



{ D }

Microsoft Windows Networking API



Visual C# 2008

雖然 Microsoft .Net Framework 提供不少的 Class Library，但在網路資源應用程式上，並無相關的 Class Library，因此必須使用 Windows API 處理。Microsoft Windows 提供了一組 API 支援網路資源應用，稱為 Microsoft Windows Networking API，簡稱為 WNet。

WNet API 函數，主要是透過 mpr.dll 提供，在 C# 中，需使用 DllImport 呼叫其 mpr.dll 檔案，並以 static 與 extern 修飾詞 (Modifier) 宣告，例如：

```
[DllImport("mpr.dll")]  
static extern [回傳型態] [API 函數名稱] (...);
```

WNet API 提供以下之函數功能：

- WNetAddConnection：建立網路資源連線。
- WNetAddConnection2：建立網路資源連線。
- WNetAddConnection3：建立網路資源連線。
- WNetCancelConnection：結束網路連線。
- WNetCancelConnection2：結束網路連線。
- WNetCloseEnum：結束網路資源列舉。
- WNetConnectionDialog：開啓連線對話盒，以建立網路資源的連線。
- WnetConnectionDialog1：開啓連線對話盒，以建立網路資源的連線。
- WNetDisconnectDialog：開啓中斷連線對話盒，以中斷網路資源的連線。
- WNetDisconnectDialog1：開啓中斷連線對話盒，以中斷網路資源的連線。
- WNetEnumResource：列舉網路資源。
- WNetGetConnection：取得本機或連線資源的網路名稱。
- WNetGetLastError：取得網路錯誤訊息。
- WNetGetNetworkInformation：取得網路中 Network Provider 的資訊。
- WNetGetProviderName：取得網路中 Network Provider 的名稱。
- WNetGetResourceInformation：取得網路中提供網路資源的 Network Provider 資訊。
- WNetGetResourceParent：取得網路中提供網路資源的 Network Provider。
- WNetGetUniversalName：取得網路中一個文件的遠程名稱及 UNC (Universal Naming Convention) 名稱。
- WNetGetUser：取得用以連接網路資源的使用者名稱。
- WNetOpenEnum：啓動網路資源列舉。

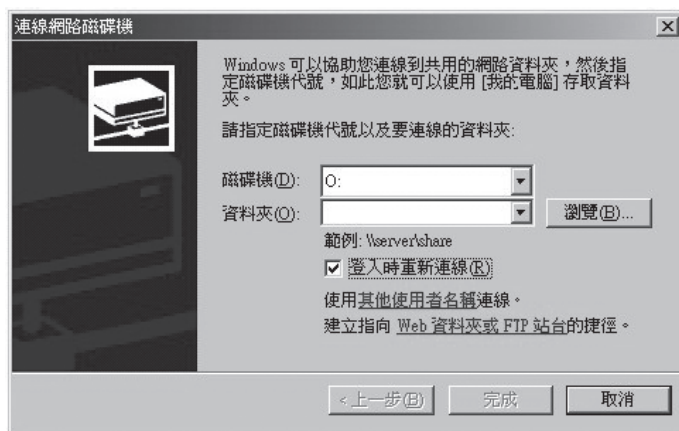
- WNetUseConnection：建立網路資源連線。
- MultinetGetConnectionPerformance：取得網路效能。

透過 Microsoft Windows Networking API，應用程式可以：

- 瀏覽網路中可使用的網路資源。
- 建立資源連線。
- 結束資源連線。
- 取得網資源的相關資訊，如使用者名稱、Network Provider 資訊等。

► 連線網路磁碟機 / 瀏覽印表機連接埠

在 Microsoft Windows 作業系統中，欲使用網路中所共用的目錄或檔案，可使用檔案總管中的 " 連線網路磁碟機 " 的功能來達成。



欲開啓連線網路磁碟機對話盒，最簡單的方法是使用 WNetAPI 所提供的 WNetConnectionDialog 函數，其宣告方式：

```
[DllImport("mpr.dll")]
static extern int WNetConnectionDialog(int hwnd, int dwType);
```

