

CMS – ROOT5 to ROOT6

What's different (summary)

- **No more REFLEX (not just for dictionaries)**
 - No more CINT/Cintex
 - Need enum and typedef declarations in spec files
 - Need more dictionaries in spec files
 - LinkDef files containing #include do not work
 - Need rootmap files for dictionary loading
 - cling, not CINT, compiles macros
 - Many missing #includes and std::qualifiers (spurious ones in ROOT5)
 - New way to check for missing dictionaries
 - Many class checksums will change
 - Minor API changes
 - Exceptions didn't propagate through cling layer (fixed in ROOT6 now)
 - Couldn't handle assembler (boost::shared_ptr) (fixed in ROOT6 now)
 - Supports C++11 (Yes!) (and C++14)
 - Build system changes (not my department) rootcint=>rootcling for dictionaries

Main Reflex classes used by CMS

- `Reflex::Type` => a type
- `Reflex::Object` => an object of a type
- `Reflex::Member` => a function/data member of a class
- `Reflex::Base` => a base class of a class
- `Reflex::TypeTemplate` => a class template
- `Reflex::MemberIterator` => iterator over class members
- `Reflex::BaseIterator` => iterator over base classes

Replacements for main Reflex classes

- `Type` => `edm::TypeWithDict`
- `Object` => `edm::ObjectWithDict`
- `Member` => `edm::MemberWithDict` (for a data member)
- `Member` => `edm::FunctionWithDict` (for a function member)
- `Base` => `edm::BaseWithDict`
- `TypeTemplate` => `TypeWithDict` member functions
- `MemberIterator` => `edm::IterWithDict` + `TypeWithDict` helpers
- `BaseIterator` => `edm::IterWithDict` + `TypeWithDict` helpers

TypeWithDict

- TypeWithDict => a data type (the big enchilada)
 - a class or struct (uses TClass)
 - a built in type (uses TDataType)
 - an enum (uses TEnum – new for ROOT6)
 - pointers (or pointers to const) to any of the above
 - C-style arrays of any of the above (except pointers)
 - const qualified any of the above
 - (l-value) references (const or not) to any of the above
- Constructed from type name or type_info or TClass
- Unlike Reflex::Type, does not support arbitrary data type
 - No pointers to pointers
 - No pointers to C-style arrays
 - No C-style arrays of pointers
 - Nothing not listed as supported above
 - Has type_info only for classes and built-ins

Other *WithDict classes

- ObjectWithDict => an object of a data type
 - A TypeWithDict and an address
- MemberWithDict => a data member
 - uses TDataMember
- FunctionWithDict => a member function
 - uses TMethod
 - can invoke the function with specified arg values
- BaseWithDict => a base class
 - uses TBaseClass
- IterWithDict => iterator (three types)
 - over member functions, data members, or base classes
 - enables range for loop to be used
 - supports only TList, but could be easily generalized <TList => T>
 - uses TIter and TList

More Reflex classes used by CMS

- Reflex::PropertyList => properties of a type
 - used for is_transient, split level, basket size
 - replaced by TClass::GetAttributeMap() (TDictAttributeMap)
- Reflex::SharedLibrary
 - usage replaced by explicit dlopen call
- Reflex::Tools (a namespace)
 - Demangle() replaced by abi::__cxa_demangle()
 - MakeVector() replaced by std::vector() and push_back() calls
- Reflex::NewDelFunctions => Object deleters
 - supported inside ObjectWithDict
- Reflex::CollFuncTable & Reflex::Environ => don't ask!
 - replaced by TVirtualCollectionProxy (very hard!)

Dictionary Spec files (.xml)

- Same as in ROOT 5, but:
 - need specs for more classes
 - not fully understood why (by me), but easy to add as needed
 - need specs for enums (enum name="XXX")
 - enums don't have dictionaries in ROOT6, but specs needed anyway
 - need specs for some typedefs (typedef name="XXX")
 - so TypeWithDict can be constructed using typedef name
 - checksums of many classes will change
 - checksums ignore typedefs (true also in latest ROOT5)
 - checksums depend on base classes (true also in latest ROOT5)
 - change caused by (unsigned) long long => (U)Long64_t
 - change caused by basic_string<char> => string
 - future change will be caused by adding "std::"
- LinkDef files with #include "MyClass.h" don't work
 - replaced by .xml
 - still works if header is on command line along with LinkDef file

Macros don't compile

- Cling much more restrictive than CINT
- 2/3 of our macros don't compile
- ~100 different compilation errors, many easy to fix
- Undeclared identifiers are most serious problem
- Still working this issue
- Most of our macros not used in tests, so problem was hidden

Miscellany

- New way to check for missing dictionaries
 - TClass::GetMissingDictionaries
- Minor ROOT API changes
 - No big deal, if it doesn't compile, replace it
- Hundreds of missing #includes and std::qualifiers
 - ROOT5 headers include <iostream> and “using” directives
- rootmap files needed for dictionary loading
- Exceptions didn't propagate through cling layer
 - We implemented workaround
 - Fixed now in ROOT6
- Could not handle assembler (boost::shared_ptr)
 - Replaced with std::shared_ptr
 - Harder than it sounds (coupled with other boost usage)
 - Fixed now in ROOT6

What took so long?

- Massive use of Reflex for introspection
- ROOT6 was a work in progress
- We were its major testers
- We found and reported numerous bugs
- Many essential features (e.g. TEnum) provided late.