

How can I cancel the INamespaceWalk::Walk operation?

 devblogs.microsoft.com/oldnewthing/20171109-00

November 9, 2017



Raymond Chen

We saw that you can stop a `INamespaceWalk::Walk` operation by returning a COM error code from the `EnterFolder` or `FoundItem` callback.

So let's do that in order to add a timeout to the namespace walk operation. At each callback, we'll check how much time has elapsed since the operation started, and if it's too long, then we return `HRESULT_FROM_WIN32(ERROR_CANCELLED)`.

```

#define STRICT
#include <windows.h>
#include <shlobj.h>
#include <wrl/client.h>
#include <wrl/implements.h>
#include <stdio.h> // Horrors! Mixing stdio and C++!

namespace wrl = Microsoft::WRL;

class WalkCallback : public wrl::RuntimeClass<
    wrl::RuntimeClassFlags<wrl::ClassicCom>,
    INamespaceWalkCB>
{
public:
    // INamespaceWalkCB
    IFACEMETHODIMP FoundItem(IShellFolder *,
        PCUITEMID_CHILD) override
    { m_itemCount++; return TimeoutStatus(); }

    IFACEMETHODIMP EnterFolder(IShellFolder *,
        PCUITEMID_CHILD) override
    { m_folderCount++; return TimeoutStatus(); }

    IFACEMETHODIMP LeaveFolder(IShellFolder *,
        PCUITEMID_CHILD) override { return S_OK; }

    IFACEMETHODIMP InitializeProgressDialog(PWSTR *ppszTitle,
        PWSTR *ppszCancel) override
    { *ppszTitle = nullptr; *ppszCancel = nullptr;
      return E_NOTIMPL; }

    int ItemCount() const { return m_itemCount; }
    int FolderCount() const { return m_folderCount; }

private:
    bool IsTimedOut()
    { return GetTickCount() - m_startTime > 1000; }

    HRESULT TimeoutStatus()
    { return IsTimedOut() ?
        HRESULT_FROM_WIN32(ERROR_CANCELLED) : S_OK; }

    DWORD m_startTime = GetTickCount();
    int m_itemCount = 0;
    int m_folderCount = 0;
};

```

Our callback object implements `INamespaceWalkCB` by tallying the number of items and folders it encounters. When we find an item or folder, we increment the appropriate counter and check whether we have reached the timeout. If so, we return `HRESULT_FROM_WIN32(ERROR_CANCELLED)` to stop the operation.

Let's take it for a spin.

```
int __cdecl wmain(int argc, PWSTR argv[])
{
    CCoInitialize coinit;

    wr1::ComPtr<INamespaceWalk> walk;
    CoCreateInstance(CLSID_NamespaceWalker, nullptr,
        CLSCTX_INPROC_SERVER, IID_PPV_ARGS(&walk));

    wr1::ComPtr<IShellItem> root;
    SHCreateItemFromParsingName(argv[1], nullptr,
        IID_PPV_ARGS(&root));

    auto callback = wr1::Make<WalkCallback>();

    HRESULT hr = walk->Walk(root.Get(), NSWF_DEFAULT,
        100, callback.Get());

    printf("Walk completed with result 0x%08x\n", hr);
    printf("Found %d items and %d folders\n",
        callback->ItemCount(), callback->FolderCount());

    return 0;
}
```

When I ran this on a small directory tree, I got

```
Walk completed with result 0x00000000
Found 43 items and 6 folders
```

The walk completed in less than one second, so the walk operation completed with `S_OK`.

I repeated the exercise on `C:\Users` and got

```
Walk completed with result 0x80070005
Found 0 items and 2 folders
```

The walk operation encountered an `E_ACCESS_DENIED` error before one second elapsed.

Next, I tried it with my own home directory.

```
Walk completed with result 0x800704c7
Found 3940 items and 2990 folders
```

It found 3940 items and 2990 folders before it ran out of time.

Next time, we'll dig a little bit deeper into cancellation.

Raymond Chen

Follow

