



{ E }

Microsoft Win32 Internet API



Visual C# 2008

Microsoft .Net Framework 在支援 Internet 網際網路應用上，提供以下的 Class Library：

- System.Net.HttpListener
- System.Net.HttpWebRequest
- System.Net.HttpWebResponse
- System.Net.FtpWebRequest
- System.Net.FtpWebResponse
- System.Net.Sockets.Socket
- System.Net.Sockets.TcpClient
- System.Net.Sockets.TcpListener

除此之外，Microsoft 尚提供 Win32 Internet API 函數支援，此類型的 API 是由 wininet.dll 動態連結函式庫所提供，因此在 C# 中，需使用 DllImport 呼叫其 wininet.dll 檔案，例如：

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

Win32 Internet API 函數主要支援網際網路之應用，主要可分為：

- 支援 FTP 通訊協定。
- 支援 HTTP 通訊協定。
- 支援 Gopher 通訊協定。
- 支援電話撥接網路應用。
- 支援網際網路連線及偵測網路連線。
- 支援 Cookie 功能。

► 支援通訊協定

不論是 FTP、HTTP 或 Gopher 通訊協定，在應用 Win32 Internet API 的開發上皆十分相似，其流程主要分為：

1. 建立 Internet Session：不論 FTP、HTTP 或是 Gopher 通訊協定，欲使用 WININET 開發應用程式，其第一步驟則是建立 Internet Session，其實就是以 InternetOpen 函數建立 Internet Session Handle，WININET 應用程式依據此 Handle 值，建立 FTP、HTTP 或 Gopher 服務。

2. 建立 FTP、HTTP 或 Gopher 服務：依據步驟一所產生的 Handle 值，使用 `InternetConnect` 函數建立以下任一服務：

- FTP：以 Port 21 為預設值的 FTP 服務。
- HTTP：以 Port 80 為預設值的 HTTP 服務。
- Gopher：以 Port 70 為預設值的 Gopher 服務。
- HTTPS：以 Port 443 為預設值的 HTTPS（HTTP SSL）服務。
- Sockets Firewall：以 Port 1080 為預設值的 Firewall（防火牆）服務。

若 `InternetConnect` 執行成功，會回傳一 `Internet Service Handle` 值，所有的 FTP、HTTP 或 Gopher 流程將依據此 Handle 為基礎產生。

3. 流程處理：此部分依據 `InternetConnect` 所設定之不同通訊協定服務而有所差異。
4. 關閉通訊協定服務：當流程處理完畢之後，應以 `InternetCloseHandle` 關閉由 `InternetConnect` 所產生之 `Internet Service Handle`，以結束 FTP 或 HTTP 服務，並釋放系統資源。
5. 關閉 Internet Session：最後一步，則是以 `InternetCloseHandle` 關閉由 `InternetOpen` 所產生之 `Internet Session Handle`，以結束整個 Internet Session，至此所有的流程則真正結束。

► 建立 Internet Session

使用 Win32 Internet API 開發 Internet 的應用程式，第一步驟則是建立 Internet Session，可使用 `InternetOpen` 產生 Internet 連結所需的 `Internet Session Handle`。其宣告方式：

```
[DllImport("wininet.dll")]
static extern int InternetOpen(string lpszAgent, int dwAccessType,
    string lpszProxyName, string lpszProxyBypass, int dwFlags);
```

