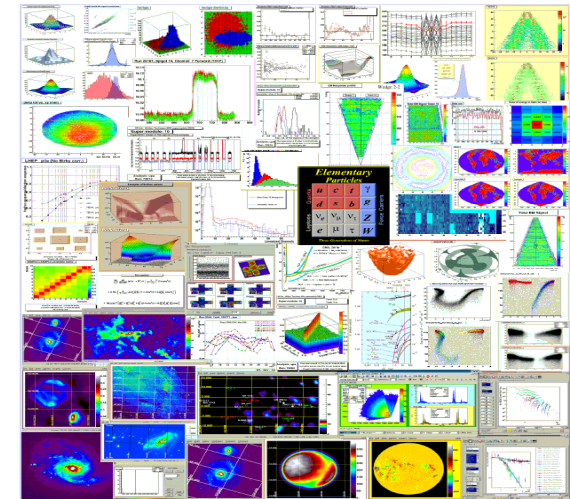


Global Overview of the Current ROOT System

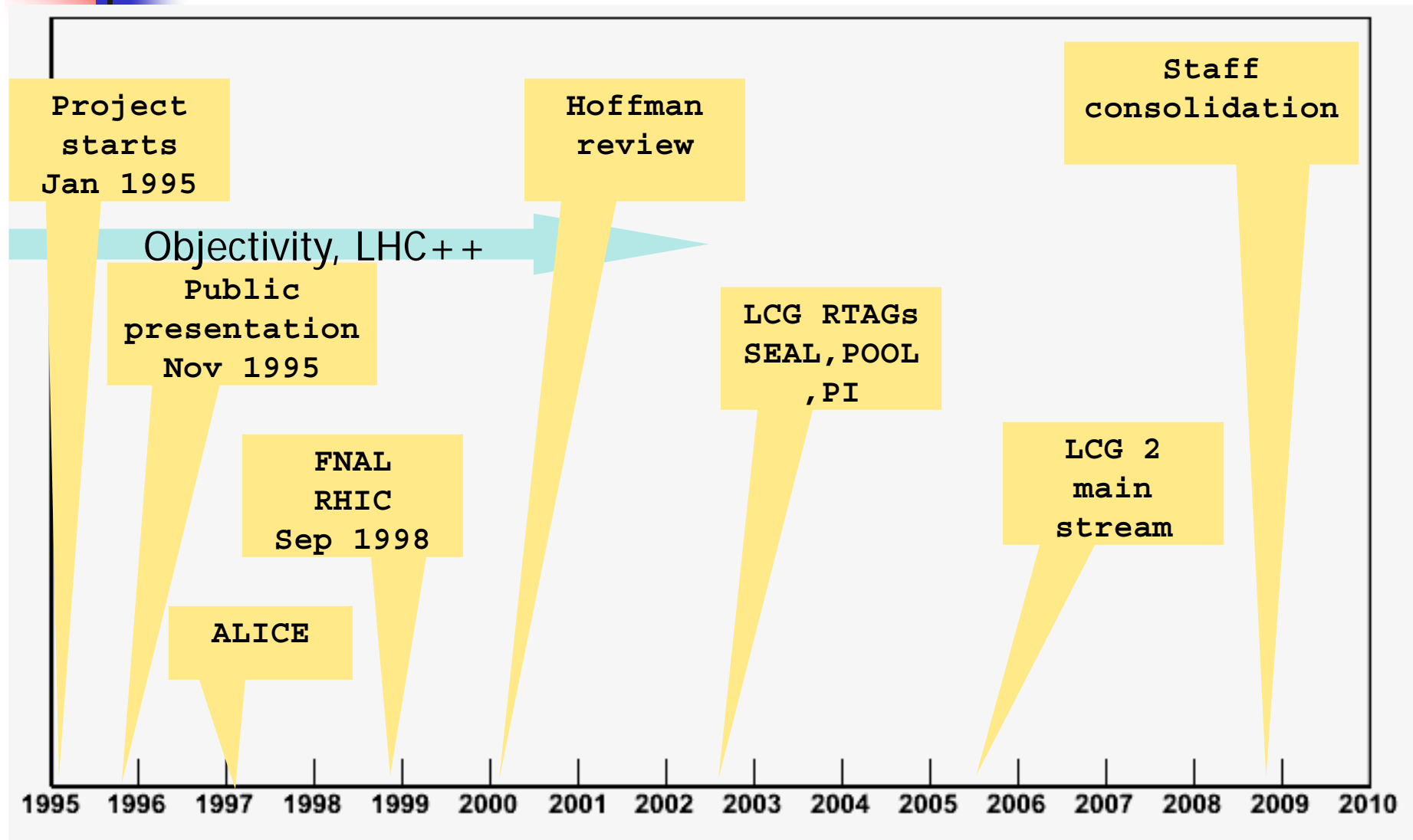
CHEP09 : Prague, 24 March 2009

René Brun/CERN
for the ROOT team



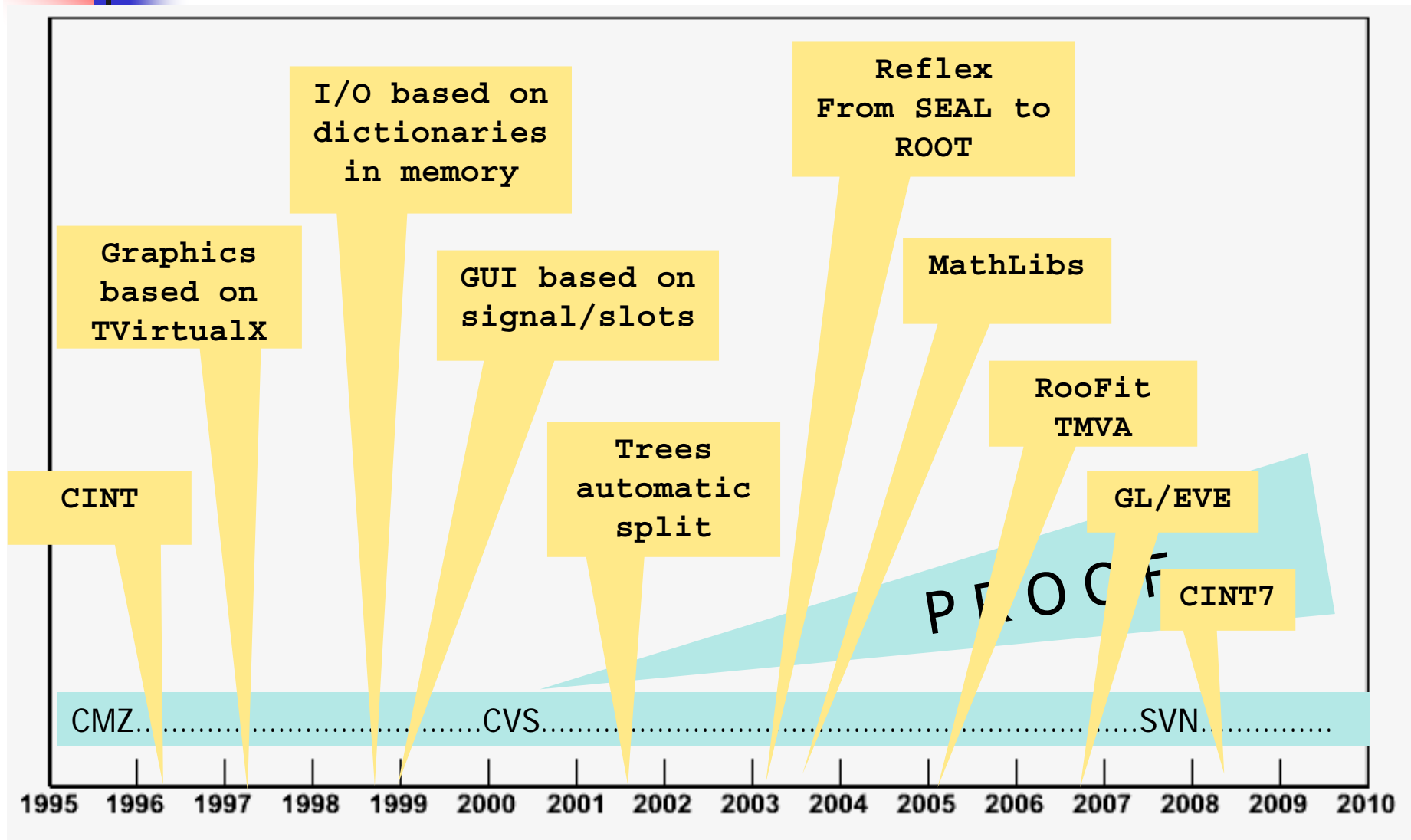


Project History : Politics





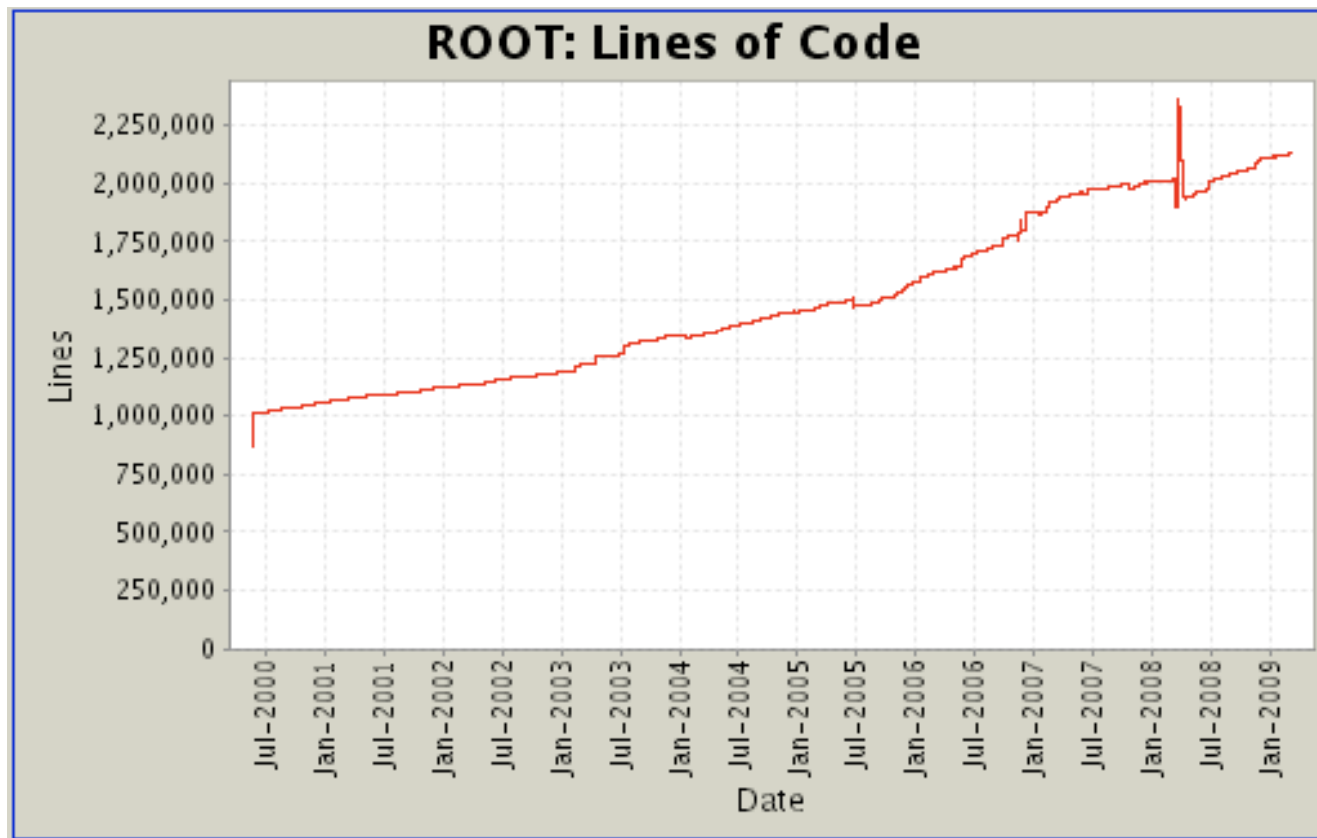
Major Technical Steps





From CVS to SVN

- Smooth and fast transition from CVS to SVN
- We are very happy with SVN





SVN Structure

20 top packages

- root
 - README
 - bin
 - bindings
 - build
 - cint
 - config
 - core
 - doc
 - etc
 - fonts
 - geom
 - graf2d
 - graf3d
 - gui
 - hist
 - html
 - icons
 - include
 - io
 - lib
 - macros
 - main
 - man
 - math
 - misc
 - montecarlo
 - net
 - proof
 - roofit
 - rootx
 - sql
 - test
 - tmva
 - tree
 - tutorials

- graf2d
 - asimage
 - doc
 - freetype
 - gpad
 - doc
 - index.txt
 - inc
 - src
 - Module.mk

root/graf2d/gpad/doc/index.txt





ROOT libs

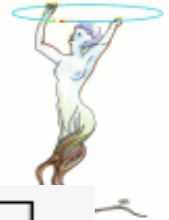
- Granularity: More than 100 shared libs
- You load what you use
- **root.exe** links 6 shared libs (VM < 20 Mbytes)

```
(macbrun2-3) [357] ls *.so
libASImage.so      libGenVector.so    libMetaTCint.so    libRecorder.so     libTree.so         libXrdSecpwd.so
libASImageGui.so   libGeom.so         libMetaTCint_7.so  libReflex.so       libTreePlayer.so   libXrdSecsss.so
X libCint.so       libGeomBuilder.so  libMinuit.so       libReflexDict.so   libTreeViewer.so   libXrdSecunix.so
libCint7.so        libGeomPainter.so  libMinuit2.so      X libRint.so        libUnuran.so       libXrdSut.so
