data. For example, what if*  
>*some dependencies could belong to 2 different groups on different platforms*  
>*or major PHP versions? Covering all possible scenarios increases the risk*  
>*of developer as well as user confusion. UNLESS, you implement some type of*  
>*dependency tree system with multiple possible package branches. For*  
>*example, imagine package branches for PHP4, PHP5, each with different*  
>*dependency trees.*

Yes, although some options are just special cases of others. A base class is a special case of a required dependency. I thought that could help the users understand more clearly why the dependency is really required.

>>*Either way the site still have to support individual and bulk updates.*  
>>*So, I will have something that sends out digests of file changes maybe*  
>>*hourly to avoid sending many messages to the users, which some have been*  
>>*complaining with reason.*  
>>  
>>*The problem of making it easy for authors to enter bulk changes in a*  
>>*single description is that some will make them vague. This is also a*  
>>*reason to have an approval stage for new or changed files.*  
>  
>  
>*Well, you know what my opinion is – it's up to you how you implement it. I*  
>*believe users would be well served by freshmeat–type update log – concise*  
>*and to the point. I don't think it's necessary to duplicate a CVS or a*

- > *similar individual file changelogs since the site is not a development*
- > *environment but rather a class/package repository. Well, unless you want to*
- > *take it in that direction.*

Yes, but Freshmeat is different. They do not keep track of individual files of projects. If you have a package with multiple classes and not all classes are interesting for all users, if you changed just a few classes, the users that are interested just on those will want to know what you have changed. So, I think that the list of changes that pertain just to a certain files is more helpful.

- >>*Personally, the problem that I saw with Sourceforge was not others*
- >>*forked the project but rather the attitude of some individuals of FSF*
- >>*that taken hostile measures causing eventual financial harm to VA.*
- >>
- >>*One thing is expressing public disagreement on the decision of VA to*
- >>*close the source of futher development and starting forks. Another thing*
- >>*is making bad PR to try to burn VA business in revenge.*
- >>
- >>*I think that was silly ungrateful attitude from some FSF individuals*
- >>*because Sourceforge has done a lot for the credibility and viability of*
- >>*many Open Source projects.*
- >
- >
- > *Well, this can happen to anyone, whether your project/service is open source*
- > *or not. There will always be differing opinions and [groups of] people who*
- > *act one way that others disapprove or do not appreciate. I guess the more*
- > *popular the project/service/website the more the likelihood of such*
- > *situations.*

Yes, but my point is that VA never had the need to really open their source code. They always had paid employees to cover for their development needs. They just followed the trend of opening the source code of their site but certainly they wish they have not done that after they realized what they had to put with.

For me this is well enough to warrant. I do not want nor need to put up with the same as VA, so I will not Open the Source code of the site.

- >>>>*From a strictly personal point of view, I wouldn't take a Sourceforge*
- >>>>*example as a means of conclusion that open sourcing a project like that*
- >>>>*is automatically "bad" since someone will likely take its controls over*
- >>>>*and start providing a competitive service. I mean, that's [a predictable]*
- >>>>*part of it, but not all of it. Take slashcode as an example, used by 100s*
- >>>>*(if not more) sites, livejournal, even Sourceforge, in this respect,*
- >>>>*remains extremely popular.*
- >>
- >>*Sure, but VA (OSDN) and its financial muscle is too large to compare*
- >>*with me. VA can leverage the resources to keep Sourceforge alive. Maybe*
- >>*now they are paying themselves but it took a lot of money to grow that*
- >>*large. When they are that large and popular, they do not need to fear*

>>>opening the source of the sites, unless they want to sell copies , which  
>>>was why they decided to close the source of Open Source later.  
>  
>  
> I agree to most of your opinion, but also consider one example I stated –  
> LiveJournal. They actually make their users pay a fee to use their service  
> (in certain circumstances) while offering all their software under GPL.  
> Now, there are other sites that run that software, but none as popular as  
> LiveJournal, and none (I am guessing) with as many members and  
> participants. I'm not implying that all circumstances are equal, or that  
> you should be charging users, but the model is something worth looking at;  
> and maybe even having a brainstorming session over.

Yes, but what would I have to gain with opening the source code of the site? The way I see it, only things that I do not absolutely need, not to mention the disadvantages of doing it.

Besides that, I have more reasons for not doing it that I have not mentioned.

>>>>I guess what I am trying to say is that going one way or another is not a  
>>>>simple straightforward line – it's a combination of a lot of different  
>>>>factors; and, even in Sourceforge example, I cannot deduce that  
>>>>sourceforge.net would have been a more popular service if it was NOT  
>>>>open-sourced itself. I just can't draw that line.  
>>>  
>>>No, I think your are forgetting how it happens. Sourceforge was already  
>>>very popular when they opened the source of the site. They really did  
>>>not need to do that. I am sure they regret that did it. I do not want to  
>>>regret too.  
>>>  
>>>  
>>>I'm not sure they regret it though. After all, wide media coverage and all  
>>>the fuss it generated did get them more attention and even more popularity.  
>>>Your situation is obviously different from VA's, so it's your judgement.

Maybe there were some positive aspects of advertising as you mentioned but that could only be positive because they ended up wanting to sell the site code as a product. That is not in my plans though.

OTOH, if they did not ever opened the source, they could have more hosted projects generating advertising revenue from the banners they put in their pages because there would not be alternative sites based on their code.

I am sure that today they are making much more money from the advertising there, than from the sales of licenses of their site product. They just did not know then that the advertising market would be as healthy as it is today, but at least I know that and it is one more reason to not open the source code.

comp.lang.php: Re: PHP Classes suggestions (was Re: Recommendations for PHP Form Validation Script)

--

Regards,

Manuel Lemos

Free ready to use OOP components written in PHP

<http://www.phpclasses.org/>