其中參數：

1. hwnd：連線網路磁碟機對話盒之父視窗的控制代碼（Handle to a Window）。
2. dwType：為以下之設定值：
 - RESOURCETYPE_DISK：瀏覽磁碟機資源。
 - RESOURCETYPE_PRINTER：瀏覽印表機資源。

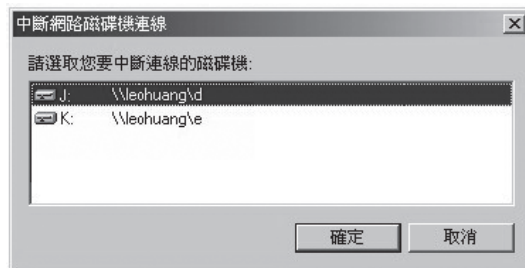
其回傳值若為 0 表示成功。如使用者取消操作，則回傳 -1。如錯誤訊息為 ERROR_EXTENDED_ERROR，可使用 WNetGetLastError 取得額外的錯誤訊息，如：

- ERROR_EXTENDED_ERROR：Network error。
- ERROR_INVALID_PASSWORD：Invalid password。
- ERROR_NO_NETWORK：No network。
- ERROR_NOT_ENOUGH_MEMORY：Insufficient memory。

WNetConnectionDialog 除了可開啓連線網路磁碟機對話盒之外，同時也支援瀏覽印表機連接埠，僅需將 WNetConnectionDialog 的 dwType 參數設為 RESOURCETYPE_PRINTER 則可。

► 中斷網路磁碟機 / 印表機

與連線網路 WNetConnectionDialog 相對應的，則是 WNetDisconnectDialog 函數，利用此函數，可出現如下圖之對話盒，以便使用者中斷網路磁碟機之連線。



WNetDisconnectDialog 宣告方式如下：

```
[DllImport("mpr.dll")]  
static extern int WNetDisconnectDialog(int hwnd, int dwType);
```

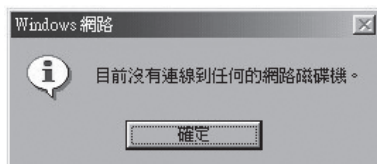
其中參數：

1. `hwnd`：連線網路磁碟機對話盒之父視窗的控制代碼（**Handle to a Window**）。
2. `dwType`：設成 `RESOURCE_TYPE_DISK` 或 `RESOURCE_TYPE_PRINTER`，決定要中斷的是磁碟機或是印表機資源。

其回傳值若為 0 表示成功。如使用者取消操作，則回傳 -1。如錯誤訊息為 `ERROR_EXTENDED_ERROR`，可使用 `WNetGetLastError` 取得額外的錯誤訊息，如：

- `ERROR_EXTENDED_ERROR`：Network error。
- `ERROR_NO_NETWORK`：No network。
- `ERROR_NOT_ENOUGH_MEMORY`：Insufficient memory。

當執行 `WNetDisconnectDialog` 時，若設定 `dwType` 參數為 `RESOURCE_TYPE_DISK`，則出現中斷網路磁碟機對話盒，使用者可選取對應之磁碟機名稱中斷。若目前並沒有任何對應之網路磁碟機連線，則 `WNetDisconnectDialog` 函數會回傳如下之畫面：



欲中斷網路印表機，則設定 `dwType` 參數為 `RESOURCE_TYPE_PRINTER`。

► 建立網路資源連線

在前幾節所提到的 `WNetConnectionDialog` 函數，在執行階段時會出現相對應的對話盒，以便使用者選擇網路資源連結。

另外，`WNet` API 提供了一組 API，用以建立網路資源連線：

- `WNetAddConnection`
- `WNetAddConnection2`
- `WNetAddConnection3`

這三個 API 都是用以建立網路源連線，其宣告方式大同小異。

WNetAddConnection 宣告方式：

```
[DllImport("mpr.dll", EntryPoint="WNetAddConnection")]
static extern int WNetAddConnectionA(string lpszNetPath,
    string lpszPassword, string lpszLocalName)
```

