

Manipulating the zone identifier to specify where a file was download from

 devblogs.microsoft.com/oldnewthing/20131104-00

November 4, 2013



Raymond Chen

When you download a file via Internet Explorer, the file is tagged with a little bit of information known as a *zone identifier* which remembers where the file was downloaded from. This is what tells Explorer to put up the “Yo, did you really want to run this program?” prompt and which is taken into account by applications so that they can do things like disable scripting and macros when they open the document, just in case the file is malicious.

Today’s Little Program is really three Little Programs: One to read the zone identifier, one to set the zone identifier, and one to clear it.

```
#define STRICT
#include <windows.h>
#include <atlbase.h>
#include <urlmon.h>
#include <stdlib.h>
int __cdecl wmain(int argc, wchar_t **argv)
{
    if (argc < 2) return 0;
    CCoInitialize init;
    CComPtr<IZoneIdentifier> spzi;
    spzi.CoCreateInstance(CLSID_PersistentZoneIdentifier);
    DWORD dwZone;
    if (SUCCEEDED(CComQIPtr<IPersistFile>(spzi)
        ->Load(argv[1], STGM_READ)) &&
        SUCCEEDED(spzi->GetId(&dwZone))) {
        printf("Zone identifier is %d\n", dwZone);
    } else {
        printf("Couldn't get zone identifier (perhaps there isn't one)\n");
    }
    return 0;
}
```

The first program takes a file name on the command line (fully-qualified path, please) and prints the zone identifier associated with it. The numeric values for the most commonly-encountered zone identifiers are

Identifier	Value
URLZONE_LOCAL_MACHINE	0
URLZONE_INTRANET	1
URLZONE_TRUSTED	2
URLZONE_INTERNET	3
URLZONE_UNTRUSTED	4

Note also that if you want your application to be sensitive to the file zone (so that you can disable features for untrusted documents), you should use the `IInternetSecurityManager::MapUrlToZone` function rather than using only the file zone identifier, because the effective zone of a file is a combination of the file's declared zone as well as its physical location. (For example, a file in the Temporary Internet Files directory or on an untrusted server should not be given full trust regardless of what it claims. [Additional reading.](#))

Here's a program that uses `IInternetSecurityManager::MapUrlToZone` to determine the effective security zone:

```
#define STRICT
#include <windows.h>
#include <atlbase.h>
#include <urlmon.h>
#include <stdlib.h>
int __cdecl wmain(int argc, wchar_t **argv)
{
    if (argc < 2) return 0;
    CCoInitialize init;
    CComPtr<IInternetSecurityManager> spism;
    spzi.CoCreateInstance(CLSID_InternetSecurityManager);
    DWORD dwZone;
    if (SUCCEEDED(spism->MapUrlToZone(
        argv[1],
        &dwZone,
        MUTZ_ISFILE | MUTZ_DONT_UNESCAPE))) {
        printf("Zone is %d\n", dwZone);
    } else {
        printf("Couldn't get zone\n");
    }
    return 0;
}
```

The `MUTZ_ISFILE` flag saves you the hassle of having to prepend `file:` in front of the path, but you still have to pass a full path because the first parameter is a URL, not a path.

Okay, that was a bit of a digression there. Let's write another Little Program which changes the zone identifier.

```
#define STRICT
#include <windows.h>
#include <atlbase.h>
#include <urlmon.h>
#include <stdlib.h>
int __cdecl wmain(int argc, wchar_t **argv)
{
    if (argc < 3) return 0;
    CCoInitialize init;
    CComPtr<IZoneIdentifier> spzi;
    spzi.CoCreateInstance(CLSID_PersistentZoneIdentifier);
    spzi->SetId(_wtol(argv[2]));
    CComQIPtr<IPersistFile>(spzi)->Save(argv[1], TRUE);
    return 0;
}
```

This program takes two parameters: A fully-qualified path and a zone (in integer form). It applies the zone to the file, overwriting the existing zone if any.

Finally, here's a Little Program to remove the zone information from the file entirely. This is the equivalent of clicking the *Unblock* button in the file property sheet.

```
#define STRICT
#include <windows.h>
#include <atlbase.h>
#include <urlmon.h>
#include <stdlib.h>
int __cdecl wmain(int argc, wchar_t **argv)
{
    if (argc < 2) return 0;
    CCoInitialize init;
    CComPtr<IZoneIdentifier> spzi;
    spzi.CoCreateInstance(CLSID_PersistentZoneIdentifier);
    spzi->Remove();
    CComQIPtr<IPersistFile>(spzi)->Save(argv[1], TRUE);
    return 0;
}
```

[Raymond Chen](#)

Follow