其中參數：

1. `lpszAgent`：設定應用程式名稱（可任意設定）。

2. `dwAccessType`：設定 Internet 連結存取方式，可為以下之任一設定值：
 - `INTERNET_OPEN_TYPE_PRECONFIG`：依【網際網路選項】中之預設 Proxy 伺服器設定為主。
 - `INTERNET_OPEN_TYPE_DIRECT`：不透過 Proxy 伺服器連結。
 - `INTERNET_OPEN_TYPE_PROXY`：透過 Proxy 伺服器連結。
 - `INTERNET_OPEN_TYPE_PRECONFIG_WITH_NO_AUTOPROXY`：依【網際網路選項】中之預設 Proxy 伺服器設定為主，且不使用 Internet Setup file。
3. `lpszProxy`：當 `dwAccessType` 為 `INTERNET_OPEN_TYPE_PROXY` 時，設定其 Proxy 伺服器名稱，若為 `null`，則不透過 Proxy 伺服器。
4. `lpszProxyBypass`：不透過 Proxy 伺服器連結之網址。
5. `dwFlags`：為以下之任一設定值：
 - `INTERNET_FLAG_FROM_CACHE`：由 Cache 中取得 Internet 連結之資訊，如同離線作業。
 - `INTERNET_FLAG_OFFLINE`：離線作業。
 - `INTERNET_FLAG_MAKE_PERSISTENT`：將 Internet 連結之資訊永久性寫入 Cache 中。
 - `INTERNET_FLAG_NO_CACHE_WRITE`：不將 Internet 連結之資訊寫入 Cache 中。
 - `INTERNET_FLAG_ASYNC`：產生非同步的 Internet Connection。

其回傳值為 Internet 連結之 Handle 值，若 Handle 為 0 表示失敗。

在應用程式結束之前，需以 `InternetCloseHandle` 關閉由 `InternetOpen` 所產生之 Handle 值。其宣告方式：

```
[DllImport("wininet.dll")]  
static extern bool InternetCloseHandle(int hInternet);
```

其中參數 `hInternet` 為欲關閉的 Internet 之 Handle 值。其回傳值若為 `true` 表示執行成功。

► 建立 Internet 服務

欲建立 Internet 服務，可利用 Win32 Internet API 所提供的 `InternetConnect` 函數。`InternetConnect` 利用 `InternetOpen` 所產生的 Handle 值建立 FTP、HTTP 或 Gopher 等

服務所需的 **Handle** 值，並且設定此服務的伺服器 IP 位址（或 DNS 名稱）、所需的通訊埠、登錄所需的使用者帳號及密碼等。

若 `InternetConnect` 執行成功，也會如 `InternetOpen` 般回傳 **Handle** 值，但兩者的意義卻不同。`InternetOpen` 所產生的 **Handle** 值，可視為整個 **Internet** 應用程式的 **Handle**，而依據此 **Handle** 值可產生多個不同類型的 **Internet** 服務，而此服務需以 `InternetConnect` 所產生的 **Handle** 代表，以區別不同的服務。

`InternetConnect` 函數的宣告方式：

```
[DllImport("wininet.dll")]
static extern int InternetConnect(int hInternetSession,
    string lpszServerName, int nServerPort,
    string lpszUsername, string lpszPassword,
    int dwService, int dwFlags, int dwContext);
```

其中參數：

1. `hInternetSession`：由 `InternetOpen` 所產生的 **Handle** 值。
2. `lpszServerName`：設定 **Internet** 服務伺服器之 DNS 名稱或 IP Address。
3. `nServerPort`：設定 **Internet** 服務之通訊埠，可為以下服務之預設通訊埠：
 - `INTERNET_DEFAULT_FTP_PORT`：FTP 服務（21）。
 - `INTERNET_DEFAULT_GOPHER_PORT`：Gopher 服務（70）。
 - `INTERNET_DEFAULT_HTTP_PORT`：HTTP 服務（80）。
 - `INTERNET_DEFAULT_HTTPS_PORT`：HTTPS 服務（443）。
 - `INTERNET_DEFAULT_SOCKS_PORT`：Socks Firewall（1080）。
 - `INTERNET_INVALID_PORT_NUMBER`：依據 `dwService` 所設定之 **Internet** 服務類型所使用之預設通訊埠。
4. `lpszUsername`：設定連結此 **Internet** 服務之使用者名稱。若為 `null`，則使用預設值。
5. `lpszPassword`：設定連結此 **Internet** 服務之使用者密碼。若為 `null`，則使用預設值。若為 `null`，則使用預設之帳號及密碼。以 FTP 服務為例，若 `lpszUsername` 及 `lpszPassword` 為 `null`，代表以 "anonymous"（匿名）為其登錄帳號及 "使用者的 e-mail 帳號" 為其登錄密碼。

