Migration of CMSSW software from 3_1_1 version to 4_2_4

Currently (r5094) we have in repository two parallel version of the code:

- **trunk** (svn+ssh://svn.cern.ch/reps/totem/trunk/offline/cmssw/src) containing software compatible with CMSSW 3_1_1
- **migration 4_2_4 branch** (svn+ssh://svn.cern.ch/reps/totem/branches/CMSSW_4_2_4/offline/cmssw/src) containing software compatible with CMSSW 4_2_4 (corresponds to r5094 of trunk)

On Monday 12.09.2011 there will be significant change and trunk will contain version of software compatible with CMSSW 4_2_4 and repository will be organized as following:

- **trunk** (svn+ssh://svn.cern.ch/reps/totem/trunk/offline/cmssw/src) containing software compatible with CMSSW 4_2_4
- **backward compatibility 3_1_1 branch** (svn+ssh://svn.cern.ch/reps/totem/branches/CMSSW_3_1_1/cmssw/src) containing software compatible with CMSSW 3_1_1 (corresponds to r5094 of trunk)

Detailed instructions how to use and compile code in both versions could be found in following places:

- short set of commands how to compile and run code in all versions
- detailed getting started for 3_1_1 version
- detailed getting started for 4_2_4 version
Important actions to take (after Monday 12.09.2011).

- **do not commit** from workspaces create before 12.09.2011. Let us assume that you did `svn checkout ...
  trunk` and have it in your directory. Now trunk contains completely different files (compatible with 4_2_4). `svn commit` could in such result with error or broken files in repository.
- **safe option to still work with 3_1_1** (only if you plan to develop and commit to repository). Recompile workspace from scratch, fetching code from different place in repository, according to instruction.
- **less safe option to still work with 3_1_1** (only if you plan to develop and commit to repository). Remove `src/` directory from workspace, fetch new one from backward compatibility 3_1_1 branch (svn+ssh://svn.cern.ch/reps/totem/branches/CMSSW_3_1_1/cmssw/src)
- doing `svn update` on workspaces create before 12.09.2011 will overwrite your code with one compatible with 4_2_4 making your workspace not compiling.
- if you have some configuration files that are not in SVN repository and want to use them with 4_2_4 framework, they need to be adjusted according to list of changes below
Main differences:

- code works only on 64bit SLC5 machines (lxplus, lxbatch, pctotem31)
- different path to cmsset_default.sh file (now /afs/cern.ch/cms/cmsset_default.sh, before /afs/cern.ch/cms/sw/cmsset_default.sh). You need to change bash scripts used to submit jobs to cluster or others where you use this path.
- no need to prepare special version of Geant4
- some MC guns have different type (EDProducer instead of InputSource). You need to change cfg files used for simulation (use empty source and put generator at the beginning of the path), see example configuration in Configuration/TotemStandardSequences/test/. Changed MC guns:
  - DPEProtons
  - FlatProtonLogKsiLogTGun
  - FlatProtonLogKsiLogTGunMul
  - MultFlatGunSourceEta EMC2
  - MultFlatGunSource
- obsolete modules from RecoTotemT1T2 removed. You need to change cfg files used for raw data reconstruction and simulation. Replace T2MakeCluster3 with T2MakeCluster, and others as following:
  - T2MakeCluster and T2MakeCluster2 removed; T2MakeCluster3 renamed as T2MakeCluster
  - T2RecHit removed; T2RecHit2 renamed as T2RecHit
  - T2RoadProducer removed; T2RoadProducer2 renamed as T2RoadProducer
  - T2TrackProducer removed
- obsolete plugins from TotemRawData/RawToDigi removed (all these plugins should be replaced with Raw2DigiProducer as this plugin covers functionality of all obsolete plugins):
  - RPDataCCProducer
  - RPDataDigiProducer
  - T1XMLDataDigiProducer
  - T2DataDigiProducer
  - T2XMLDataDigiProducer
- TotemDQM package will not compile by default, you need to configure qtROOT package and enable compilation in BuildFile.xml files of TotemDQM package.
- important when adding new package: configuration of SCRAM is now different. CMSSW_4_2_4 is using BuildFile.xml instead of BuildFile. See how does it look like in migrated packages. There is automatic converter to migrate between BuildFile and BuildFile.xml
- important when adding new class: in CMSSW 3_1_1 there was a method virtual void beginJob(const edm::EventSetup&), now it doesn't take any parameter and is virtual void beginJob();. If you add put it in the code in the old way, method will not be called but you will also not see any compiler error. To use EventSetup at the beginning of the processing, use virtual void beginRun(const edm::EventSetup&).