Subject: Closing SVN access on Wednesday 11 February 2009
Posted by Mohammad Al-Turany on Sat, 07 Feb 2009 07:44:03 GMT
View Forum Message <> Reply to Message

Dear colleagues,

On Wednesday 11.02.09 we will close the read and write access to FairRoot/CbmRoot/PandaRoot repository, So that we can rename the base classes and update the Runime Data Base, this should take one or two days, during these two days:

1. All code in the SVN will be adapted to these changes
2. All ASCII Parameter will be also adapted

After these changes, we will lose the backward compatibility to the ROOT files already produce. However old files can still be used with the old releases!

Please synchronize your development code with the trunk before Wednesday. Otherwise you will have to merge the changes yourself!

Regards

Mohammad

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Mohammad Al-Turany on Mon, 16 Feb 2009 12:11:26 GMT
View Forum Message <> Reply to Message

Hallo,

The svn access is opened again, during the last few days we restructure many things beside of the naming changes, the most important changes are:

1. Rename the base classes, this has been also propagated to the user code

2. The mcstack directory have been renamed to pndbase

3. Data classess (hits, points, etc..) has been moved to pndbase except for emc, dch and tpc. these detectors have data classes that includes many files from the detectors classes, something which should be discussed if this is really necessary!

4. I comment out two classes in the dch, because one of them inherits from one of the STT classes! please correct this

5. Beam energy is now implemented in the FairRunSim, choosing an energy will automatically take the proper field map for the dipole etc.

6. New Field maps are in SVN, Dipole and Trans. are beam energy dependent, Solenoid not, there was a bug in the Solenoid maps (Solenoid1-4) it is corrected now! please test it and let

me know! an example of how to use this and the beam energy is in macro/run/run_sim.C and macro/emc/sim_emc.C

6. Detector list is implemented in pndbase.

8. Stack filtering is changed in the PndStack, the user does not see the Bit array anymore, he just have to call:

```
 PndStack* stack = (PndStack*) gMC->GetStack();
stack->AddPoint(DetectorId);
```

9. Stack filtering is set by default to 1, which means particles that do not hit any detector and do not have any daughter that hits any detector will not be saved.

to switch off this filtering call PndStack::SetMinPoints(0); in the g3/4Config.C

10. The Runtime data base has some major changes (From Ilse König). ROOT basic types and arrays are used not, fill/addBinary are replaced by fill/addObject, in the ASCII files, Int_t, Double_t ..etc are used instead of i,d ..etc. These changes are also propagated to the user code and parameters.

That was the most important changes, there was a lot of cleaning and removal of historical stuff also!

regards

Mohammad

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Tobias Stockmanns on Mon, 16 Feb 2009 14:49:08 GMT
View Forum Message <> Reply to Message

Dear Mohammad,

after updating to the newest version of pandaRoot, I get the following cmake error message:

```
make install
--- Found a Linux ssytem
--- Found Intel compiler collection
--- Build Type: Debug
--- Compiler Flags:
-- Looking for Root...
-- Looking for Root... - found /home/stockman/fairroot/cbmsoft/tools/root/bin/root
-- Looking for Root... - version 5.20/00
-- Looking for Pluto...
-- Looking for Pluto... - found /home/stockman/fairroot/cbmsoft/generators/lib
```

