What is New in Core

Fons Rademakers

Library Reorganization

- Split off from libCore:
  - libRIO
  - libNet
  - libMath (soon)
- These libraries are now loaded at run-time via the plugin manager

Minimal Set of Libraries

- root.exe only links with -lCore -lCint -lRint
- Same done for proofserv.exe
- Reduced startup time and use of resources
- “root-config --libs” still provides same list of libraries for backward compatibility

Plugin Manager Recap

- With the increased relying on “loading-on-demand” comes and increased role for the plugin manager
- Plugins are described in the [system].rootrc files
- Class to library mapping and library dependencies are described in the [system].rootmap files

Rootmap Files

- Two fold functionality:
  - Maps classes, typedefs and globals, to library
    - Used by CINT when unknown “type” found
  - Describes library dependencies
    - Used by TSystem::Load()

Layout of Rootmap

```
Library.TH1:                              libHist.so libMatrix.so

Library.TCanvas:                          libGpad.so libGraf.so libHist.so

Library.TXNetFile:                        libNetx.so libNet.so libRIO.so

Library.TMatrixD:                         libMatrix.so

Library.ROOT@@Math@@Cartesian3D<Double32_t>: libMathCore.so
```

Location of Rootmap

- Rootmap files will be loaded at startup from the following locations:
  - $ROOTSYS/etc/system.rootmap
  - $HOME/.rootmap
  - ./.rootmap
  - Any file in $[DY]LD_LIBRARY_PATH with a name like:
    - rootmap_<xxx>
    - <xxx>.rootmap

Rootmap Generation

- The system rootmap file is generated via
  - "make map" or "make install"
- With the next release we will automatically generate one rootmap file per library, i.e. rootmaps will always be up-to-date
- On MacOS X ACliC generates a macro_C.rootmap file to describe dependencies

Incompatibilities

- The library reorganization and include file optimization caused some backward incompatibilities:
  - TBuffer is now pure virtual
  - A large number of header files have been removed from other header files
- Please try the current dev version asap and update your code

SEAL Migration

- Last remaining item to be migrated was the SEAL plugin service
- Now available as part of libReflex for the experiments that need it
- Main features:
  - Exclusive dependency on libReflex
  - Plugin factory declared via one macro in the user code
  - Load on demand via rootmap files (genmap)

SEAL Migration

- Last remaining item to be migrated was the SEAL plugin servic

```
class MyClass : public ICommon {
  MyClass(int, ISvc*);
  ...
};
```

MyClass.h

- Now available as part of libReflex for the experiments that need it
- Main features:
  - Exclusive dependency on libReflex
  - Plugin factory declared via one macro in the user code
  - Load on demand via rootmap files (genmap)

SEAL Migration

- Last remaining item to be migrated was the SEAL plugin service
- Now available as part of libReflex for the experiments that need it
- Main features:
  - Exclusive dependency on libReflex
  - Plugin factory declared via one macro in the user code
  - Load on demand via rootmap files (genmap)

```
class MyClass : public ICommon {
  MyClass(int, ISvc*);
  ...
};
```

MyClass.h

```
PLUGINSVC-FACTORY(MyClass,ICommon*(int,ISvc*));
/* implementation */
```

MyClass.cpp

SEAL Migration

- Last remaining item to be migrated was the SEAL plugin service
- Now available as part of libReflex for the experiments that need it
- Main features:
  - Exclusive dependency on libReflex
  - Plugin factory declared via one macro in the user code
  - Load on demand via rootmap files (genmap)

```
class MyClass : public ICommon {
  MyClass(int, ISvc*);
  ...
};
```

MyClass.h

```
PLUGINSVC-FACTORY(MyClass,ICommon*(int,ISvc*));
/* implementation */
```

MyClass.cpp

```
Library.MyClass:          MyLibrary.so
Library.AnotherClass:     MyLibrary.so
```

rootmap

SEAL Migration

- Last remaining item to be migrated was the SEAL plugin service
- Now available as part of libReflex for the experiments that need it
- Main features:
  - Exclusive dependency on libReflex
  - Plugin factory user code
  - Load on demand via rootmap files (genmap)

```
class MyClass : public ICommon {
  MyClass(int, ISvc*);
  ...
};
```

MyClass.h

```
PLUGINSVC-FACTORY(MyClass,ICommon*(int,ISvc*));
/* implementation */
```

MyClass.cpp

```
Library.MyClass:          MyLibrary.so
Library.AnotherClass:     MyLibrary.so
```

rootmap

```
...
ISvc* svc = ...
ICommon* myc;
myc = PluginSvc::create<ICommon*>("MyClass",10, svc);
if ( myc ) {
  myc->doSomething();
}
```