其中參數：

1. lpszNetPath：設定欲連線的網路名稱，為 string 型別。
2. lpPassword：設定密碼，為 string 型別。如為 null，表示採用目前使用者的密碼。如為一個空字串，則不用任何密碼。
3. lpszLocalName：磁碟機代號，為 string 型別，例如：G: or LPT1:。

其回傳值若為 0 表示成功。如使用者取消操作，則回傳 -1。如錯誤訊息為 ERROR_EXTENDED_ERROR，可使用 WNetGetLastError 取得額外的錯誤訊息。

WNetAddConnection 共有三個參數，分別代表：

- 欲連線的共享路徑 (lpszNetPath)：例如筆者網路中有一 Server 名為 Athena，其中 D 為其共享名稱，則其共享路徑為 \\Athena\D。
- 密碼 (lpPassword)：為登錄共享路徑所須之密碼。
- 磁碟機代碼 (lpszLocalName)：則本機顯示共享路徑之代碼，可為磁碟機名稱或印表機連接埠，例如：F: 或 LTP1。

例如欲連結網路中名為 Athena 的 Server 所共享之資源 "D"，並以磁碟機代碼 "F:" 代表，若 WNetAddConnection 執行成功，則會在檔案總管中建立其相對應的連線網路磁機名稱 "D 於 'Athena' (F)"。

相對於 WNetAddConnection 函數，WNetAddConnection2 函數則更為複雜且完整，並適用各類 Microsoft Windows 系統，主要的差異在於 WNetAddConnection2 需定義：

- 欲連線的共享路徑名稱。
- 磁碟機代碼。
- 使用者登錄名稱。
- 密碼。
- 連線資源型態，如磁碟機、印表機等。

- 網路 Provider 名稱，如 Microsoft Windows Network。

WNetAddConnection2 宣告方式：

```
[DllImport("mpr.dll", EntryPoint="WNetAddConnection2")]
static extern int WNetAddConnection2A(ref NETRESOURCE lpNetResource,
    string lpPassword, string lpUsername, int dwFlags)
```

其中參數：

1. lpNetResource：定義連線網路資源，為 NETRESOURCE 結構型別。
2. lpPassword：設定密碼，為 string 型別。如為 null，表示採用目前使用者的密碼。如為一個空字串，則不用任何密碼。
3. lpUserName：設連線的使用者名稱，為 string 型別。如為 null，表示使用目前使用者。
4. dwFlags：為常數 CONNECT_UPDATE_PROFILE 表示建立永久性連線。

其中 NETRESOURCE 結構之宣告為：

```
[StructLayout(LayoutKind.Sequential)]
public class NETRESOURCE{
    public int dwScope;
    public int dwType;
    public int dwDisplayType;
    public int dwUsage;
    public string lpLocalName;
    public string lpRemoteName;
    public string lpComment;
    public string lpProvider;
}
```

其中：

1. dwScope：代表連線範圍，可為以下之參數：
 - RESOURCE_CONNECTED：已連線之資源（&H1）。

- RESOURCE_GLOBALNET：所有可連線之資源（&H2）。
 - RESOURCE_REMEMBERED：曾經被連線之資源（&H3）。
2. dwType：代表連線資源為磁碟機或印表機，可為以下之參數：
- RESOURCETYPE_ANY：所有資源（&H0）。
 - RESOURCETYPE_DISK：磁碟機（&H1）。
 - RESOURCETYPE_PRINT：印表機（&H2）。
3. dwDisplayType：為以下之參數：
- RESOURCEDISPLAYTYPE_GENERIC：任意（&H0）。
 - RESOURCEDISPLAYTYPE_DOMAIN：Domain（&H1）。
 - RESOURCEDISPLAYTYPE_SERVER：Server（&H2）。
 - RESOURCEDISPLAYTYPE_SHARE：共享（&H3）。
4. dwUsage：為以下之參數：
- RESOURCEUSAGE_CONNECTABLE：可連線之資源（&H1）。
 - RESOURCEUSAGE_CONTAINER：可由 WNetOpenEnum 列舉之網路資源（&H2）。
5. lpLocalName：磁碟機代號，如 F: 或 LTP1。
6. lpRemoteName：欲連線的網路名稱。
7. lpProvider：網路 Provider 名稱。

