CoGetInterfaceAndReleaseStream does not mix with smart pointers

devblogs.microsoft.com/oldnewthing/20151023-00

October 23, 2015

Raymond Chen

One horrible gotcha of the `CoGetInterfaceAndReleaseStream` function is that *it releases the stream*. This is a holdover from the old days before smart pointers. The function released the stream to save you from having to call `Release` yourself. But nowadays, everybody is using smart pointers, so you never had to type `Release` to begin with. The problem is that you can fall into a double-`Release` situation without realizing it.

```
// Code in italics is wrong
void GetTheInterface(REFIID iid, void** ppv)
{
  Microsoft::WRL::ComPtr<IStream> stream;
  GetTheStream(&stream);
  CoGetInterfaceAndReleaseStream(stream.Get(), iid, ppv);
}

void GetTheInterface(REFIID iid, void** ppv)
{
  ATL::CComPtr<IStream> stream;
  GetTheStream(&stream);
  CoGetInterfaceAndReleaseStream(stream, iid, ppv);
}

void GetTheInterface(REFIID iid, void** ppv)
{
  _com_ptr_t<IStream> stream;
  GetTheStream(&stream);
  CoGetInterfaceAndReleaseStream(stream, iid, ppv);
}

struct Releaser
{
    void operator()(IUnknown* p) { if (p) p->Release(); }
};

void GetTheInterface(REFIID iid, void** ppv)
{
  IStream* rawStream;
  GetTheStream(&rawStream);
  std::unique_ptr<IStream, Releaser> stream(rawStream);
  CoGetInterfaceAndReleaseStream(stream.get(), iid, ppv);
}
```

All of the code fragments above look completely natural, and they all have a bug because the smart pointer object `stream` is going to call `Release` at destruction, which will double-release the pointer because `CoGetInterfaceAndReleaseStream` *already released it*.

This type of bug is really hard to track down.

One way to fix this is to call the function and tell the smart pointer class that you are transferring ownership of the stream to the function.

```
void GetTheInterface(REFIID iid, void** ppv)
{
  Microsoft::WRL::ComPtr<IStream> stream;
  GetTheStream(&stream);
  CoGetInterfaceAndReleaseStream(stream.Detach(), iid, ppv);
}

void GetTheInterface(REFIID iid, void** ppv)
{
  ATL::CComPtr<IStream> stream;
  GetTheStream(&stream);
  CoGetInterfaceAndReleaseStream(stream.Detach(), iid, ppv);
}

void GetTheInterface(REFIID iid, void** ppv)
{
  _com_ptr_t<IStream> stream;
  GetTheStream(&stream);
  CoGetInterfaceAndReleaseStream(stream.Detach(), iid, ppv);
}

void GetTheInterface(REFIID iid, void** ppv)
{
  IStream* rawStream;
  GetTheStream(&rawStream);
  std::unique_ptr<IStream, Releaser> stream(rawStream);
  CoGetInterfaceAndReleaseStream(stream.release(), iid, ppv);
}
```

Another way to fix this is to simply stop using `CoGetInterfaceAndReleaseStream` with smart pointers, because the function was designed for dumb pointers. For smart pointers, use `CoUnmarshalInterface`.

```
void GetTheInterface(REFIID iid, void** ppv)
{
  Microsoft::WRL::ComPtr<IStream> stream;
  GetTheStream(&stream);
  CoUnmarshalInterface(stream.Get(), iid, ppv);
}

void GetTheInterface(REFIID iid, void** ppv)
{
  ATL::CComPtr<IStream> stream;
  GetTheStream(&stream);
  CoUnmarshalInterface(stream, iid, ppv);
}

void GetTheInterface(REFIID iid, void** ppv)
{
  _com_ptr_t<IStream> stream;
  GetTheStream(&stream);
  CoUnmarshalInterface(stream, iid, ppv);
}

void GetTheInterface(REFIID iid, void** ppv)
{
  IStream* rawStream;
  GetTheStream(&rawStream);
  std::unique_ptr<IStream, Releaser> stream(rawStream);
  CoUnmarshalInterface(stream.get(), iid, ppv);
}
```

Raymond Chen

Follow