-- Looking for Pythia6...
-- Looking for Pythia6... - found /home/stockman/fairroot/cbmsoft/generators/lib
-- Looking for GEANT3...
-- Looking for GEANT3... - found
/home/stockman/fairroot/cbmsoft/transport/geant3/lib/tgt_linux/libgeant3 21.so
-- Looking for GEANT4...
-- Looking for GEANT4... - found
/home/stockman/fairroot/cbmsoft/transport/geant4/lib/Linux-g++
-- Looking for GEANT4VMC...
-- Looking for GEANT4VMC... - found
/home/stockman/fairroot/cbmsoft/transport/geant4_vmc/lib/tgt_linux
-- Looking for VGM...
-- Looking for VGM... - found /home/stockman/fairroot/cbmsoft/transport/vgm/lib
-- Looking for CLHEP...
-- Looking for CLHEP... - found /home/stockman/fairroot/cbmsoft/cern/clhep/lib
-- Looking for Java...
-- Looking for Java... - found /usr/bin/java
-- Looking for Perl...
-- Looking for Perl... - found /usr/bin/perl
-- Looking for Rule Checker...
HIER
-- Looking for Rule Checker... - Not found
CMake Error: Error in cmake code at
/home/stockman/fairroot/cbmsoft/pandaroot/mvd/CMakeLists.txt:104:
SET_TARGET_PROPERTIES called with illegal arguments, maybe missing a PROPERTIES
specifier?
Current CMake stack:
[1] /home/stockman/fairroot/cbmsoft/pandaroot/mvd/CMakeLists.txt
CMake Error: Error in cmake code at
/home/stockman/fairroot/cbmsoft/pandaroot/mvd/CMakeLists.txt:152:
SET_TARGET_PROPERTIES called with illegal arguments, maybe missing a PROPERTIES
specifier?
Current CMake stack:
[1] /home/stockman/fairroot/cbmsoft/pandaroot/mvd/CMakeLists.txt
CMake Error: Error in cmake code at
/home/stockman/fairroot/cbmsoft/pandaroot/mvd/CMakeLists.txt:189:
SET_TARGET_PROPERTIES called with illegal arguments, maybe missing a PROPERTIES
specifier?
Current CMake stack:
[1] /home/stockman/fairroot/cbmsoft/pandaroot/mvd/CMakeLists.txt
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/sim/QAmacro_sim_G3.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/sim/QAmacro_sim_G4.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/dch/QAmacro_dch_1.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/dch/QAmacro_dch_2.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/mdt/QAmacro_mdt_1.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/mdt/QAmacro_mdt_2.sh 2>&1

running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/emc/QAmacro_emc_1.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/emc/QAmacro_emc_2.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/emc/QAmacro_emc_3.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/lhetrack/run_sim_tpccomb i_pgun.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/lhetrack/run_digi_tpccom bi.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/lhetrack/run_reco_tpccom bi.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/tpc/QAmacro_tpc_1.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/drc/QAmacro_drc_1.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/ftof/QAmacro_ftof_1.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/gem/QAmacro_gem_1.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/gpid/QAmacro_gpid_1.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/hyp/QAmacro_hyp_1.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/mvd/QAmacro_mvd_1.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/mvd/QAmacro_mvd_2.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/rho/QAmacro_rho_1.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/rpc/QAmacro_rpc_1.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/stt/QAmacro_stt_1.sh 2>&1
running /bin/chmod u+x
/home/stockman/fairroot/cbmsoft/cbuild/macro/qa/tof/QAmacro_tof_1.sh 2>&1
-- Configuring done
make: *** [cmake_check_build_system] Error 255

Cheers,

Tobias

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Mohammad Al-Turany on Mon, 16 Feb 2009 14:57:00 GMT
View Forum Message <> Reply to Message

Hallo Tobias,

It seems that you have a locally modified CMakeList.txt in the mvd, please look for the CBMROOT in your file and replace it with FAIRROOT if you want to keep your modification, otherwise revert the CMakeList.txt and update the svn again.

Mohammad

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Tobias Stockmanns on Mon, 16 Feb 2009 15:01:16 GMT
View Forum Message <> Reply to Message

Hi Mohammad,

I replaced CBMROOT by FAIRROOT in the CMakeLists.txt file and now it is running.

Nevertheless the file with CBMROOT in was coming from the SVN and is not a locally modified version.

I will upload the fixed file to the SVN.

Cheers,

Tobias

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by StefanoSpataro on Mon, 16 Feb 2009 15:05:16 GMT
View Forum Message <> Reply to Message

In my case it has compiled.

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Tobias Stockmanns on Mon, 16 Feb 2009 15:07:41 GMT
View Forum Message <> Reply to Message

Hi,

Ralf just reminded me that only the trunk was modified. My MVD folder is using the development branch. This is the reason for the error in my case and no error for Stefano.

Cheers,

Tobias

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Mohammad Al-Turany on Mon, 16 Feb 2009 15:18:26 GMT

View Forum Message <> Reply to Message

Hi,

please be careful that I moved the most of the MvdData in the trunk to the pndbase!! also I removed the local detector list, now there is a global detector list in pndbase!

