
Subject: user rights for svn and software release tools
Posted by [Bertram Kopf](#) on Thu, 22 Jan 2009 14:39:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi all,

I have now heard about the policy for the usage of the svn. One point is that specific directories like base, geobase, etc. are shared by a couple of experiments like Panda, Cbm and Hades. For these parts only the FairRoot IT group is allowed to modify something. This is absolutely understandable since this is the core for the individual experiments and one has to ensure that this part works properly for everybody. On the other hand I think that this could also be an obstacle for development of the panda code. If you would like to modify or add new classes, it needs some time to check whether everything fulfills the requirements.

Therefore I would propose to introduce a layer below the FairRoot base directories, e.g. PndBase, where one can implement panda specific base classes. Such a directory should have write permissions for the Panda developer. What is your opinion? Does it make sense to introduce such a directory?

With regard to this I have another question: Can somebody tell me the procedure how the files "CMakeLists.txt, Makefile.am and xyzLinkDef.h" will be created. Are there automated scripts available which create these make- and dependency-files.

Thanks in advance for your help and best regards,
Bertram.

Subject: Re: user rights for svn and software release tools
Posted by [Mohammad Al-Turany](#) on Thu, 22 Jan 2009 15:16:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hallo Bertram,

Quote:On the other hand I think that this could also be an obstacle for development of the panda code. If you would like to modify or add new classes, it needs some time to check whether everything fulfills the requirements.

Tell now this was never a problem, usually you add new classes for your experiment and not to all other! and I do not think it is a good idea that everybody can add global stuff even on one experiment level!

Quote:Therefore I would propose to introduce a layer below the FairRoot base directories, e.g. PndBase, where one can implement panda specific base classes. Such a directory should have write permissions for the Panda developer. What is your opinion? Does it make sense to introduce such a directory?

This was suggested more than one year ago for panda as we make it for Cbm, look to the SVN, cbm has CbmBase which contains specific stuff for CBM, the last meeting we also discuss this and it is in the minutes,

<http://panda-wiki.gsi.de/cgi-bin/view/Computing/Minutes13Jan2009>

We already decided some time ago to rename the mcstack directory (Which is Panda specific) to PandaBase! look to the minutes: where everybody already agreed on that!!

<http://panda-wiki.gsi.de/cgi-bin/view/Computing/Minutes19March2008>

Quote:With regard to this I have another question: Can somebody tell me the procedure how the files "CMakeLists.txt, Makefile.am and xyzLinkDef.h" will be created. Are there automated scripts available which create these make- and dependency-files.

There is no script to automatically generate these very simple stuff, there is only a template on the fairroot.gsi.de. the Makefile.am is not used any more and I will remove it and all autoconf/automake stuff

regards

Mohammad

Subject: Re: user rights for svn and software release tools
Posted by [Bertram Kopf](#) on Thu, 22 Jan 2009 16:03:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Mohammed,

Quote:

Quote:On the other hand I think that this could also be an obstacle for development of the panda code. If you would like to modify or add new classes, it needs some time to check whether everything fulfills the requirements.

Tell now this was never a problem, usually you add new classes for your experiment and not to all other! and I do not think it is a good idea that everybody can add global stuff even on one experiment level!

Yesterday I posted a message concerning the non properly treated usage of the "std::map" (message 7709 in bugs, fixes, releases"std::map not treated properly"). To fix this it would be fine to introduce a template which should go to one of the base directories. Since I have no access to these directories it is not possible for me to do it. And what about new abstract classes which one would like to introduce and which should be used in some specific directories?

Quote:

Quote:Therefore I would propose to introduce a layer below the FairRoot base directories, e.g. PndBase, where one can implement panda specific base classes. Such a directory should have write permissions for the Panda developer. What is your opinion? Does it make sense to introduce such a directory?

