Subject: Abandonware
Posted by StefanoSpataro on Fri, 17 Aug 2007 10:37:23 GMT
View Forum Message <> Reply to Message

I am wondering if it is the case to remove from the PandaRoot root directory all those packages which are now obsolete.

These are, if I have understood well:

bfield (obsolete, now field is the official one)
stt1 (obsolete, now stt is the official one)
stt2 (obsolete, now stt is the official one)
fsc (it is now included inside emc)

I think it is meaningless to continue to download and compile code that we do not use anymore.
Maybe it could be the case to create an "archive" directory that is not downloaded from svn, where we can put all these obsolete packages.

Comments and ideas are welcome

Ste

Subject: Re: Abandonware
Posted by Aleksandra Wronska on Fri, 17 Aug 2007 10:52:42 GMT
View Forum Message <> Reply to Message

No objections from my side concerning fsc.

ola

Subject: Re: Abandonware
Posted by Pablo Genova on Tue, 21 Aug 2007 10:23:18 GMT
View Forum Message <> Reply to Message

I perfectly agree with you, Pablo

Subject: Re: Abandonware
Posted by StefanoSpataro on Wed, 22 Aug 2007 15:49:22 GMT
View Forum Message <> Reply to Message

Dear all,
I removed from the main makefile the libraries stt1 stt2 fsc and bfield, and I added stt (that was missing before) and tof.

Even the Dart.sh file should be updated.

I will keep them for another week, then I will delete them.

Stt package was never compiled before under Dashboard, so all the deprecated warings are still there.

There are 19 warnings still existing, could you please remove them? I could do it, if you agree, but there is one warning (the last one) that I could not understand.

Thanks

Subject: Re: Abandonware
Posted by Pablo Genova on Thu, 23 Aug 2007 14:00:37 GMT
View Forum Message <> Reply to Message

Hi Stefano,

I fixed the trivial warnings due to the deprecated headers and committed to svn. Now you should see only 2 warnings, which I do not know how to fix.

One is
/data10/pablo/tmp/pandaroot/stt/CbmSttTrackFinderIdeal.cxx: In member function
`virtual Int_t CbmSttTrackFinderIdeal::DoFind(TClonesArray*)':
/data10/pablo/tmp/pandaroot/stt/CbmSttTrackFinderIdeal.cxx:202: warning: passing
`Double_t' for argument 1 of `_Tp& std::map<_Key, _Tp, _Compare,
_Alloc>::operator[](const _Key&) [with _Key = Int_t, _Tp = Int_t, _Compare =
std::less<Int_t>, _Alloc = std::allocator<std::pair<const Int_t, Int_t> >]'

and the other is

/data10/pablo/tmp/pandaroot/stt/CbmSttHelixTrackFitter.cxx: In member function
`virtual Int_t CbmSttHelixTrackFitter::DoFit(CbmSttTrack*, int)':
/data10/pablo/tmp/pandaroot/stt/CbmSttHelixTrackFitter.cxx:168: warning: NULL used in
arithmetic

due to a control like (function() !=NULL).

I do not know how to fix them, I'm surprised of the latter because it is quite common that one checks if a pointer is not null, so I do not know how to keep the test but eliminate the warning.

ciao, Pablo

Subject: Re: Abandonware

Posted by StefanoSpataro on Thu, 23 Aug 2007 14:47:12 GMT
View Forum Message <> Reply to Message

Hi,
I will check the first one.

For the second one I think the problem is that you are not comparing a pointer, but a number.

Indeed in the expression:

(pTrack->GetParamLast()->GetTx()) != NULL

pTrack is a pointer to a CbmSttTrack,
pTrack->GetParamLast() is a pointer to a CbmTrackParam,
but pTrack->GetParamLast()->GetTx() is a double so a number! Maybe you should remove GetTx or something else, to clean that condition.

P.S. I removed the headers in tof code

Subject: Re: Abandonware
Posted by Pablo Genova on Thu, 23 Aug 2007 15:02:19 GMT
View Forum Message <> Reply to Message

Stefano,

you are perfectly right! That check was completely meaningless . Now it has been properly modified ((..)>0).

Thank you very much for the remark.

Pablo

Subject: Re: Abandonware
Posted by StefanoSpataro on Thu, 23 Aug 2007 15:04:01 GMT
View Forum Message <> Reply to Message

About the first warning, this is not so clear to me.

You define hitMap as:

map<Int_t, map<Int_t, Int_t> > hitMap;

Then:

```
Double_t wireX = pMhit->GetX(), wireY = pMhit->GetY();
    (hitMap[mcTrackIndex])[wireX * wireX + wireY * wireY]++;
```

The map is <Int_t, Int_t> but then you give one index as [double].
I think you can (one of the two)
a) correct the map putting one double inside (map<Int_t, Double_t>)
b) overcast the wire id (Int_t)(wireX * wireX + wireY * wireY)

Just at a first sight the first should be the correct one, in each case I am wondering how does it work with this int/double ambiguity...

Subject: Re: Abandonware
Posted by Pablo Genova on Mon, 27 Aug 2007 11:07:48 GMT
View Forum Message <> Reply to Message

the map has been corrected according to method a),
the plots of the standard reconstruction macros are OK.

ciao, Pablo

Subject: Re: Abandonware
Posted by StefanoSpataro on Mon, 27 Aug 2007 11:40:29 GMT
View Forum Message <> Reply to Message

Ladies and Gentlemen,
I am proud to announce that for the first time in the PandaRoot life we have absolutely NO WARNING at all!!

Let's hope to keep this profile as long as possible ...

Bye

Subject: Removed obsolete packages
Posted by StefanoSpataro on Thu, 13 Sep 2007 12:24:40 GMT
View Forum Message <> Reply to Message

Hello,

the obsolete packages since revision 1496 are removed from the repository.

If somebody wants to retrieve the old code, he should call the revision 1495 (or below).

Regards
