How do I create a Windows Runtime method that accepts a lambda?

devblogs.microsoft.com/oldnewthing/20190925-00

September 25, 2019

Raymond Chen

A customer wanted to create a Windows Runtime method that accepted a lambda. But lambdas aren't valid Windows Runtime types.

```
// C++/WinRT IDL
runtimeclass MyClass
{
  template<typename TLambda>
  void ApplyFilter(TLambda&& filter); // Does not compile
}

// C++/CX
public ref class MyClass
{
public:
  template<typename TLambda>
  void ApplyFilter(TLambda&& filter); // Does not compile
};
```

What can we use instead of lambdas?

The Windows Runtime doesn't support lambdas, but it supports what you typically use the lambda for, which is providing a callback. The thing that represents a function is a *delegate*.

You can think of a delegate as the Windows Runtime version of a `std::function`. It's an object that can hold a lambda or a static function or an object combined with a member function.

```
// C++/WinRT IDL
delegate Boolean ThingFilter(Thing thing);

runtimeclass MyClass
{
  void ApplyFilter(ThingFilter filter);
}

// C++/CX
public delegate bool ThingFilter(Thing^ thing);

public ref class MyClass
{
public:
  void ApplyFilter(ThingFilter^ filter);
};
```

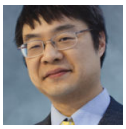
You can use the delegate like a function object.

```
// C++/WinRT
void MyClass::ApplyFilter(ThingFilter const& filter)
{
  things.erase(std::remove_if(things.begin(), things.end(),
    [&](auto&& thing) { return !filter(thing); }),
    things.end());
}

// C++/CX
void MyClass::ApplyFilter(ThingFilter^ filter)
{
  things.erase(std::remove_if(things.begin(), things.end(),
    [&](Thing^ thing) { return !filter(thing); }),
    things.end());
}
```

Most language projections let you pass a lambda directly. C++/CX is the outlier here. It requires you to wrap it inside an explicit `ThingFilter`. The other languages will do the wrapping automatically.

```
// C++/WinRT
c.ApplyFilter([=](auto&& thing) { return IsOkay(thing); });

// C++/CX
c->ApplyFilter(
    ref new ThingFilter([=](Thing^ thing) { return IsOkay(thing); }));

// C#
c.ApplyFilter(thing => IsOkay(thing));

// JavaScript
c.applyFilter(thing => isOkay(thing));

// Visual Basic
c.ApplyFilter(Function (thing As Thing) IsOkay(thing))
```

Raymond Chen



Follow