其回傳值若為 0 表示成功。如使用者取消操作，則回傳 -1。如錯誤訊息為 ERROR_EXTENDED_ERROR，可使用 WNetGetLastError 取得額外的錯誤訊息。

使用 WNetAddConnection2 函數建立網路資源連線，需定義 NETRESOURCE，分別為 dwScope、dwType、dwDisplayType、dwUsage、lpRemoteName（欲連線網路資源名稱）及 lpLocalName（磁碟機代碼）。

若 WNetAddConnection2 執行成功，則會在檔案總管中建立其相對應的連線網路磁機名稱 "D 於 'Athena' (F)"。

除了 `WNetAddConnection` 與 `WNetAddConnection2` 之外，`WNet` 另外提供 `WNetAddConnection3` 建立網路資源連線。

```
[DllImport("mpr.dll", EntryPoint="WNetAddConnection3")]
static extern int WNetAddConnection3A(int hwnd,
    ref NETRESOURCE lpNetResource, string lpPassword,
    string lpUsername, int dwFlags)
```

其中參數：

1. `hwnd`：連線網路磁碟機對話盒之父視窗的控制代碼（Handle to a Window）。
2. `lpNetResource`：定義連線網路資源，為 `NETRESOURCE` 結構型別。
3. `lpPassword`：設定密碼，為 `string` 型別。如為 `null`，表示採用目前使用者的密碼。如為一個空字串，則不用任何密碼。
4. `lpUserName`：設連線的使用者名稱，為 `string` 型別。如為 `null`，表示使用當前使用者。
5. `dwFlags`：為常數 `CONNECT_UPDATE_PROFILE` 表示建立永久性連線。

其回傳值若為 0 表示成功。如使用者取消操作，則回傳 -1。如錯誤訊息為 `ERROR_EXTENDED_ERROR`，可使用 `WNetGetLastError` 取得額外的錯誤訊息。

基本上 `WNetAddConnection3` 與 `WNetAddConnection2` 所定義的參數和構變數均相同，唯一的不同點在於 `WNetAddConnection3`，多定義了 `hwnd` 參數，此參數用以定義開啓對話盒之父視窗的控制代碼，若此參數設為 `null`，則 `WNetAddConnection3` 與 `WNetAddConnection2` 完全相同。

► 中斷網路資源連線

在 Microsoft Windows 系統中，欲中斷某一網路資源連線，可利用檔案總管在連線資源上按下滑鼠右鍵，當出現 Popup Menu 選單時，選取 "中斷 (D)" 則可完成中斷網路連線的程序。

相對於前幾節所介紹的連線功能，WNet 中用以中斷網路資源連線的 API 有：

- WNetCancelConnection
- WNetCancelConnection2

此 API 與之前所介紹的 WNetDisconnectDialog 雷同，均為中斷資源連線之用，不同的是 WNetDisconnectDialog 會出現【中斷網路資源】對話盒，而 WNetCancelConnection 與 WNetCancelConnection2，則直接中斷所指定之磁碟機代號。

WNetCancelConnection 宣告方式：

```
[DllImport("mpr.dll", EntryPoint="WNetCancelConnection")]  
static extern int WNetCancelConnectionA(string lpName, int fForce)
```

其中參數：

1. lpName：設定已連線資源的遠程名稱或本地名稱，為 string 型別。
2. fForce：如為 true，即使連線的資源上有正在打開的文件或作業，則依然中斷連線。如為 false，則若連線的資源上有正在打開的文件或作業，則中斷連線失敗。

其回傳值若為 0 表示成功。如使用者取消操作，則回傳 -1。如錯誤訊息為 ERROR_EXTENDED_ERROR，可使用 WNetGetLastError 取得額外的錯誤訊息。

其中 fForce 參數，若設定為 true，則不論所連線之資源是否有被開啓的檔案、目錄或作業 WNetCancelConnection 均會中斷其連線作業。反之，若 fForce 參數設為 false，則若有被開啓的檔案、目錄等，則中斷作業將失敗。