libCintex.so       libGpad.so         libNet.so          libRooFit.so       libVMC.so          libdequeDict.so
X libCore.so       libGraf.so         libNetx.so         libRooFitCore.so   libX3d.so          liblistDict.so
libEG.so          libGraf3d.so       libNew.so          libRooStats.so     libXMLIO.so        libmap2Dict.so
libEGPythia6.so    libGui.so          libPhysics.so      libRootAuth.so     libXMLParser.so    libmapDict.so
libEGPythia8.so    libGuiBld.so       libPostscript.so   libRuby.so         libXrdBwm.so       libminicern.so
libEve.so          libGuiHtml.so      libProof.so        libSPlot.so        libXrdClient.so    libmultimap2Dict.so
libFFTW.so         libHbook.so        libProofDraw.so    libSQL.so          libXrdCrypto.so     libmultimapDict.so
libFTGL.so         X libHist.so       libProofPlayer.so  libSessionViewer.so libXrdCryptossl.so  libmultisetDict.so
libFitPanel.so     libHistPainter.so  libProofx.so       libSmatrix.so      libXrdOfs.so       libsetDict.so
libFoam.so         libHtml.so         libPyROOT.so       libSpectrum.so     libXrdProofd.so    libvalarrayDict.so
libFumili.so       libKrb5Auth.so     libQuadp.so        libSpectrumPainter.so libXrdRootd.so     libvectorDict.so
libGX11.so         libMLP.so          libRGL.so          libSrvAuth.so      libXrdSec.so
libGX11TTF.so      X libMathCore.so   libRIO.so          libTMVA.so         libXrdSecgsi.so
libGdml.so         libMathMore.so     libRLDAP.so        libTable.so        libXrdSecgsiGMAPLDAP.so
libGed.so          X libMatrix.so     libRODBC.so        libThread.so       libXrdSeckrb5.so
```

