A Few Tips for Using Microsoft Visual C++*

Jeffrey D. Oldham

2000 Jan 28

The only experience I have using Microsoft Visual C++ 6.0 was tonight (2000 Jan 28). I am writing what I learned, hoping that it will reduce your frustration if you choose to use these tools. **Note I do not have time to support using Microsoft compilers.**

(In the best of all worlds, every compiler would strictly obey the ANSI/-ISO C++ standard. Unfortunately, standards change through time, and programmers cannot keep up. Thus, different compilers sometimes require different syntax.)

1 Using STL Containers

Microsoft seems to require specifying `std::` before the beginning of STL container names. For example, one can declare a `vector` using

```
std::vector<int> v;
```

`std` is the "standard" namespace. A namespace is a set of functions, classes, and objects. Namespaces were introduced because different libraries used the same function or class names. Prepending the library's namespace followed by `::` yields unique names.

2 Friend Templates

Microsoft has not yet updated their products to reflect the ANSI/ISO C++ standard regarding friend templates. If you see a `friend` function declaration with empty `<>` inside some class definition like

```
friend void bar<>(foo<T>);
```

convert it to

```
friend void bar(foo<T>);
```

to get it to compile. This may break your code somewhere else.

We do not yet have the vocabulary to explain the issues, but an explanation is available.