boost
C++ LIBRARIES

Boost.Any

Dmitri Nesteruk
http://activemesa.com
dn@activemesa.com

pluralsight
hardcore developer training

Overview

- **Languages such as C# or Java have a root object type**
  - It's possible to cast a primitive to that type, i.e.
    ```
    object o = 123; // boxing
    ```
  - Thus, we can have a list of objects whose type can be checked at runtime.
- **C++ has discriminated unions**
  - But these are limited to just a few types that you define
  - Only accepts primitive types
- **boost::any**
  - Can contain values of any type
  - Values are stored as they are (i.e., 42 is strictly an `int`)
  - Can be used in collections
    - E.g., ProgramOptions' `variables_map`
  - API for getting/checking the value of a particular type

Interface

- **`boost::any<T>`**
  - T can be anything
  - T must be copy-constructable
- **Creation is easy**
  - `any w; // has no value`
  - `any x { 2.0 };`
  - `vector<any> y { 42, "life" };`
  - `any z = string("test");`
- **Queries**
  - `empty()`
    checks if we have a value
  - `type()`
    returns `typeid` of containing instance
- **`any_cast`**
- **Global function**
- **Can pass either pointer or reference to an instance of any**
- **When passed a reference**
  - Returns ref to contained value if types match
  - Throws otherwise
- **When passed a pointer**
  - Returns pointer to contained value if types match
  - `nullptr` otherwise
  - No exception thrown

Summary

- **`#include <boost/any.hpp>`**
- **Create an instance of any and assign it a value**
  - Any value ☺
- **Use `any_cast<Type>(anyInstance)` to get the value**
- **Passing a reference to any will return**
  - A reference to the contained object; or
  - Will throw an exception
- **Passing a pointer to any will return**
  - A pointer to the contained object if the types match
  - `nullptr` otherwise