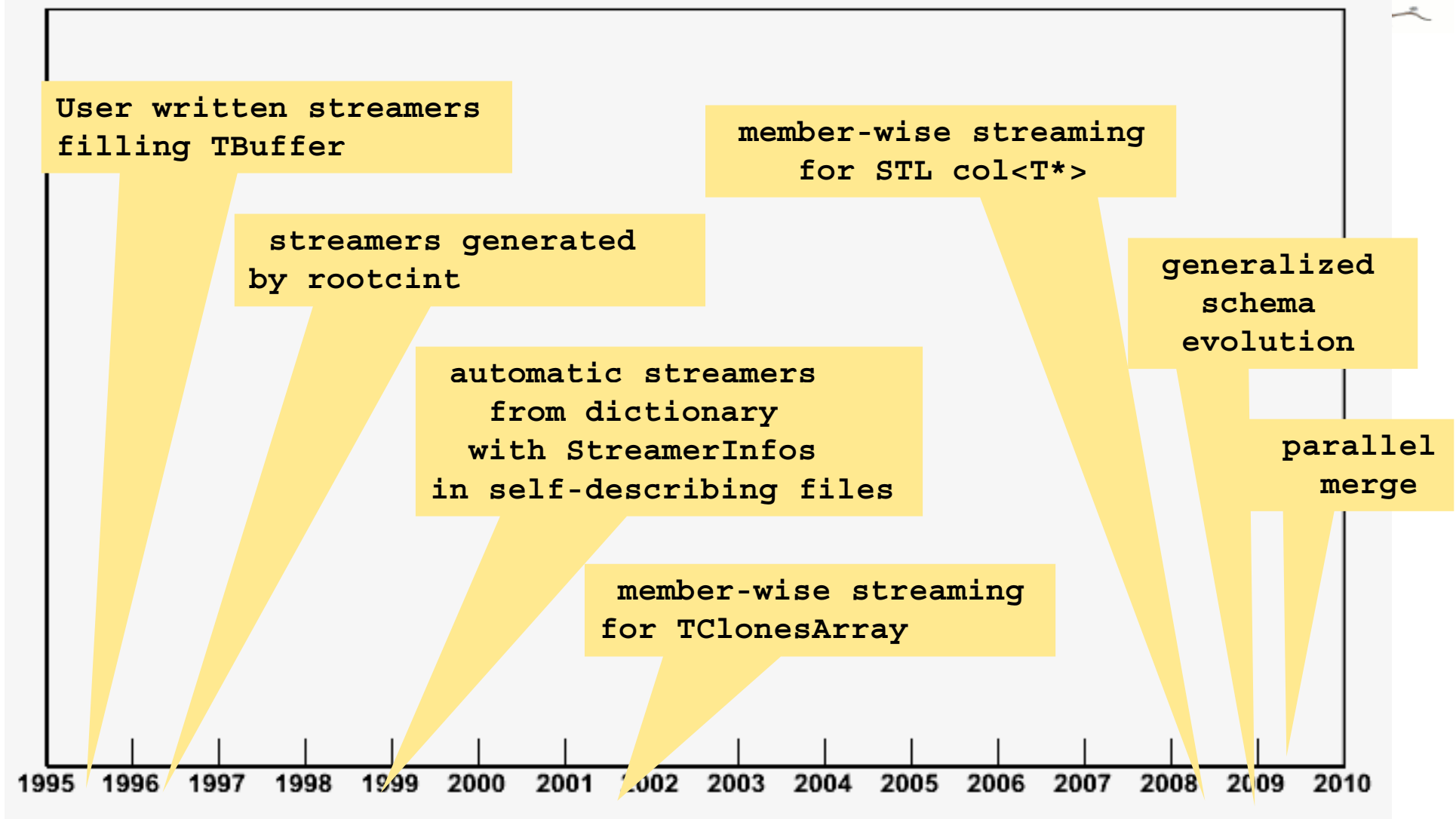


CINT → LLVM

- CINT is the CORE of ROOT for
 - Parsing code
 - Interpreting code
 - Storing the classes descriptions
- A new version of CINT (**CINT7**) is based on Reflex, but found too slow to go to production.
- We are considering an upgrade of CINT using **LLVM** (Apple-driven OS project). LLVM is a GCC compliant compiler with a parser and a **Just In Time** Compiler. (see talk by Axel Naumann)
- CINT/LLVM (**CLING**) should be **c++0x** compliant



Input/Output: Major Steps





Files: Local/Remote

- Local files
 - From 32 to 64 bits pointers
 - What about Lustre/ZFS
 - `TFile::Open("mylocalfile.root")`
- Web files on a remote site using a standard **Apache web server**
 - `TFile::Open("http://myserver.xx.yy/file.root")`
- Remote files served by **xrootd**
 - Support for parallelism
 - Support for intelligent read-ahead (via **TTreeCache**)
 - Support for multi-threading
 - At the heart of **PROOF**
 - `TFile::Open("root://myxrootdserver.xx.yy/file.root")`



Files in XML



```
Canvas.Print("x.C")  
Canvas.Print("x.root")  
Canvas.Print("x.xml")
```

