How can I tell the WIL RETURN_IF_FAILED macro that some errors are ignorable?

devblogs.microsoft.com/oldnewthing/20220414-00

April 14, 2022

Raymond Chen

The WIL `RETURN_IF_FAILED` macro evaluates its argument, and if it is a COM failure `HRESULT`, then it returns immediately from the function with that `HRESULT`. There is also a `RETURN_IF_FAILED_EXPECTED` macro that behaves the same, except that any failures are not recorded in the error log. The `EXPECTED` version is for the case where a function fails, but you don't want it cluttering your error log.

But one case that's not covered is "Call this function, and return immediately if the call fails, except that certain errors should be ignored and allow execution to continue." How do you do that?

You can build a filter function that you send the result through, and the filter function can convert all the harmless errors into `S_OK`.

```
HRESULT AllowPending(HRESULT hr)
{
    if (hr == E_PENDING) return S_OK;
    return hr;
}
```

You can then combine this with `RETURN_IF_FAILED`:

```
    RETURN_IF_FAILED(AllowPending(GetItem()));
```

You can even generalize this to allow the list of allowed errors to be passed as a parameter:

```
HRESULT IgnoreSpecificErrors(
    HRESULT hr,
    std::initializer_list<HRESULT> ignored)
{
    for (auto candidate : ignored) {
        if (hr == candidate) return S_OK;
    }
    return hr;
}


    RETURN_IF_FAILED(IgnoreSpecificErrors(GetItem(), { E_PENDING }));
```

You can also create sets of ignorable errors:

```
constexpr HRESULT rpc_errors[] {
    RPC_E_DISCONNECTED,
    RPC_E_SERVER_DIED,
    RPC_E_SERVER_DIED_DNE,
};

    RETURN_IF_FAILED(IgnoreSpecificErrors(GetItem(), rpc_errors));
```

The nice thing about using a filter function is that you can add whatever other features you like.

```
HRESULT IgnoreSpecificErrors(
    HRESULT hr,
    std::initializer_list<HRESULT> ignored)
{
    for (auto candidate : ignored) {
        if (hr == candidate) {
            LOG_HR(hr); // log the failure before transforming it
            return S_OK;
        }
    }
    return hr;
}
```

Raymond Chen

Follow