6. dwService：設定 Internet 服務類型，為以下之任一設定值：

- INTERNET_SERVICE_FTP：FTP 服務。
- INTERNET_SERVICE_GOPHER：Gopher 服務。
- INTERNET_SERVICE_HTTP：HTTP 服務。

7. dwFlags：參數值。

8. dwContext：預留值（0）。

其回傳值為 Internet 服務之 Handle 值，若 Handle 為 0 表示失敗。

欲關閉某 Internet 服務連結，可使用 InternetCloseHandle。

```
[DllImport("wininet.dll")]  
static extern bool InternetCloseHandle(int hInternet);
```

► 下載網頁內容

Win32 Internet API 在支援 HTTP 通訊協定上，主要可分為兩部分：

1. 處理 URL：應用程式可以透過 InternetOpenUrl 和 InternetReadFile 的使用，下載指定 URL 的資訊至用戶端，此類資訊如 htm、html、asp（Active Server Page）、jsp（Java Server Page）等，應用程式如同下載網頁原始檔功能一般。
2. 傳送 HTTP 請求至網站伺服器：如同瀏覽器執行 URL 命令列指令一樣，Win32 Internet API 另一項重要功能則是傳送 HTTP Request 至網站伺服器。

欲使用 Win32 Internet API 開發網頁下載程式，其流程為：

1. 產生 Internet Session：第一步驟是以 InternetOpen 產生應用程式 Internet Session 的 Handle 值。
2. 建立 URL 資源之 Handle：第二步驟則是以 InternetOpenUrl 函數產生欲下載 URL 網頁資源之 Handle，以作為 InternetReadFile 之用。
3. 處理及下載 URL 網頁資源：以 InternetReadFile 讀取 URL 網頁之內容。
4. 關閉由 InternetOpen 及 InternetOpenUrl 所產生的 Handle 以釋放資源。

InternetOpenUrl 的宣告方式：

```
[DllImport("wininet.dll")]
static extern int InternetOpenUrl(int hInternet,
    string lpszUrl, string lpszHeaders, int dwHeadersLength,
    int dwFlags, int dwContext);
```

其中參數：

1. hInternet：由 InternetOpen 所產生的 Handle 值。
2. lpszUrl：設定 URL 位址，需以 http:// 起始。
3. lpszHeaders：設定 URL 的標題資訊。
4. dwHeadersLength：設定 lpszHeaders 參數之大小。
5. dwFlags：參數值。
6. dwContext：0。

InternetReadFile 的宣告方式：

```
[DllImport("wininet.dll")]
static extern bool InternetReadFile(int hFile, string lpBuffer,
    int dwNumberOfBytesToRead, int lpdwNumberOfBytesRead);
```

其中參數：

1. hFile：由 InternetOpenUrl 所產生的 Handle 值。
2. lpBuffer：回傳 URL 之內容。
3. dwNumberOfBytesToRead：設定 lpBuffer 之大小（bytes）。
4. lpdwNumberOfBytesRead：回傳 lpBuffer 之大小（bytes）。

當 InternetOpenUrl 的 Handle 建立成功之後，可利用 InternetReadFile 依序讀取網頁每行的內容，其中 InternetReadFile 的第二個參數 lpBuffer 會回傳網頁每一段落之內容。第四個參數 lpdwNumberOfBytesRead 回傳 lpBuffer 字串的大小長度，因此程式中可藉由此參數判斷網頁是否已至結尾，若回傳 lpBuffer 字串長度為 0，則表示 InternetReadFile 已將所指定之網頁內容讀取至完畢。

► FTP 通訊協定

Win32 Internet API 除了支援 HTTP 通訊協定之外，另一項重要功能則是支援 FTP 通訊協定。一開始需以 `InternetOpen` 建立 `Internet Session` 的 `Handle`，接著以 `InternetConnect` 建立 FTP 服務的 `Handle`。不同的是，`InternetConnect` 的第三個參數 `nServerPort` 及第六個參數 `dwService` 需設定為為 FTP 相關的參數值。