Mohammad

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Mohammad Al-Turany on Mon, 16 Feb 2009 19:06:53 GMT
View Forum Message <> Reply to Message

Hi everybody,

The field maps in SVN are corrupted!! I am trying to reproduce them again, please do not use the new fields, I will replace them as soon as possible!

Mohammad

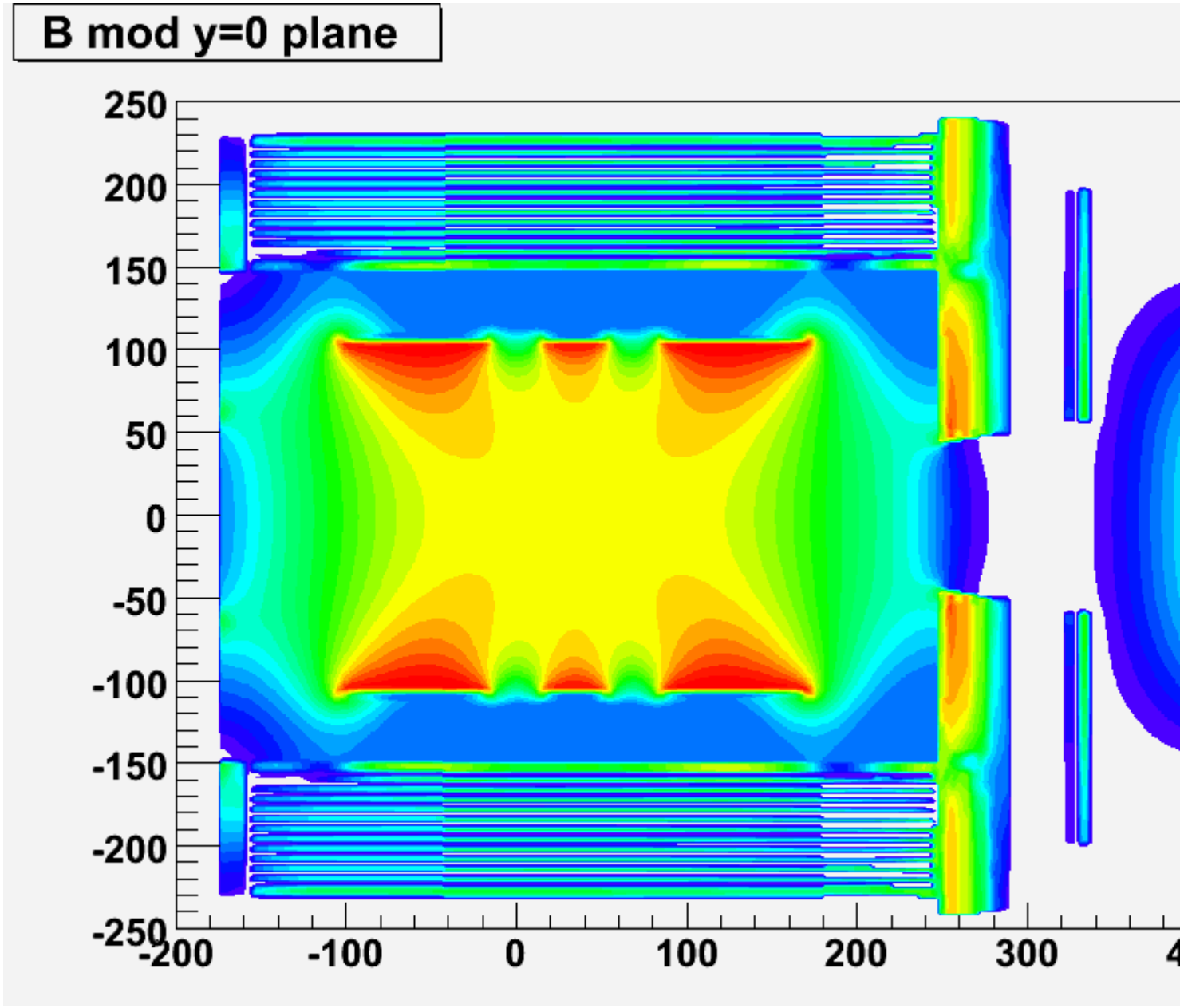
Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Mohammad Al-Turany on Mon, 16 Feb 2009 22:34:18 GMT
View Forum Message <> Reply to Message

