C++11 Regular-Expression Library

By Brian Overland

Jun 25, 2013

20.8. Catching RegEx Exceptions

Catching exceptions can be important when working with regular expressions, especially if you build a new regular-expression string at runtime or let the end user specify a pattern.

Remember that a regular expression object is compiled at runtime. If the pattern is ill-formed, the program throws an exception, causing abrupt termination unless the exception is caught. Catching the exception enables you to terminate more gracefully, or even—if you choose—continue execution after taking appropriate action (such as reporting the error to the user).

Regular-expression errors belong to the **regex_error** class, which is declared in the <regex> header and is part of the **std** namespace.

```
#include <regex>
using std::regex_error;
```

The **regex_error** class supports a **what** function, which returns a text string, and a **code** function, which returns an integer. The following example (which assumes the appropriate **#include** directives and **using** statements have been given) responds to a poorly formed **regex** object.

```
try {
     regex re("*\\bW[a-z]*");
     string s = "The White Rabbit is very late.";
     if (regex_search(s, re)) {
          cout << "Search string found." << endl;
     }
} catch (regex_error e) {
     cout << e.what() << endl;
     cout << "CODE IS: " << e.code() << endl;
}
```

When these statements are run, the program prints:

```
regex_error(error_badrepeat):
One of *?+{ was not preceded by a valid regular expression.
CODE IS: 10
```

The error is thrown as soon as there is an attempt to build a bad regular expression (because the regular-expression object, remember, is compiled at runtime).

```
     regex re("*\\bW[a-z]*");  // Throws an exception!
```

Here you should be able to see the problem. The first use of the asterisk (*)—which means to repeat the preceding expression zero or more times—was not preceded by any expression; it was therefore not a meaningful use of regular-expression grammar. This is duly reported as a “badrepeat” error as long as exception handling is provided as just shown.

Share This Save To Your Account < Back Page 8 of 10 Next >