This was suggested more than one year ago for panda as we make it for Cbm, look to the SVN, cbm has CbmBase which contains specific stuff for CBM, the last meeting we also discuss this and it is in the minutes,

<http://panda-wiki.gsi.de/cgi-bin/view/Computing/Minutes13Jan2009>

We already decided some time ago to rename the mcstack directory (Which is Panda specific) to PandaBase! look to the minutes: where everybody already agreed on that!!

<http://panda-wiki.gsi.de/cgi-bin/view/Computing/Minutes19March2008>

Sorry that I didn't notice it. Great! Does it mean that all Panda related base classes should go into this directory?

Quote:

Quote:With regard to this I have another question: Can somebody tell me the procedure how the files "CMakeLists.txt, Makefile.am and xyzLinkDef.h" will be created. Are there automated scripts available which create these make- and dependency-files.

There is no script to automatically generate these very simple stuff, there is only a template on the fairroot.gsi.de. the Makefile.am is not used any more and I will remove it and all autoconf/automake stuff

There was an evo meeting before christmas where release management tools was one major topic. As far as I understood it there similar tools like the BaBar srt-tools are (potentially) available in the PandaRoot software. It seems to me that at least these things are not there. Does it also mean that one has to control all dependencies by hand?

Cheers,
Bertram.

Subject: Re: user rights for svn and software release tools
Posted by [Mohammad Al-Turany](#) on Thu, 22 Jan 2009 21:02:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

Quote:And what about new abstract classes which one would like to introduce and which

should be used in some specific directories?

they will go either to Pandabase or pandadata, and if you write these classes then you will have access to these directories!

Quote:There was an evo meeting before christmas where release management tools was one major topic. As far as I understood it there similar tools like the BaBar srt-tools are (potentially) available in the PandaRoot software. It seems to me that at least these things are not there. Does it also mean that one has to control all dependencies by hand?

This is absolutely un-understandable for me!! what does release management tools to do with dependancies? and what do you mean by these things are not there? what are these things?

When you write a class that has to be available in CINT, you have to tell root to create a dictionary and streamers for your class (this what you call xxxLinkDef.h) so here you put the name of your class and weather it has a custom streamer (-) or it should be generated automatically (+) do you need an automatic script to do this?

In the file CMakList.txt you put the names of your implementation and header files and the non-standard include directories which you use, from this information CMake generate the dependancy and Makefiles or project files if you use eclipse, KDevelop or Xcode! So what is the point here?

regards

Mohammad

Subject: Re: user rights for svn and software release tools
Posted by [Bertram Kopf](#) on Thu, 22 Jan 2009 23:04:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Mohammad,

Quote:

Quote:There was an evo meeting before christmas where release management tools was one major topic. As far as I understood it there similar tools like the BaBar srt-tools are (potentially) available in the PandaRoot software. It seems to me that at least these things are not there. Does it also mean that one has to control all dependencies by hand?

This is absolutely un-understandable for me!! what does release management tools to do with dependancies? and what do you mean by these things are not there? what are these things?

When you write a class that has to be available in CINT, you have to tell root to create a dictionary and streamers for your class (this what you call xxxLinkDef.h) so here you put the name of your class and weather it has a custom streamer (-) or it should be generated automatically (+) do you need an automatic script to do this?

In the file CMakeList.txt you put the names of your implementation and header files and the non-standard include directories which you use, from this information CMake generate the dependancy and Makefiles or project files if you use eclipse, KDevelop or Xcode! So what is the point here?

The SRT tools provide among other things few commands which are related to the dependency control. Here is an extraction of the srt manual:

```
...
+) Link Dependency Related commands:
    analyze-linkfiles  [Validate link_XXX.mk against package sources]
    check-link-order  [Validate link_XXX.mk against link_all_reco.mk]
    depend-circle     [Check package for circular dependencies]
    depend-tree       [Generate formatted link dependency tree]
    make-linkfiles    [construct link_XXX.mk file, bin_XXX.mk files]
...
```

The most important command here is "make-linkfiles" which automatically creates the relevant dependency files. That means that the necessary dependency files will be created by just typing this command. So my question was if similar things are available in the PandaRoot environment.

