Why do I get a winrt::hresult_no_interface exception when I try to register an auto-revoke event handler?

devblogs.microsoft.com/oldnewthing/20191016-00

October 16, 2019

Raymond Chen

If you register an event handler in C++/WinRT, the registration function returns a `winrt::event_token`, and you can pass that event token back to the event in order to unregister the handler.

An alternative that is popular in some circles is to register with `winrt::auto_revoke` as a marker first parameter ahead of the usual event handler parameters. If you do this, then the registration function returns a `winrt::event_revoker` instead of a `winrt::event_token`. The `event_revoker` automatically unregisters the event at destruction.

What does it mean when attempting to register an auto-revoke event handler throws the `hresult_no_interface` exception?

Internally, the `event_revoker` is a class with three members:

- A weak reference to the event source.
- A method to call to unregister the event.
- The `event_token` to unregister with.

The `hresult_no_interface` exception can come out of the first part: Creating the weak reference. You get the exception if the event source doesn't support weak references. (This is common in the `Windows.UI.Composition` namespace.) In that case, you cannot use the auto-revoke feature will have to fall back to revoking your event handlers manually.

Raymond Chen

Follow