Hi,

I corrected the maps now in SVN (r 4648), I also added a macro which can draw the fields (macro/run/DrawField.C)

Mohammad

File Attachments
1) Picture 1.png, downloaded 1595 times



Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Aleksandra Wronska on Tue, 17 Feb 2009 13:00:03 GMT
View Forum Message <> Reply to Message

Dear all,

as for Mohammad's point 4, indeed, one of my classes inherits from one of the stt classes, namely SttRecoHit. I want to use exactly the same code, so I saw no point doing copy&paste. Some time ago I suggested to create a bit more general class in another directory, e.g. in recotasks, such, that both stt and dch can make use of it without being dependent on each other. As this has never been done, either dch package must depend on stt or I duplicate the code of SttRecoHit and WirepointHitPolicy. I also find both solutions unsatisfactory. Any suggestions?

cheers,

ola

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Lia Lavezzi on Tue, 17 Feb 2009 14:27:49 GMT
View Forum Message <> Reply to Message

Hi Ola,
actually the easiest way to proceed is that you create your own PndDchRecoHit without inheriting from the PndSttRecoHit, by copying it and the WirepointHitPolicy files and adapting them to your detector. In fact in the PndSttRecoHit the calculation of the detector plane in PndSttRecoHit::detPlane(AbsRecoHit* hit, AbsTrackRep* rep) contains a check on the distance of the point of closest approach from the firing wire that you may not want to use:

```
// check vpf inside tube
  if(distance>0.5) {
    cout << "vpf outside the firing tube" << endl;
    FitterException exc("distance vpf-wire > 0.5", __LINE__,__FILE__);
    throw exc;
  }
```

in order to throw away the extrapolated hits outside the tube!

In addition to this the WirepointHitPolicy is not general enough to be inserted in a directory such as recotask or better genfit, where the other reco hit policies are, because it is specific for our particular choice of plane...

So, in order to avoid any problems, in my opinion it is better to duplicate these two classes, even if they will be almost the same for STT and DCH.

Ciao,
Lia.

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Bertram Kopf on Tue, 17 Feb 2009 14:47:31 GMT
View Forum Message <> Reply to Message

Hi Ola and all others,

Aleksandra Wronska wrote on Tue, 17 February 2009 14:00
as for Mohammad's point 4, indeed, one of my classes inherits from one of the stt classes, namely SttRecoHit. I want to use exactly the same code, so I saw no point doing copy&paste. Some time ago I suggested to create a bit more general class in another directory, e.g. in recotasks, such, that both stt and dch can make use of it without being dependent on each other. As this has never been done, either dch package must depend on stt or I duplicate the code of SttRecoHit and WirepointHitPolicy. I also find both solutions unsatisfactory. Any suggestions?

I fully agree with you: it would be good to have a more general (abstract) class for this purpose. And this is not only related to the Stt and Dch. I think that also some other classes of the

remaining dectoctor parts need to make use of exactly the same code or of just a few modifications. Therefore it is the best solution to define such a general class in the sense that the relevant detector specific classes inherit from this general object. Another advantage of such a structure is that everything is defined only at "one" place which makes it easier to modify and to maintain the code. In my point of view it would make sense to discuss such a design issue within the tracking core group.

In this context I have another remark to Mohammad's point 3. I took a look into the pndbase directory and I am surprised that this base directory contains only detector specific data classes. I my point of view a base directory should only contain "base classes", i.e. abstract classes or classes without (more or less) any dependencies, which can be used everywhere. An example would be a general RecoHit class as suggested by Ola. The dectector specific data classes should be defined in directories which are at least one layer below.

Best regards,
Bertram.

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Tobias Stockmanns on Tue, 17 Feb 2009 14:55:58 GMT
View Forum Message <> Reply to Message

Hi Lia and Ola,

In general I do not think that duplicating code/classes is really a good idea. Why not creating a common base class from which both detector specific classes inherit from?

This solves the problem of having 98% of the code the same and a 2% specific part for STT and DCH.

