

Translating from the C++

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.interop/2007-02/msg00052.html>

- *From:* "Mark Rockman" <mrockman@xxxxxxx>
 - *Date:* Thu, 8 Feb 2007 19:39:49 -0500
-

Much of the Win32 API has yet to become natively supported in the dotnet Framework. So often we come across sample code that only works in C++. What is the standard approach to making these calls from C#? Do I write a DLL and decorate its call appropriately (and what would that entail) in C#? I took that approach without luck. Do I use the interop namespace and recode using IntPtr and such? What, exactly, is the translation of an HRESULT in that case?

Is there a one-stop place for the deep down dirty direct way to make these calls from C# without frustration?

I'd like to recode the following in C#, but I don't know how. Any help would be appreciated.

```
// The following example can be used to enable or disable the
// backup privilege. By making the indicated substitutions, you can
// also use this example to enable or disable the restore privilege
// Use the following statement to enable the privilege:
// hr = ModifyPrivilege(SE_BACKUP_NAME, TRUE);
// Use the following statement to disable the privilege:
// hr = ModifyPrivilege(SE_BACKUP_NAME, FALSE);
// Use SE_RESTORE_NAME for the restore privilege.
// The main function in this example enables the backup privilege.
```

```
#include <windows.h>
#include <stdio.h>
```

Translating from the C++

```
HRESULT ModifyPrivilege(
    IN LPCTSTR szPrivilege,
    IN BOOL fEnable)
{
    HRESULT hr = S_OK;
    TOKEN_PRIVILEGES NewState;
    LUID luid;
    HANDLE hToken = NULL;

    // Open the process token for this process.
    if (!OpenProcessToken(GetCurrentProcess(),
        TOKEN_ADJUST_PRIVILEGES | TOKEN_QUERY,
        &hToken ))
    {
        printf("Failed OpenProcessToken\n");
        return ERROR_FUNCTION_FAILED;
    }

    // Get the local unique ID for the privilege.
    if ( !LookupPrivilegeValue( NULL,
        szPrivilege,
        &luid ))
    {
        CloseHandle( hToken );
```

Translating from the C++

```
printf("Failed LookupPrivilegeValue\n");  
return ERROR_FUNCTION_FAILED;  
}
```

```
// Assign values to the TOKEN_PRIVILEGE structure.
```

```
NewState.PrivilegeCount = 1;  
NewState.Privileges[0].Luid = luid;  
NewState.Privileges[0].Attributes =  
(fEnable ? SE_PRIVILEGE_ENABLED : 0);
```

```
// Adjust the token privilege.
```

```
if (!AdjustTokenPrivileges(hToken,  
FALSE,  
&NewState,  
0,  
NULL,  
NULL))  
{  
printf("Failed AdjustTokenPrivileges\n");  
hr = ERROR_FUNCTION_FAILED;  
}
```

```
// Close the handle.
```

```
CloseHandle(hToken);
```

Translating from the C++

```
return hr;
```

```
}
```

```
void main(void)
```

```
{
```

```
    HRESULT hr;
```

```
    hr = ModifyPrivilege(SE_BACKUP_NAME, TRUE);
```

```
    if (!SUCCEEDED(hr))
```

```
        printf("\nFailed to modify privilege.\n");
```

```
    else
```

```
        printf("\nSuccessfully modified privilege.\n");
```

```
}
```

```
.
```