ROOT/XML files
A good tool to
understand ROOT
I/O

Rene Brun

```
<?xml version="1.0"?>  
<root setup="2xoo" ref="null" created="2009-03-13 09:20:31" modified="2009-03-13 09:20:31" uuid="  
<XmlKey name="c1" cycles="1" created="2009-03-13 09:20:31">  
  <Object class="TCanvas">  
    <Version v="6"/>  
    <TPad version="10">  
      <TVirtualPad version="2">  
        <TObject fUniqueID="0" fBits="3030008"/>  
        <TAttLine version="1">  
          <fLineColor v="1"/>  
          <fLineStyle v="1"/>  
          <fLineWidth v="1"/>  
        </TAttLine>  
        <TAttFill version="1">  
          <fFillColor v="19"/>  
          <fFillStyle v="1001"/>  
        </TAttFill>  
        <TAttPad version="3">  
          <fLeftMargin v="0.100000"/>  
          <fRightMargin v="0.100000"/>  
          <fBottomMargin v="0.100000"/>  
          <fTopMargin v="0.100000"/>  
          <fXfile v="2.000000"/>  
          <fYfile v="2.000000"/>  
          <fAfile v="1.000000"/>  
          <fXstat v="0.990000"/>  
          <fYstat v="0.990000"/>  
          <fAstat v="2.000000"/>  
          <fFrameFillColor v="19"/>  
          <fFrameLineColor v="1"/>  
          <fFrameFillStyle v="1001"/>  
          <fFrameLineStyle v="1"/>  
          <fFrameLineWidth v="1"/>  
          <fFrameBorderSize v="1"/>  
          <fFrameBorderMode v="1"/>  
        </TAttPad>  
        <TQObject/>  
      </TVirtualPad>  
      <fX1 v="1993.000000"/>  
      <fY1 v="-1.246875"/>  
      <fX2 v="2013.000000"/>  
      <fY2 v="11.221875"/>  
      <fXtoAbsPixelk v="-79321.398762"/>  
      <fXtoPixelk v="-79321.398762"/>  
      <fXtoPixel v="39.799999"/>  
      <fYtoAbsPixelk v="514.800047"/>  
      <fYtoPixelk v="514.800047"/>  
      <fYtoPixel v="-45.874686"/>  
      <fUtoAbsPixelk v="0.000050"/>  
      <fUtoPixelk v="0.000050"/>  
      <fUtoPixel v="796.000000"/>  
      <fVtoAbsPixelk v="572.000050"/>  
      <fVtoPixelk v="572.000000"/>
```



I/O and Trees

- from branches of basic types created by hand
- to branches automatically generated from very complex objects
- to branches automatically generated for complex polymorphic objects
- Support for object weak-references across branches (TRef) with load on demand
- Tree Friends
- TEntryList

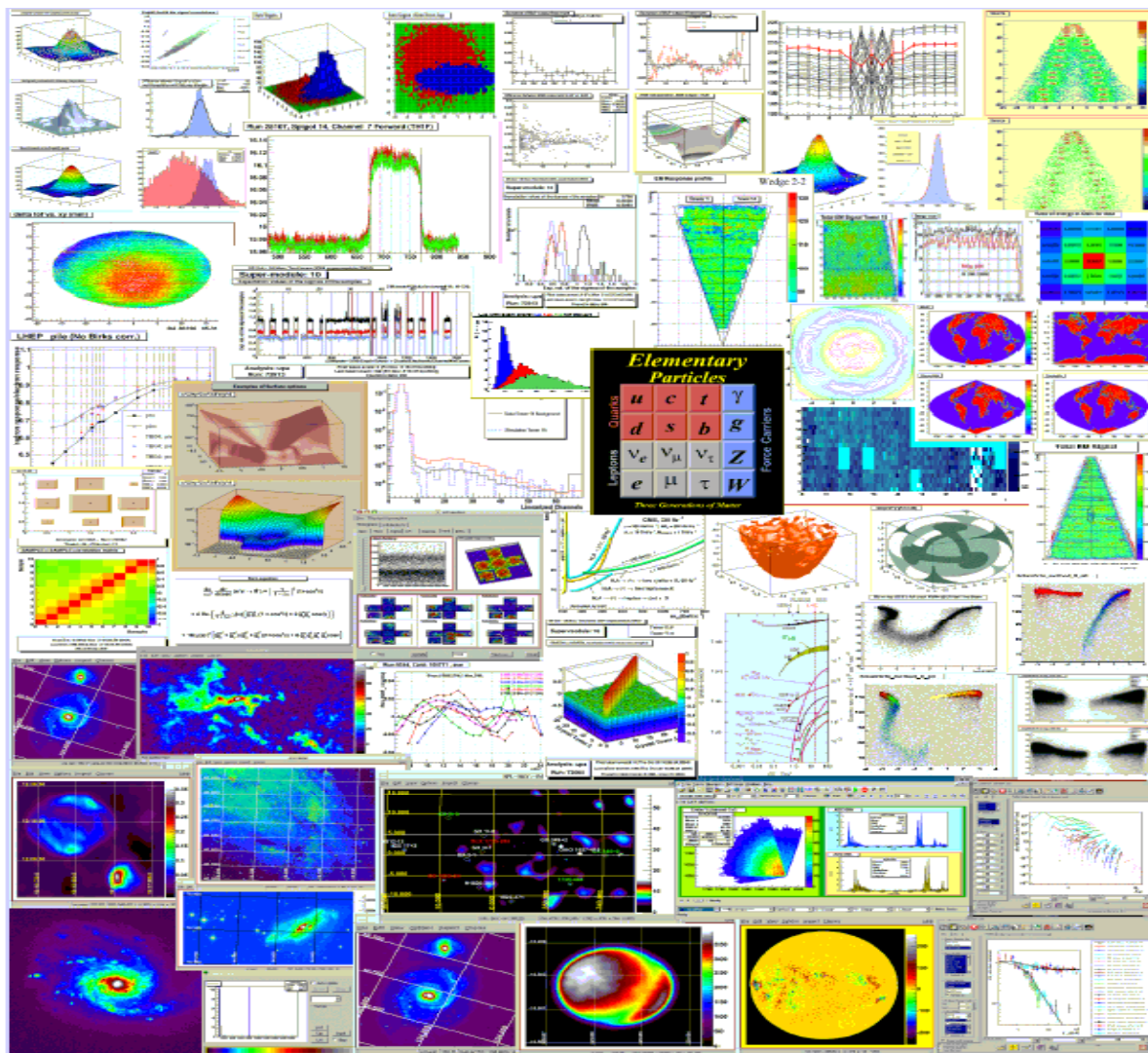


2-D Graphics



- New functions added at each new release.
- Always new requests for new styles, coordinate systems.
- ps, pdf, svg, gif, jpg, png, c, root, etc

Move to
GL ?