Cheers,

Tobias

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Anonymous Poster on Tue, 17 Feb 2009 15:14:14 GMT
View Forum Message <> Reply to Message

Hi,

I agree. The idea of these policies was to share functionality between different detectors. And if one fixes something on the STTpolicy but not on the others... What a mess that'd be.

Cheers, Christian

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Mohammad Al-Turany on Tue, 17 Feb 2009 15:41:10 GMT

View Forum Message <> Reply to Message

Hallo Bertram,

Quote:In this context I have another remark to Mohammad's point 3. I took a look into the pndbase directory and I am surprised that this base directory contains only detector specific data classes. I my point of view a base directory should only contain "base classes", i.e. abstract classes or classes without (more or less) any dependencies, which can be used everywhere. An example would be a general RecoHit class as suggested by Ola. The dectector specific data classes should be defined in directories which are at least one layer below.

pndbase contain only data classes and no functionality, and the abstract classes you mentioned are in base! The idea of pndbase was to collect the whole data classes which should contain no functionality there, may be we should call it panda data or something like this?

Mohammad

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Lia Lavezzi on Tue, 17 Feb 2009 16:03:03 GMT
View Forum Message <> Reply to Message

Hi all,
the WirepointHitPolicy has never been put into genfit as a general class because as it is written it is not so general, in fact it works only with a particular plane orientation in space, i.e. (see also the attached slides from Cracow presentation by Susanna) the plane must have one axis along the wire...
If we assume that all wire detectors use this kind of plane, we can move the WirepointHitPolicy from the stt/sttreco directory to genfit.

Concerning the PndSttRecoHit, the main problem I see is that if we have this reco hit class in common with the dch our changes might affect the dch results (for example, in our last commit we changed the origin of the detector plane, and I' m not sure that this would not cause any problem for the dch...). Would it be so wrong to write two different recohits, which are not connected?

Regards,
Lia.

File Attachments
1) wirehitpolicy.pdf, downloaded 566 times

Subject: Re: Closing SVN access on Wednesday 11 February 2009

Posted by StefanoSpataro on Tue, 17 Feb 2009 16:06:21 GMT
View Forum Message <> Reply to Message

In each case in the SttRecoHit there are even the hit definitions from PndSttHelixHit, and I think dch code does not need those parts (or at least I think no HelixHit are needed by the dch code).

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Bertram Kopf on Tue, 17 Feb 2009 16:21:49 GMT
View Forum Message <> Reply to Message

Hi Lia,

Lia Lavezzi wrote on Tue, 17 February 2009 17:03
Concerning the PndSttRecoHit, the main problem I see is that if we have this reco hit class in common with the dch our changes might affect the dch results (for example, in our last commit we changed the origin of the detector plane, and I' m not sure that this would not cause any problem for the dch...). Would it be so wrong to write two different recohits, which are not connected?

The point is that the PndSttRecoHit should not be defined as a common class. Instead, one can introduce an abstract/common class in addition from which the PndSttRecoHit and the PndDchRecoHit and maybe also some other PndXyzRecoHit classes inherit from. If some functionalities are diffenerent for the concrete classes one can/should make use of virtual functions and overwrite them.

Ciao,
Bertram.

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Bertram Kopf on Tue, 17 Feb 2009 16:57:37 GMT
View Forum Message <> Reply to Message

Hi Mohammad,
thank you for the prompt answer and your explanation about the idea behind this new directory.

Mohammad Al-Turany wrote on Tue, 17 February 2009 16:41
pndbase contain only data classes and no functionality, and the abstract classes you mentioned are in base! The idea of pndbase was to collect the whole data classes which should contain no functionality there, may be we should call it panda data or something like this?

yes, I would also prefere to call it panda data since this is a directory which is just a collection of detector specific data classes.
As far as I can see, the "base" directory collects all base classes which are related to the whole "Fair" software. Therefore my question: Does it make sense to introduce (in the future) also a

"panda base" directory which collects all Panda (and not Fair) related base classes? A potential common PndRecoHit class (which has been also discussed in this thread today) would be one example which could go into this directory. What do you think?

Cheers,
Bertram.

