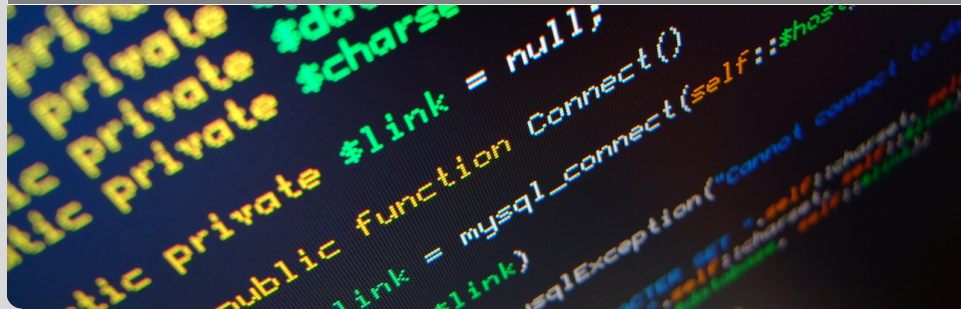# rootJS - results of the implementation phase

Software Engineering Practice - Winter Term 2015/16

C. Wolff, M. Früh, S. Rajgopal, C. Haas, J. Schwabe, T. Beffart | February 12, 2016

Outline

Required Criteria

The bindings must

- Work on Linux
- Allow the user to interact with any ROOT class from the Node.js JavaScript interpreter
- Accept C++ code for just-in-time compilation
- Update dynamically following changes to C++ internals
- Provide asynchronous wrappers for common I/O operations (i.e. file and tree access)

Required Criteria

The bindings must

- Work on Linux ✓
- Allow the user to interact with any ROOT class from the Node.js JavaScript interpreter
- Accept C++ code for just-in-time compilation
- Update dynamically following changes to C++ internals
- Provide asynchronous wrappers for common I/O operations (i.e. file and tree access)

Required Criteria

The bindings must

- Work on Linux ✓
- Allow the user to interact with any ROOT class from the Node.js JavaScript interpreter ✓
- Accept C++ code for just-in-time compilation
- Update dynamically following changes to C++ internals
- Provide asynchronous wrappers for common I/O operations (i.e. file and tree access)

Required Criteria

The bindings must

- Work on Linux ✓
- Allow the user to interact with any ROOT class from the Node.js JavaScript interpreter ✓
- Accept C++ code for just-in-time compilation ✓
- Update dynamically following changes to C++ internals
- Provide asynchronous wrappers for common I/O operations (i.e. file and tree access)

Required Criteria

The bindings must

- Work on Linux ✓
- Allow the user to interact with any ROOT class from the Node.js JavaScript interpreter ✓
- Accept C++ code for just-in-time compilation ✓
- Update dynamically following changes to C++ internals ✓
- Provide asynchronous wrappers for common I/O operations (i.e. file and tree access)

Required Criteria

The bindings must

- Work on Linux ✓
- Allow the user to interact with any ROOT class from the Node.js JavaScript interpreter ✓
- Accept C++ code for just-in-time compilation ✓
- Update dynamically following changes to C++ internals ✓
- Provide asynchronous wrappers for common I/O operations (i.e. file and tree access) ✓

Fullfilled criteria
Changes in design
C. Wolff, M. Früh, S. Rajgopal, C. Haas, J. Schwabe, T. Beffart – rootJS - implementation
February 12, 2016
3/6

Limiting Criteria

The bindings should not

- Add any extending functionality to the existing ROOT framework
- Necessarily support previous/future ROOT versions

Limiting Criteria

The bindings should not

- Add any extending functionality to the existing ROOT framework ✓
- Necessarily support previous/future ROOT versions

Limiting Criteria

The bindings should not

- Add any extending functionality to the existing ROOT framework ✓
- Necessarily support previous/future ROOT versions ✓

AsyncRunner

- We added a helper class for asynchronous operations, 'AsyncRunner'
- What does it do?
- Why do we need it?

MetaInfo

- We added a helper class to encapsulate differences between TGlobalFunctions and ...

- What does it do?

- Why do we need it?

Classexposer

- We added a helper class for asynchronous operations, 'AsyncRunner'
- What does it do?
- Why do we need it?

Statistics

- 4200 LOC
- 300+ commits