Subject: NULL
Posted by phirox on Tue, 10 Jun 2008 19:45:39 GMT
View Forum Message <> Reply to Message

C++ does not inherently support a null pointer yet, but it will in C++0x. It is handy to avoid confusion between 0 and null(as in arithmetic or a pointer address).

I have always used a trick for this(not by my own design) and adapted it a little for U++.

This should go into a header file:

```
namespace Upp
{
 namespace nil
 {
  class null
  {
   // Cannot make another instance, copy or assign.
   null(null const &);
   null &operator=(null const &);

  protected:
   null() {}

  public:
   // null may be implicitly converted to any pointer type.
   template<typename T>
   operator T*() const { return 0; }

   // null is always false.
   bool operator!() const { return true; }

   // The address or reference of null is null.
   null const &operator&() const { return *this; }
   null const &operator*() const { return *this; }
  };

  struct nullptr : null {
   static nullptr const instance;
  private:
   nullptr() {}
  };
 }

 // Constant reference to constant singular nullptr instance.
 static nil::null const &null = nil::nullptr::instance;
}
```

```
// Replace any NULL macro for null
#ifdef NULL
#undef NULL
#endif
#define NULL ::Upp::null
```

And this into a library file:

```
nil::nullptr const nil::nullptr::instance;
```

You can use either null or NULL and when you use them with numbers instead of pointers it will create a compiler error. It is also compatible with the older styles of NULL.