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by StefanoSpataro on Tue, 17 Feb 2009 17:01:26 GMT
View Forum Message <> Reply to Message

Maybe the current "pndbase" could be called simply "pnddata", without misunderstanding (there were already folders called EmcData, MvdData, MdtData, and so on... the nomenclature could be similar to pnddata)

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Lia Lavezzi on Tue, 17 Feb 2009 17:31:17 GMT
View Forum Message <> Reply to Message

Hi Bertram,
I attach here a scheme which explains from which classes the reco hit and hit policy classes inherit (this contains specific references to STT classes).

The abstract class for the reco hit already exists (AbsRecoHit) and then, for the different kinds of hits (i.e. 3D hits, for example for tpc, or planar hits, for example for mvd) the SpacepointHitPolicy and the PlanarHitPolicy specialized the generic hit to the desired type.

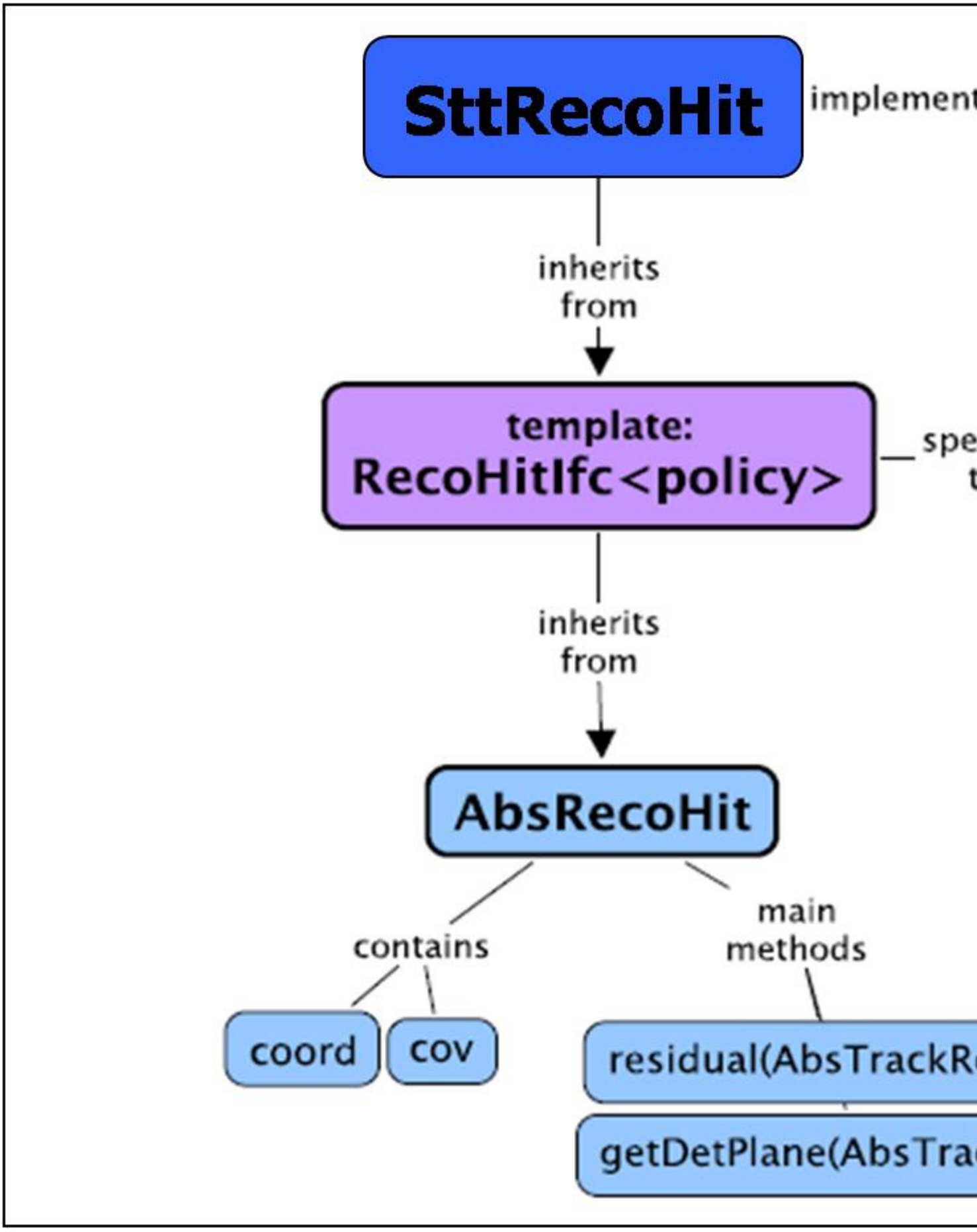
When we inserted the STT, we just added the WirepointHitPolicy in order to handle our particular hit, which is a 3D hit that can' t be described by x,y,z coordinates, but is better described by a wire and the distance from that wire.

