My namespace importing trick imported the same three namespaces into each top-level namespace, yet it worked?

devblogs.microsoft.com/oldnewthing/20180525-00

May 25, 2018

Raymond Chen

A little while ago, I noted a technique formally known as namespace composition. There was one section that appeared to confuse some people:

```
namespace ABI
{
  using namespace Windows::System::Profile::SystemManufacturers;
  using namespace Windows::UI::ViewManagement;
  using namespace Windows::Security::Cryptography;
}

namespace cx
{
  using namespace Windows::System::Profile::SystemManufacturers;
  using namespace Windows::UI::ViewManagement;
  using namespace Windows::Security::Cryptography;
}

namespace winrt
{
  using namespace Windows::System::Profile::SystemManufacturers;
  using namespace Windows::UI::ViewManagement;
  using namespace Windows::Security::Cryptography;
}
```

Was this a copy/paste error? After all, the same three namespaces are being imported each time.

Well, no, actually. The text is the same, but each one is interpreted differently.

Let's take a simpler example:

```
namespace X { namespace W { void f(); }}
namespace Y { namespace W { void f(); }}
namespace W { void f(); }

namespace X
{
    using namespace W;
    auto do_something = f;
}

namespace Y
{
    using namespace W;
    auto do_something = f;
}

namespace Z
{
    using namespace W;
    auto do_something = f;
}
```

Each of the three namespaces contain a `using namespace W;`, but each one refers to a different namespace, which you can see by pasting the above into Compiler Explorer and observing the definitions of `X::do_something`, `Y::do_something`, and `Z::do_something`.

The first `using namespace W;` takes place inside a `namespace X`, so the search begins relative to that namespace, and we find it at `::X::W`.

Similarly, the second `using namespace W;` takes place inside a `namespace Y`, so the search begins relative to that namespace, and we find it at `::Y::W`.

The third `using namespace W;` takes place inside a `namespace Z`, so the search begins relative to that namespace. There is no `::Z::W`, so we resume our search at the next outer namespace, which is the global namespace, and we find it as `::W`.

Even though the three namespace imports are textually identical, they have different effects because they each occur in different contexts.

I wrote it this way because it showed that I was "pulling in" the relative namespace declarations into the corresponding first-level namespace.

Raymond Chen

Follow