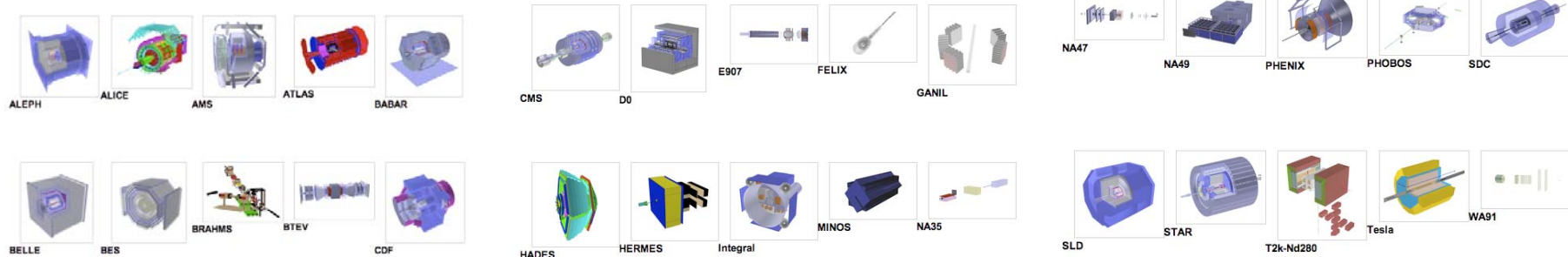




The Geometry Package TGeo



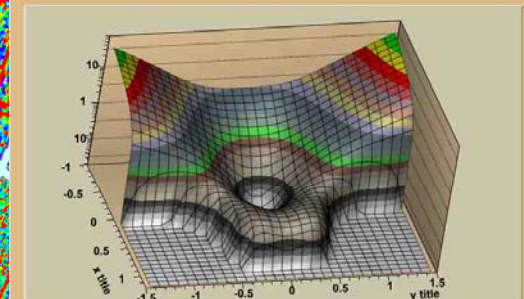
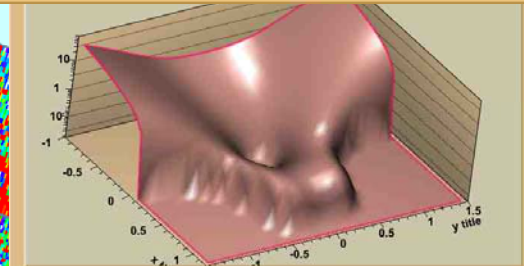
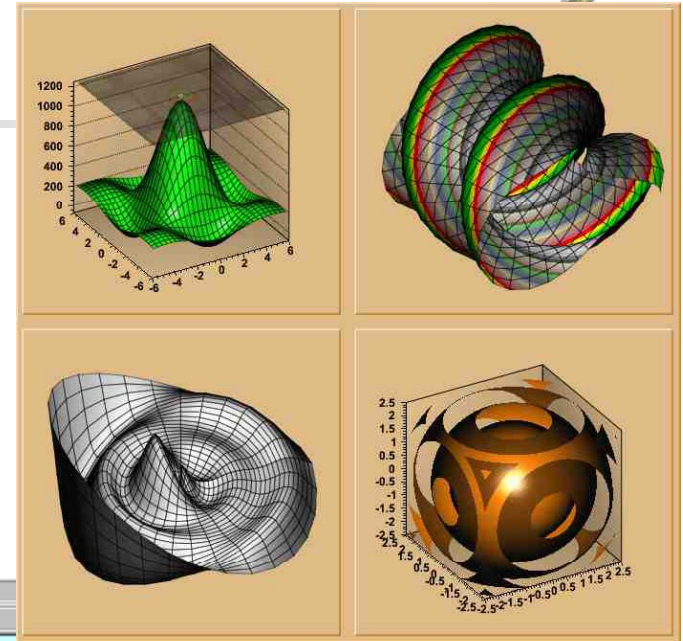
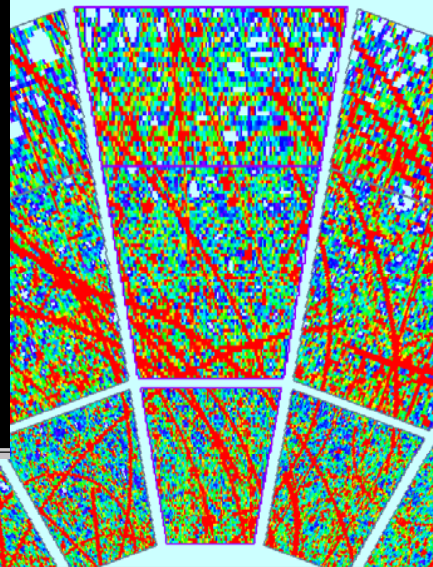
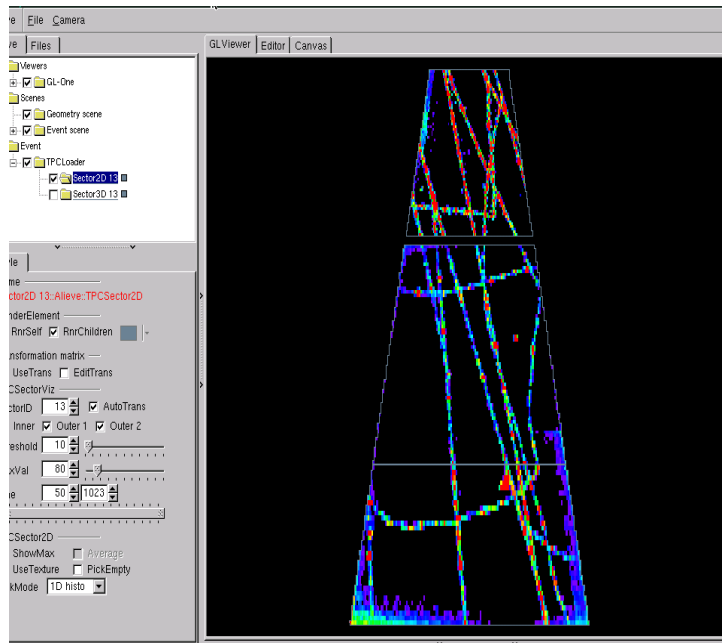
- The **TGeo** classes are now stable.
- Can work with different simulation engines (G3,G4,Fluka) (See **Virtual Monte Carlo**)
- G3->G4, G4->TGeo, TGeo \leftrightarrow GDML
- Used in online systems and reconstruction programs
- Built-in facilities for alignment
- Impressive gallery of experiments (35 detectors in **\$ROOTSYS/test/stressGeometry**)





3-D Graphics

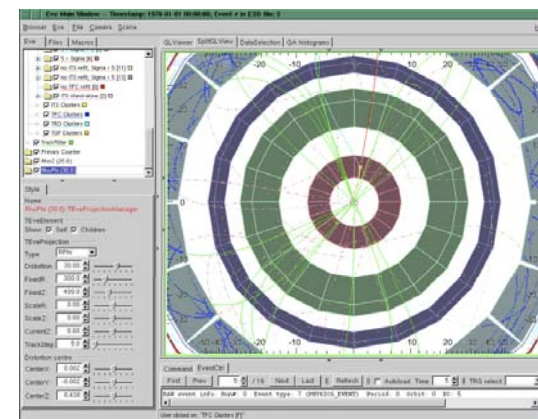
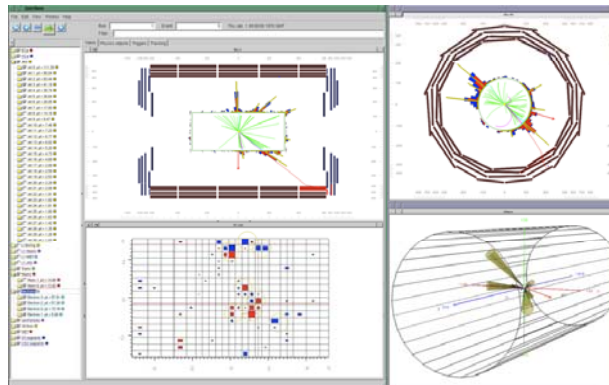
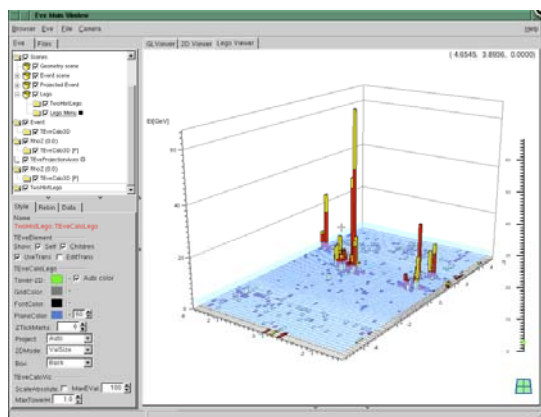
- Highly optimized GL views in
 - TPad
 - the GL viewer





Event Display: EVE

- **EVE** is a ROOT package (GL-based) for event displays.
- Developed in collaboration with Alice (**AliEve**) and CMS (**FireWorks**).
- Provides all the GUI widgets, browsers, GL infrastructure (far better than the old OpenInventor).
- Used now by many experiments (see eg **FAIRROOT**, **ILCROOT**) to display raw data, MC events or detector oriented visualization.