除了 WNetCancelConnection 之外，WNet 另有 WNetCancelConnection2 支援中斷網路連線。WNetCancelConnection2 與 WNetCancelConnection 雷同，除了之前的 lpName 及 fForce 參數之外，另加入 dwFlags 參數，為常數 CONNECT_UPDATE_PROFILE，表示結束永久性連線，此參數亦為 WNetCancelConnection2 中所介紹的 dwFlags 參數。

WNetCancelConnection2 的宣告方式：

```
[DllImport("mpr.dll", EntryPoint="WNetCancelConnection2")]  
static extern int WNetCancelConnection2A(string lpName,  
int dwFlags, int fForce)
```

假設目前有已連線之 "F:" 磁碟機，則只需將之傳入 `WNetCancelConnection2` 的第一個參數 `lpszName` 則可，若將 `fForce` 參數設為 `true`，當被連線的資源中若有被開啓的檔案或目錄等時，`WNetCancelConnection2` 仍會中斷此連線資源。

► 取得連線資源名稱

在 Microsoft 網路中，欲表達某一網路資源名稱，是以下列方式表示：

`\\<Server>\< 共享資源名稱 >`

而在 Microsoft Windows 系統中，當成功的連結至網路上的資源後，檔案總管中會顯示其連結資源名稱，並以下列方式表示：

" 共享資源名稱 " 於 'Server' (本機磁碟機名稱)

例如 "D 於 'Athena' (F)"，則表示本機已連接上名為 Athena 伺服器上的 D 共享資源，並以本機 "F:" 磁碟機表示之。

若想取得此連線資源名稱，可使用 WNet 的 `WNetGetConnection`。

```
[DllImport("mpr.dll", EntryPoint="WNetGetConnection")]
static extern int WNetGetConnectionA(string lpszLocalName,
    string lpszRemoteName, ref int cbRemoteName)
```

其中參數：

1. `lpszLocalName`：磁碟機代號，例如：G: or LPT1:。
2. `lpszRemoteName`：設定字串 `buffer`，用於儲存 `lpszLocalName` 參數所回傳的連線資源名稱。
3. `cbRemoteName`：設定 `lpszRemoteName` 參數的長度。

其中第二個參數 `lpszRemoteName` 用以儲存 `lpszLocalName` 參數所回傳的連線資源名稱，假設 "F:" 磁碟機為一連線資源，其相對應的名稱為 "\\Athena\D"，則僅需設定其 `lpszLocalName` 為 F:，則 `WNetGetConnection` 會回傳 `lpszRemoteName` 參數為 "\\Athena\D" 字串。

► 取得網路連線用者名稱

在 Microsoft Windows 網路環境中，不論登入本機或網路上所共享的資源，均需使用者名稱以茲登入，欲取得此使用者名稱，可使用 `WNetGetUser` 函數取得用以連接網路資源的使用者名稱。

```
[DllImport("mpr.dll", EntryPoint="WNetGetUser")]  
static extern int WNetGetUserA(string lpName, string lpUserName,  
    ref int lpnLength)
```

其中參數：

1. `lpName`：設定磁碟機代號，例如 F:。若為 `null`，則取得本機的使用者名稱。
2. `lpUserName`：指定字串 `buffer`，用於取得回傳連接網路資源的使用者名稱。
3. `lpnLength`：`lpUserName` 參數的長度。

`WNetGetUser` 有兩種功能：

- 取得用以連接網路資源的使用者名稱。
- 取得本機登入網路的使用者名稱。

其主要差別在於 `WNetGetUser` 的第一個參數 `lpName` 的設定方式，若指定磁碟機代碼，則回傳用以連接網路資源的使用者名稱，反之，若設定為 `null`，則回傳本機登入網路的使用者名稱。例如某一網路名為 `domain`，欲連接至其所開放的資源之使用者名稱為 `Athena`，則 `WNetGetUser` 函數將取得 `domain\Athena` 的字串。另外，若本機登入之使用者名稱為 `athena`，則回傳之字串為 `athena`，為本機登入網路的使用者名稱。

✳ 【參考資料】

[1] Microsoft Developer Network.