Program.cpp

New Core Features

- New TFileStager class defining interface to generic stager
- Implementation for xrootd staging interface

New Core Features

- New TFileStager class defining interface to generic stager
- Implementation for xrootd staging interface

```
// Open connection to the stager
root[] stg = TFileStager::Open("root://lxb6046.cern.ch")

// Issue a stage request
root[] stg->Stage("/alice/sim/2006/pp_minbias/121/168/root_archive.zip")

// Check the status
root[] stg->IsStaged("/alice/sim/2006/pp_minbias/121/168/root_archive.zip")
```

New Core Features

- New static function TFile::Cp() to copy any files (also non-ROOT files) via the ROOT I/O plugins
- TFile::Open() has new option “CACHEREAD”

New Core Features

- New static function TFile::Cp() to copy any files (also non-ROOT files) via the ROOT I/O plugins
- TFile::Open() has new option "CACHEREAD"

```
root [0] TFile::SetCacheFileDir("/tmp/fons")

root [1] TFile *f = TFile::Open("http://root.cern.ch/files/aleph.root", "CACHEREAD")
[TFile::Cp] Total 0.11 MB       |====================| 100.00 % [8.8 MB/s]
Info in <TFile::Open>: using local cache copy of http://root.cern.ch/files/aleph.root
[/tmp/fons/files/aleph.root]

root [2] f->GetName()
(const char* 0x41dd2d0)"/tmp/fons/files/aleph.root"
```

New Core Features

- New system, CPU, memory and process info methods in TSystem:
  - GetSysInfo(), GetCpuInfo(), GetMemInfo(), GetProcInfo()
- Support for TSystem::StackTrace() on Mac OS X
- New class TAtomicCount providing atomic operations on a long
  - Thread safe reference counting

Updated IO Plugins

- The TCastorFile and TRFIOFile plugins support the latest Castor 2.1.2
- TCastorFile supports any form of authentication, including GSI/Globus
- Supports an incredible set of "url's"

Updated IO Plugins

- The TCastorFile and TRFIOFile plugins support the latest Castor 2.1.2
- TCastorFile supports any form of authentication, including GSI/Globus

```
castor:/castor/cern.ch/user/r/rdm/bla.root

castor://castor.cern.ch/user/r/rdm/bla.root

castor://stager_host:stager_port/?path=/castor/cern.ch/user/r/rdm/
bla.root&svcClass=MYSVCLASS&castorVersion=MYCASTORVERSION

castor://stager_host/?path=/castor/cern.ch/user/r/rdm/
bla.root&svcClass=MYSVCLASS&castorVersion=MYCASTORVERSION

castor:///?path=/castor/cern.ch/user/r/rdm/
bla.root&svcClass=MYSVCLASS&castorVersion=MYCASTORVERSION

castor:///?path=/castor/cern.ch/user/r/rdm/
bla.root&svcClass=MYSVCLASS&castorVersion=MYCASTORVERSION&rootAuth=3
```

root.cern.ch

- Provides:
  - web, ftp, cvs, ssh, wiki, mailing lists, forum, mysql, ...
- Will soon migrate from one 6 CPU PIII HP Netserver to two dual Core 2 Duo (4 cores each) servers (one hot spare)
- Thanks IT !

Forum and Mailing List

- Forum
  - Currently 2185 registered users
  - 4560 topics, 18620 posts, 14.3 posts/day
- Mailing list
  - Currently 1275 registered users
  - 5.3 posts/day

Bug Reporter

- Bug reporting and management is done via Savannah platform
- Statistics:
  - 96 of 913 Open issues (817, 89% closed)
  - About 10 issues opened per week
  - Most closed within a week

Web Site

- Overhaul needed
- Gradually replacing static pages by Wiki pages to make maintenance more collaborative
- CINT and PROOF pages already using Wiki
- Wiki experience will improve with faster root.cern.ch

Version Control

- The recent code movements exposed one of the many short comings of CVS
  - No history tracking of moved files and directories
- As soon as we move to the new IT servers we plan to migrate to SVN

License

- ROOT is now available under the LGPL
- More importantly, all included 3rd party code is also LGPL or LGPL compatible
- ROOT passed the strict "Debian Legal" scrutiny

Miscellaneous

- Keep aggressively following new OS and compiler releases
  - gcc 4.2, Intel icc v10 and VC++ v8 supported
  - Upcoming Mac OS X Leopard supported (32 and 64 bit versions)
- gfortran of the gcc 4.2 suite now good enough to compile CERNLIB, needed for Pythia6 and h2root and g2root