GUI

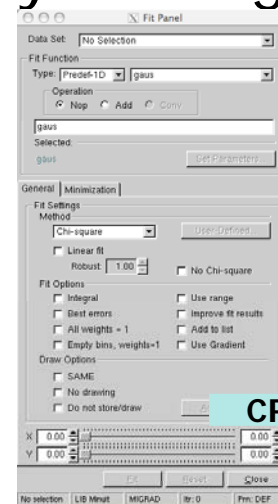


- Many enhancements in the GUI classes: browser, html browser, TABs, EVE widgets.
- GUI builder with C++ code generator. Note that the code generator works from any existing widget (CTRL/S). see poster

- class **TRecorder** see poster

- can store and replay a GUI session:
- All mouse events
- Keyboard input, including macro execution

- QT interfaces: a big pain, difficult to maintain with the successive versions of Qt.

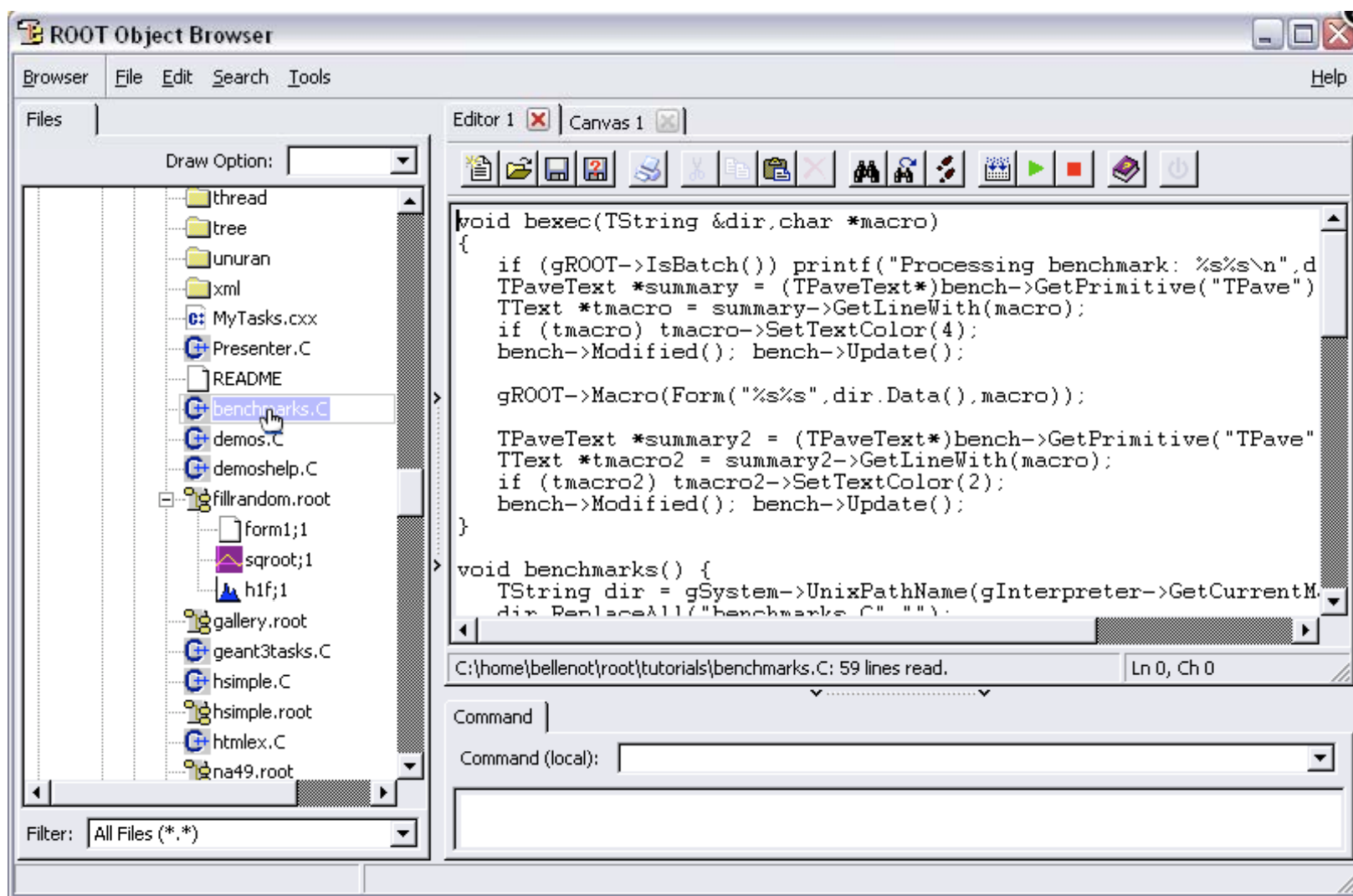


```
#include "kiostr.h"
void Fitpanel()
{
    // main frame
    TMainFrame *FitEditor47 = new TMainFrame(gClient->GetRoot(), 10, 10, kMainFrame, kVerticalFrame, k
    // composite frame
    TCompositeFrame *FitEditor47 = new TCompositeFrame(FitEditor47, 30, 20, kHorizontalFrame);
    TLabel *Label49 = new TLabel(FitEditor47, "Data Set:");
    Label49->SetJustify(kJLeft);
    Label49->SetMargin(0, 0, 0, 0);
    Label49->SetLength(1);
    TCompositeFrame *FitEditor47 = new TCompositeFrame(FitEditor47, 30, 20, kHorizontalFrame);
    TLong_t u_color; // will reflect user color changes
    gClient->SetColorByLabel("ffffff", u_color);
    // combo box
    TComboBox *ComboBox58 = new TComboBox(FitEditor47, 70, kHorizontalFrame, kSunkenFrame, kBox
    ComboBox58->AddEntry("No Selection", 0);
    ComboBox58->SetSize(20, 20);
    ComboBox58->Select(0);
    TCompositeFrame *FitEditor47 = new TCompositeFrame(FitEditor47, 70, 20, kHorizontalFrame);
    TComboBox *ComboBox58 = new TComboBox(FitEditor47, 70, kHorizontalFrame, kSunkenFrame, kBox
    ComboBox58->AddEntry("Selected(int)", 0, 0, "Data Set(int)");
    TFitEditor47->AddFrame(FitEditor47, new TLayoutHints(kHintsLeft, kHintsTop, 1, 0, 0, 0));
    // "Fit Function" group frame
    TGroupFrame *FitEditor47 = new TGroupFrame(FitEditor47, "Fit Function", kFitWidth);
    // composite frame
    TCompositeFrame *FitEditor47 = new TCompositeFrame(FitEditor47, 32, 25, kHorizontalFrame);
    TLabel *Label164 = new TLabel(FitEditor47, "Type:");
    Label164->SetJustify(kJLeft);
    Label164->SetMargin(0, 0, 0, 0);
    Label164->SetLength(1);
    TCompositeFrame *FitEditor47 = new TCompositeFrame(FitEditor47, 32, 25, kHorizontalFrame);
    TLabel *Label164 = new TLabel(FitEditor47, "Type:");
    Label164->SetJustify(kJLeft);
    Label164->SetMargin(0, 0, 0, 0);
    Label164->SetLength(1);
    gClient->SetColorByLabel("ffffff", u_color);
    // combo box
    TComboBox *ComboBox58 = new TComboBox(FitEditor47, 70, kHorizontalFrame, kSunkenFrame, kBox
    ComboBox58->AddEntry("User Func", 0);
    ComboBox58->SetSize(20, 20);
    ComboBox58->Select(0);
    TCompositeFrame *FitEditor47 = new TCompositeFrame(FitEditor47, 70, 20, kHorizontalFrame);
    TComboBox *ComboBox58 = new TComboBox(FitEditor47, 70, kHorizontalFrame, kSunkenFrame, kBox
    ComboBox58->AddEntry("User Func", 0);
    ComboBox58->SetSize(20, 20);
    ComboBox58->Select(0);
}
```