So I don' t think we need one more level, but the generic class for the wire detector (to be moved to genfit) could be the WirepointHitPolicy (with the warning already mentioned about the plane orientation).

Regards,
Lia.

File Attachments

1) scheme.jpg, downloaded 1534 times


SttRecoHit
implement
inherits
from
template:
RecoHitIfc<policy>
spe
inherits
from
AbsRecoHit
contains
main
methods
coord
cov
residual(AbsTrackR
getDetPlane(AbsTra

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Mohammad Al-Turany on Tue, 17 Feb 2009 18:36:42 GMT
View Forum Message <> Reply to Message

Hi,

pndbase (libPndBase) is now renamed to pnddata (libPndData).

About AbsRecoHits I don't think it is a good idea to put it in a pndbase, first it is very detector specific and second this will make a PndBase library dependent on GenFit!

For me any general functionality concerning GenFit should be in GenFit itself, the reco hits can also stay in the detector directory but not in a base directory!

Mohammad

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Bertram Kopf on Tue, 17 Feb 2009 19:56:49 GMT
View Forum Message <> Reply to Message

Hi Lia,
thanks for posting the scheme. Sorry for my ignorance and - in my point of view - the scheme looks great. The only thing is that one should avoid that the detector specific hits depend on the SttRecoHit.
Anyhow, I agree with Tobias that one should avoid code duplication. Therefore - I think - it would make sense to introduce additional common classes from which both detector specific classes inherit from.
e.g. a common class WirepointHitPolicy which stays in genfit with concrete classes WirepointSttHitPolicy and WirepointDchHitPolicy which are located in the detector specific directories. A similar structure can be used for the PndXyzRecoHits.
What do you think? Does it make sense?

Ciao,
Bertram.

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Tobias Stockmanns on Wed, 18 Feb 2009 09:25:51 GMT
View Forum Message <> Reply to Message

Hi Mohammad,

libPNDDATA is missing in the rootlogon() script.

Can you fix this?

Cheers,

Tobias

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Mohammad Al-Turany on Wed, 18 Feb 2009 10:04:10 GMT
View Forum Message <> Reply to Message

Hallo Tobias,

I just put it there.

Mohammad

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Lia Lavezzi on Wed, 18 Feb 2009 15:37:41 GMT
View Forum Message <> Reply to Message

Hi Bertram and hi all,
I tried moving the WirepointHitPolicy from stt/sttreco to genfit. By doing this, the policy class would be in common between the stt and the dch.
Concerning the reco hits, two separate reco hits, specific for stt and dch, should be implemented. I understand the point about avoiding code duplication, and I know that there could still be a "problem" about the two detPlane(AbsRecoHit* hit, AbsTrackRep* rep) functions. They will be similar, but actually they will not be exactly the same, because in stt we have the 0.5 cm check (not needed in dch) and the origin of the frame is placed in the center of the wire (I' m not sure about the dch choice).
In any case, I cannot imagine how to move this to the WirepointHitPolicy (and so in genfit) since as far as I know the genfit package should be independent from geane and in the detPlane(...) function we use it.
Moreover this function is explicitly implemented for GeaneTrackRep, because in the stt case we did not use LSLTrackRep, and this adds a dependence of genfit on the trackrep package...

Do you think this could be a solution?

Regards,
Lia.

Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Lia Lavezzi on Thu, 19 Feb 2009 17:14:06 GMT
View Forum Message <> Reply to Message

Hi,
since it seems that no particular objection to the last proposed solution arised, I moved the WirepointHitPolicy to genfit, in order to make it usable also by dch detector without unwanted cross dependences.

I tested it with the stt and it works; please, Ola tell me if you find any problem with the dch.

Best regards,
Lia.