

Enumerating all the programs that can launch a particular protocol

devblogs.microsoft.com/oldnewthing/20151207-00

December 7, 2015



Raymond Chen

Today's Little Program is a variation on the one from last time. This time, instead of enumerating all the handlers for a file extension, we enumerate all the handlers for a protocol. We then open a Web site with that chosen program.

Start with the program from last time and make these changes:

```
std::vector<CComPtr<IAssocHandler>> LoadHandlers(
    PCWSTR extension)
    // ASSOC_FILTER filter
)
{
    std::vector<CComPtr<IAssocHandler>> handlers;
    CComPtr<IEnumAssocHandlers> enumerator;
    SHAssocEnumHandlersForProtocolByApplication(
        L"http", IID_PPV_ARGS(&enumerator));
    for (CComPtr<IAssocHandler> handler;
        enumerator->Next(1, &handler, nullptr) == S_OK;
        handler.Release()) {
        handlers.push_back(handler);
    }
    return handlers;
}
```

The `SHAssocEnumHandlersForProtocolByApplication` function does not have a filter option, so we delete that parameter from `LoadHandlers`.

The `ChooseHandler` function is unchanged.

```
int __cdecl main(int, char**)
{
    CCoInitialize init;
    ProcessReference ref;

    auto handlers = LoadHandlers(L"http");
    auto selection = ChooseHandler(handlers, false);

    if (selection < handlers.size()) {
        CComPtr<IDataObject> dobj;
        GetUIObjectOfFile(nullptr, L"http://www.microsoft.com/",
            IID_PPV_ARGS(&dobj));
        handlers[selection]->Invoke(dobj);
    }
    return 0;
}
```

This version is shorter because there is no filter option, so we just load up all the handlers, pick one, and invoke it.

Notice that we are using `GetUIObjectOfFile` for something that isn't a file. If you go back to that function, you'll see that there's nothing in it that actually requires a file. It can accept any parseable name. A more accurate name for the function would have been `GetUIObject-OfParsingName`, but it's too late now.

Raymond Chen

Follow