In addition the makefiles are standardized so that one can in principle - apart from some exceptions- use exactly the same makefile in each directory.

Ciao,
Bertram.

Subject: Re: user rights for svn and software release tools
Posted by [Florian Uhlig](#) on Fri, 23 Jan 2009 07:52:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Bertram

I don't know how the "babarians" organized their build process, but if i understand it correctly for each directory the developer had to provide several *.mk files. In the end someone had to use some tools to validate if these files are consistent.

Quote:

```
...
+) Link Dependency Related commands:
    analyze-linkfiles  [Validate link_XXX.mk against package sources]
    check-link-order  [Validate link_XXX.mk against link_all_reco.mk]
    depend-circle     [Check package for circular dependencies]
    depend-tree       [Generate formatted link dependency tree]
    make-linkfiles    [construct link_XXX.mk file, bin_XXX.mk files]
...
```

The most important command here is "make-linkfiles" which automatically creates the relevant dependency files. That means that the necessary dependency files will be created by just typing this command. So my question was if similar things are available in the PandaRoot environment.

In addition the makefiles are standardized so that one can in principle - apart from some exceptions- use exactly the same makefile in each directory.

With CMake everything you need is one file per subdirectory (CMakeLists.txt) where the source/(header) files the include directories and the link directories are defined. Everything else is done automatically. CMake generates takes care of the dependencies, creates the link commands and many things more. In the end CMake generates a complete set of Makefile (or IDE input files if you use KDevelop, Eclipse ...) which are then used to generate the project. In this sense i don't have to validate my input. If there is somewhere a problem i will get an error message. If there are problems to understand the CMakeLists.txt file we can provide an empty one where the user only has to fill the required information.

Hope this helps.

Ciao

Florian

Subject: Re: user rights for svn and software release tools
Posted by [Bertram Kopf](#) on Fri, 23 Jan 2009 15:54:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Florian,
thanks a lot for your mail. This is exactly that what I wanted to know.

Quote:

I don't know how the "babarians" organized their build process, but if i understand it correctly for each directory the developer had to provide several *.mk files. In the end someone had to use some tools to validate if these files are consistent.

Each directory contains only "one" link*.mk file and if an application is defined there it contains just "one" bin*.mk file as well. As I already wrote in the message before these dependency files can be created automatically with the srt tools. In case that the compilation or the link procedure fails there are in addition some dignostic tools available for checking whether there are circular dependencies etc..

Quote:

With CMake everything you need is one file per subdirectory (CMakeLists.txt) where the source/(header) files the include directories and the link directories are defined. Everything else is done automatically. CMake generates takes care of the dependencies, creates the link commands and many things more. In the end CMake generates a complete set of Makefile (or

IDE input files if you use KDevelop, Eclipse ...) which are then used to generate the project. In this sense i don't have to validate my input. If there is somewhere a problem i will get an error message. If there are problems to understand the CMakeLists.txt file we can provide an empty one where the user only has to fill the required information.

O.k.. This means that the developer has to provide / write the "CMakeLists.txt" file and all other things will then be done automatically. As far as I understood it the developer has to maintain this CMakeLists.txt manually. If one adds new classes, includes new files, etc. it is necessary to adjust this file.

Cheers,
Bertram.

Subject: Re: user rights for svn and software release tools
Posted by [Florian Uhlig](#) on Fri, 23 Jan 2009 17:34:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Bertram

Quote:

O.k.. This means that the developer has to provide / write the "CMakeLists.txt" file and all other things will then be done automatically. As far as I understood it the developer has to maintain this CMakeLists.txt manually. If one adds new classes, includes new files, etc. it is necessary to adjust this file.

Correct. But normaly you only have to add new include directories and source files.

Ciao

Florian
