

How can I tell whether two programs will share drive letter mappings?



A customer wanted to know how to determine whether two programs will share drive letter mappings. If one program creates a drive letter mapping, say via `WNetAddConnection3`, will it be visible to the other program?

This information is documented under the topic “[Defining an MS-DOS Device Name](#)”. There are two types of drive letter mappings: Global and Local. Only processes running under the LocalSystem account can manipulate Global drive letter mappings. Everybody else manipulates the Local mapping that is associated with its token’s authentication ID.

So how do you get the authentication ID? You ask for the `TokenStatistics`.

```
#include <wil/token_helpers.h>
```

```
LUID GetAuthenticationIdForCurrentThread()
{
    wil::get_token_information<TOKEN_STATISTICS>().AuthenticationId;
}
```

```
LUID GetAuthenticationIdForProcess(HANDLE process)
{
    wil::unique_handle token;
    THROW_IF_WIN32_BOOL_FALSE(OpenProcessToken(process, TOKEN_QUERY, token.put()));
    return wil::get_token_information<TOKEN_STATISTICS>
(token.get()).AuthenticationId;
}
```

If you’d rather not use `wil`, or if you just want to see how it works inside:

```

HRESULT GetAuthenticationIdForToken(HANDLE token, LUID* luid)
{
    HRESULT hr;
    *luid = {};

    TOKEN_STATISTICS stats;
    DWORD length;
    if (GetTokenInformation(token, TokenStatistics,
        &stats, sizeof(stats), &length)) {
        *luid = stats.AuthenticationId;
        hr = S_OK;
    } else {
        hr = HRESULT_FROM_WIN32(GetLastError());
    }
    return hr;
}

```

```

HRESULT GetAuthenticationIdForCurrentThread(LUID* luid)
{
    return GetAuthenticationIdForToken(
        GetCurrentThreadEffectiveToken(), luid);
}

```

```

BOOL GetAuthenticationIdForProcess(HANDLE process, LUID* luid)
{
    HRESULT hr;
    *luid = {};

    HANDLE token;
    if (OpenProcessToken(process, TOKEN_QUERY, &token)) {
        hr = GetAuthenticationIdForToken(token, luid);
        CloseHandle(token);
    } else {
        hr = HRESULT_FROM_WIN32(GetLastError());
    }

    return hr;
}

```