# Table of Contents

How to compile entire framework agains a different version of ROOT

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>How to compile entire framework against a different version of ROOT</td>
<td>1</td>
</tr>
<tr>
<td>Do you really need it?</td>
<td>1</td>
</tr>
<tr>
<td>Compiling CMSSW_3_1_x with ROOT 5.24.00</td>
<td>1</td>
</tr>
<tr>
<td>Step 0</td>
<td>1</td>
</tr>
<tr>
<td>Step 1</td>
<td>1</td>
</tr>
<tr>
<td>Step 2</td>
<td>1</td>
</tr>
<tr>
<td>Step 3</td>
<td>1</td>
</tr>
<tr>
<td>Step 4</td>
<td>1</td>
</tr>
<tr>
<td>Step 5: rebuild POOL</td>
<td>2</td>
</tr>
<tr>
<td>Step 6: Recompile everything</td>
<td>2</td>
</tr>
<tr>
<td>Step 7: Solve problems</td>
<td>2</td>
</tr>
</tbody>
</table>
How to compile entire framework against a different version of ROOT

Do you really need it?

Very probably not - this is a rather emergency case. An example where it was really needed. The CMSSW_3_1_x uses ROOT 5.22.x. This ROOT version has a bug in the Qt interface. This bug has only been corrected in version 5.24.

Compiling CMSSW_3_1_x with ROOT 5.24.00

Step 0

Ask Leszek to install this CMSSW version to the standard location (/afs/cern.ch/exp/totem/scratch/Release) :-)\)

Step 1

In /afs/cern.ch/exp/totem/scratch/Release/cmssw/slc4_ia32_gcc345/lcg/root do

ln -s /afs/cern.ch/sw/lcg/app/releases(ROOT/5.24.00/slc4_ia32_gcc34_dbg/root/ 5.24.00-totem

Step 2

Save the following script

```bash
export VO_CMS_SW_DIR=/afs/cern.ch/exp/totem/scratch/Release/cmssw/
export SCRAM_ARCH=slc4_ia32_gcc345
export LANG=C
source $VO_CMS_SW_DIR/cmsset_default.sh
# this is in fact $CMSSW_RELEASE_BASE/src
cd /afs/cern.ch/exp/totem/scratch/Release/cmssw/slc4_ia32_gcc345/cms/cmssw/CMSSW_3_1_1/src
eval `scramv1 runtime -sh`

and run it.
```

Step 3

Go to

/aafs/cern.ch/exp/totem/scratch/Release/cmssw/slc4_ia32_gcc345/cms/cmssw/CMSSW_3_1_1/config/toolbox

and edit all *root* files.

1. Replace version with 5.24.00-totem.
2. in rootrflx and rootcore change the path accordingly
3. in xrootd modify the INCLUDE line like:

   `<environment name="INCLUDE" default="/afs/cern.ch/sw/lcg/app/releases/ROOT/5.24.00/src/root/net/xrootd/src/xrootd/src"/>

Step 4

Run the following commands in bash:
scram tool list | grep root | while read tool; do name=${tool%% *}; scram setup $name; done

**Step 5: rebuild POOL**

Go to [Release/cmssw/slc4_ia32_gcc345/cms/pool/POOL_2_9_0_pre5-cms9](http://Release/cmssw/slc4_ia32_gcc345/cms/pool/POOL_2_9_0_pre5-cms9/) and

- copy `*root*` tool config files from CMSSW to POOL
- repeat step 4 to load the changes
- rebuild POOL: scram b distclean; scram b

**Step 6: Recompile everything**

scram b distclean
scram b

**Step 7: Solve problems**

- In `DataFormats/Math/src/classes.h` comment out lines like `ROOT::Math::SymMatrixOffsets symo2`. The `SymMatrixOffsets` is not a part of the official ROOT release.

- In `SimDataFormats/CrossingFrame/interface/CrossingFrame.h` add lines

  ```
  template <class T> const int CrossingFrame<T>::lowTrackTof = 0;
  template <class T> const int CrossingFrame<T>::highTrackTof = 0;
  template <class T> const int CrossingFrame<T>::minLowTof = 0;
  template <class T> const int CrossingFrame<T>::limHighLowTof = 0;
  ```

- In `DataFormats/Candidate/src/Candidate.cc` add lines

  ```
  const unsigned int Candidate::longLivedTag = 0;
  ```

- In `DataFormats/ParticleFlowReco/src/PFBlockElement` add `const unsigned
  PFBlockElement::nTypes_ = 0;`

- In `PhysicsTools/MVAComputer/src/ProcTMVA.cc` replace

  ```
  #include <TMVA/Methods.h>
  ```

  by

  ```
  #include "TMVA/IMethod.h"
  #include "TMVA/MethodANNBase.h"
  #include "TMVA/MethodBDT.h"
  #include "TMVA/MethodBase.h"
  #include "TMVA/MethodBayesClassifier.h"
  #include "TMVA/MethodBoost.h"
  #include "TMVA/MethodCFMlpANN.h"
  #include "TMVA/MethodCFMlpANN_Utils.h"
  #include "TMVA/MethodCFMlpANN_def.h"
  #include "TMVA/MethodCommittee.h"
  #include "TMVA/MethodCompositeBase.h"
  #include "TMVA/MethodCuts.h"
  #include "TMVA/MethodDT.h"
  #include "TMVA/MethodFDA.h"
  #include "TMVA/MethodFisher.h"
  #include "TMVA/MethodHMatrix.h"
  #include "TMVA/MethodKNN.h"
  #include "TMVA/MethodLD.h"
  #include "TMVA/MethodLikelihood.h"
  #include "TMVA/MethodMLP.h"
  #include "TMVA/MethodPDEFoam.h"
  #include "TMVA/MethodPDERes.h"
  #include "TMVA/MethodRuleFit.h"
  #include "TMVA/MethodSVM.h"
  ```
```c
#include "TMVA/MethodSeedDistance.h"
#include "TMVA/MethodTMlpANN.h"

... and probably many others ... I gave up ...

-- JanKaspar - 28 Jul 2009
```