CTRL/S

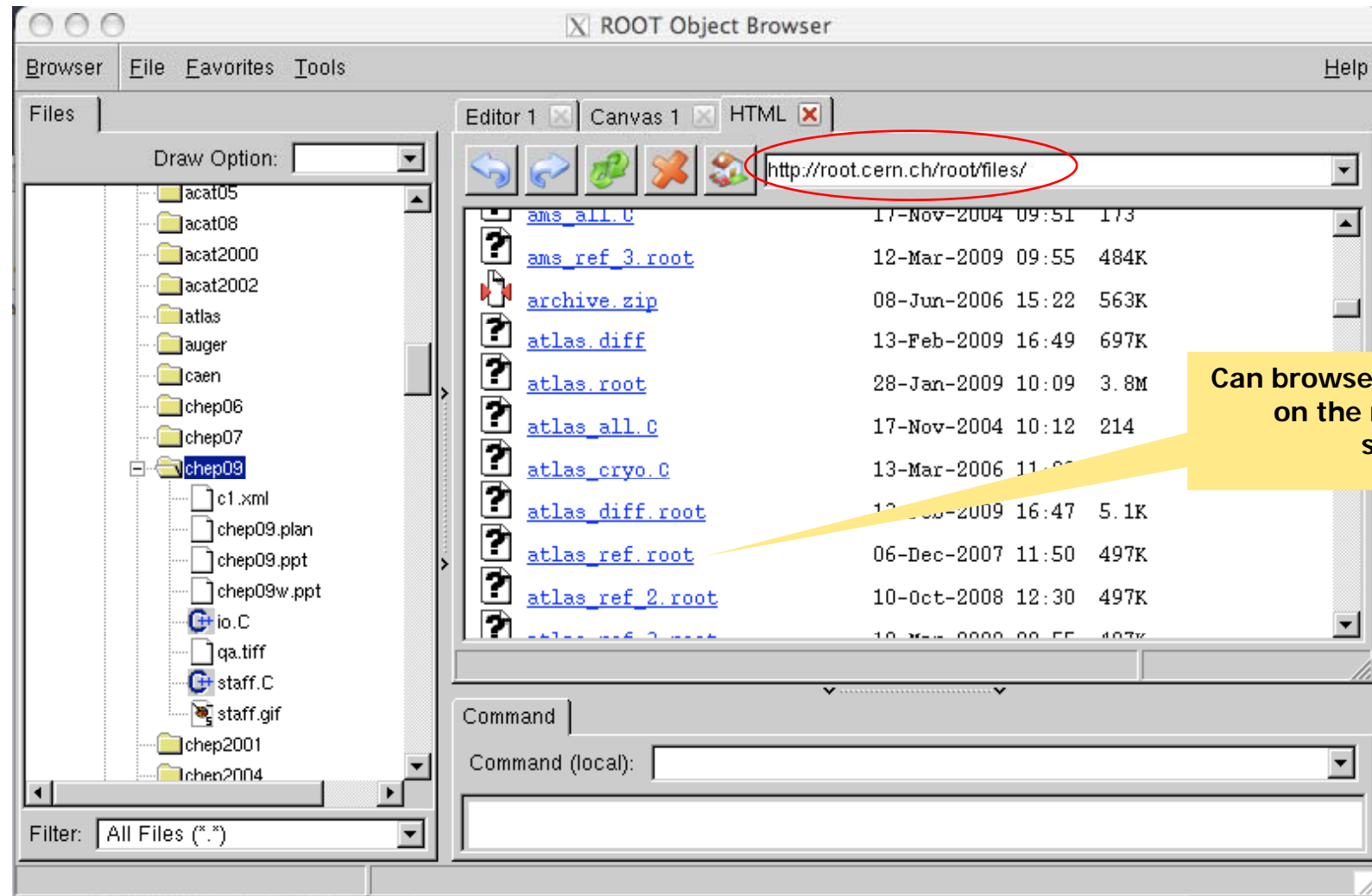


GUI Examples





GUI Examples II





RooFit/ RooStats

- The original Babar **RooFit** package has been considerably extended by **Wouter Verkerke**.
- Now structured in **RooFitCore** and RooFit
- RooFit is the base for the new **RooStats** package developed by Atlas and CMS (see Lorenzo's talk)



Documentation

- The Reference Guide is generated by **THtml** including **dynamic graphics**, **latex**, **macros**.
- The Tutorials doc is now generated automatically by THtml including graphics and the index.
- Release and Development Notes also generated automatically from SVN by THtml.
- Users Guide still in MS/Word format. Plan to convert it to (DocBook) with pdf and html versions.



Documentation examples

- Put as much as possible in the source in SVN, including html pages, scripts and graphics processed by **THtml**.

ROOT

//create the file, the tree and a new branch
TFile f("tree.root", "recreate");
TTree t1("t1", "a simple tree with 2 branches");
t1.Branch("px", &px, "px/F");
t1.Branch("py", &py, "py/F");

Quick Links:	ROOT Homepage	Class Index	Class Hierarchy	viewVC source	Search documentation...
Source:	header file	source file	viewVC header	class charts	Search
Sections:	class description	function members	data members		

ROOT » GRAF2D » GRAF » TGraphPolar

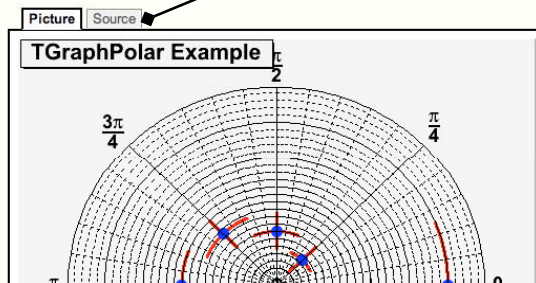
class TGraphPolar: public TGraphErrors



TGraphPolar : to draw a polar graph

TGraphPolar creates a polar graph (including error bars). A TGraphPolar is a TGraphErrors represented in polar coordinates. It uses the class TGraphPolargram to draw the polar axis.

Example:



TGraphPolar : to draw a polar graph

TGraphPolar creates a polar graph (including error bars). A TGraphPolar is a TGraphErrors represented in polar coordinates. It uses the class TGraphPolargram to draw the polar axis.

Example:

```
Picture Source

TCanvas * CPol = new TCanvas("CPol", "TGraphPolar Example", 500, 500);

Double_t theta[8];
Double_t radius[8];
Double_t etheta[8];
Double_t eradius[8];

for (int i=0; i<8; i++) {
    theta[i] = (i+1)*(TMath::Pi()/4.);
    radius[i] = (i+1)*0.05;
    etheta[i] = TMath::Pi()/8.;
    eradius[i] = 0.05;
}

TGraphPolar * grPl = new TGraphPolar(8, theta, radius, etheta, eradius);
grPl->SetTitle("TGraphPolar Example");

grPl->SetMarkerStyle(20);
grPl->SetMarkerSize(2.);
grPl->SetMarkerColor(4);
grPl->SetLineColor(2);
grPl->SetLineWidth(3);
grPl->Draw("PE");

// Update, otherwise GetPolargram returns 0
CPol->Update();
grPl->GetPolargram()->SetToRadian();

return CPol;
}
```




The new <http://root.cern.ch>

new

The old web site <http://root.cern.ch> has been replaced by a better version with improved contents and navigation using the **drupal** system.

old





Supported Platforms

- Linux (RH, SLCx, Suse, Debian, Ubuntu)
 - gcc3.4, gcc4.3 (32 and 64 bits)
 - icc10.1
- MAC (ppc, 10.4, 10.5)
 - gcc4.0.1, gcc4.3
 - icc10.1, icc11.0
- Windows (XP, Vista)
 - VC++7.1, VC++9
 - Cygwin gcc3.4, 4.3
- Solaris + OpenSolaris
 - CC5.2
 - gcc3.4, gcc4.3



Robustness & QA

- Impressive test suite **roottest** run in the nightly builds (several hundred tests)
- Working on GUI test suite (based on Event Recorder)

Project	Version	slc4_ia32_gcc34_dbg (Fri Mar 13 05:04 2009)		slc4_amd64_gcc34_dbg (Fri Mar 13 04:23 2009)		slc4_amd64_gcc34 (Fri Mar 13 06:45 2009)		slc4_amd64_gcc43 (Fri Mar 13 08:50 2009)		i686-slc5-gcc34-opt (Fri Mar 13 04:47 2009)		i686-slc5-gcc43-opt (Fri Mar 13 07:03 2009)		x86_64-slc5-gc (Fri Mar 13 04:..	
LCGCMT	LCGCMT-preview	build	tests	build	tests	build	tests	build	tests	build	tests	build	tests	build	
ROOT	ROOT_today	build	tests	build	tests	build	tests	build	tests	build	tests	build	tests	build	
RELAX	RELAX-preview	build	tests	build	tests	build	tests	build	tests	build	tests	build	tests	build	
CORAL	CORAL-preview	build (10)	tests (3)	build	tests (6)	build	tests (6)	build (62)	tests (6)	build (10)	tests (3)	build (10)	tests (3)	build	t
POOL	POOL-preview	build (1)	tests	build (1)	tests	build (1)	tests	build (44)	tests	build (1)	tests (1)	build (44)	tests	build (1)	
COOL	COOL-preview	build	tests	build	tests	build	tests	build	tests	build	tests (7)	build	tests (9)	build	
GAUDIATLAS	GAUDI_ATLAS	build (56)	tests	build (56)	tests	build (56)	tests	build (103)	tests	build (56)	tests	build (103)	tests	build (56)	
GAUDI	GAUDI_HEAD	build (54)	tests	build (54)	tests	build (54)	tests	build (100)	tests	build (54)	tests	build (100)	tests	build (54)	

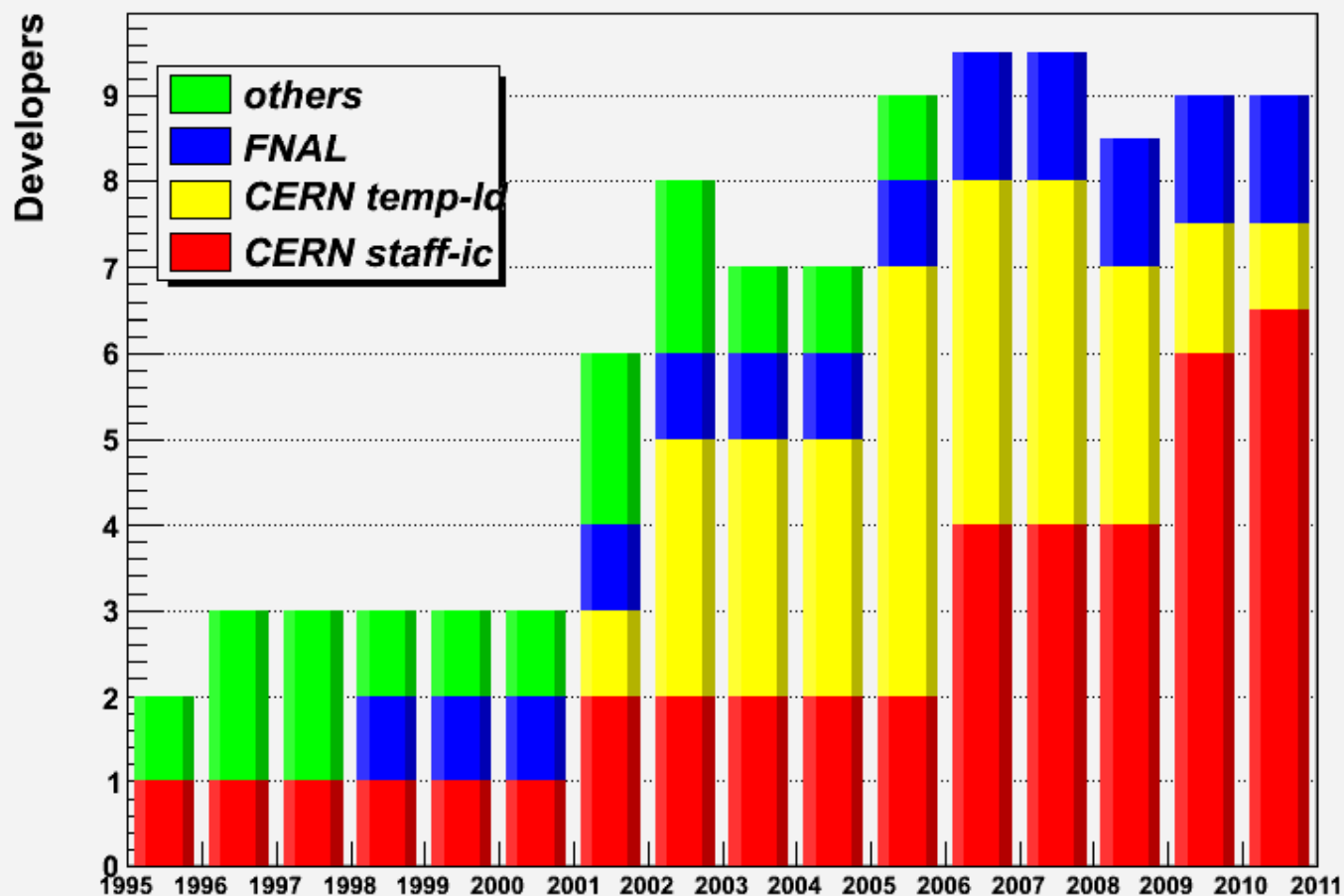
Platforms not ready: osx105_ia32_gcc401_dbg win32_vc71_dbg



ROOT developers more stability



ROOT manpower





Summary

- After 15 years of development, good balance between consolidation and new developments.
- The ROOT main packages (I/O & Trees) are entering a consolidation, optimization phase.
- We would like to upgrade CINT with the LLVM C++0x compliant compiler (CLING).
- Better documentation and User Support
- More stable manpower
- Usage rapidly expanding outside HEP