- `nServerPort`：`INTERNET_DEFAULT_FTP_PORT` (=21)。
- `dwService`：`INTERNET_SERVICE_FTP` (=1)。

另外參數 `lpszUsername` 及 `lpszPassword` 代表登錄此 FTP 伺服器所需之使用者帳號及密碼，若兩者皆為空字串，則代表是以 `anonymous` (匿名) 身份登入，登錄帳號則為 `anonymous`，而密碼為使用者的 `e-mail` 帳號。

下表為 `lpszUsername` 及 `lpszPassword` 兩參數的可能組合：

<code>lpszUsername</code>	<code>lpszPassword</code>	使用者名稱	使用者密碼
null	null	anonymous	e-mail
非 null	null	<code>lpszUsername</code> 值	無密碼
null	非 null	錯誤	錯誤
非 null	非 null	<code>lpszUsername</code> 值	<code>lpszPassword</code> 值

► 瀏覽 FTP 伺服器目錄

本節將介紹如何執行 FTP 的 `pwd`、`dir` 及 `ls` 指令，以取得 FTP 伺服器目前的目錄名稱及目錄中內容。

欲建立 FTP 用戶端瀏覽器，第一步驟是與 FTP 伺服器建立連線。當連線建立成功之後，需回傳 FTP 伺服器針對不同使用者所開放的目錄名稱 (可視為其根目錄)，這個步驟可使用 `FtpGetCurrentDirectory` 來完成，此函數如同執行 FTP 的 `pwd` 指令一般，回傳 FTP 伺服器目前的目錄名稱。其宣告方式：

```
[DllImport("wininet.dll")]
static extern bool FtpGetCurrentDirectory(int hFtpSession,
    string lpszCurrentDirectory, int lpdwCurrentDirectory);
```


其中參數：

1. `hFtpSession`：由 `InternetConnect` 所產生的 FTP 服務。
2. `lpszCurrentDirectory`：回傳 FTP 伺服器目前的目錄名稱。
3. `lpdwCurrentDirectory`：`lpszCurrentDirectory` 之大小。

其回傳值若為 `true` 表示執行成功。若 `FtpGetCurrentDirectory` 執行成功，其 `lpszCurrentDirectory` 參數將回傳 FTP 伺服器目前的目錄名稱。

► FTP 下載功能

FTP 瀏覽器除了這些基本功能之外，最重要的就是下載及上傳檔案的功能。在 FTP 的指令中，下載可使用 `get` 或 `mget` 指令，上傳則使用 `put` 或 `mput` 指令。

在 Win32 Internet API 中，如同 FTP 的 `get` 或 `mget` 指令的函數為 `FtpGetFile` 函數，其功能則是將 FTP 伺服器上的檔案下載至 Local 端的目錄中，其宣告方式：

```
[DllImport("wininet.dll")]
static extern bool FtpGetFile(int hFtpSession,
    string lpszRemoteFile, string lpszNewFile,
    bool fFailIfExists, int dwLocalFlagsAndAttributes,
    int dwInternetFlags, int dwContext);
```

其中參數：

1. `hFtpSession`：由 `InternetConnect` 所產生的 FTP 服務 Handle 值。
2. `lpszRemoteFile`：欲下載的 FTP 伺服器的檔案名稱。
3. `lpszNewFile`：下載至 Local 端的檔案名稱。
4. `fFailIfExists`：若 `fFailIfExists` 設定為 `true` 且 `lpszNewFile` 所指定的檔案名稱已存在，則 `FtpGetFile` 會執行失敗。
5. `dwLocalFlagsAndAttributes`：Local 檔案的屬性。
6. `dwContext`：預留值 (0)。
7. `dwInternetFlags`：設定下載的檔案格式是以 ASCII（文字模式）或 Binary（二進位模式）傳輸，可為下列任一設定值：

- FTP_TRANSFER_TYPE_UNKNOWN：預設為 Binary 模式（&H0）。
- FTP_TRANSFER_TYPE_ASCII：ASCII 模式（&H1）。
- FTP_TRANSFER_TYPE_BINARY：Binary 模式（&H2）。
- INTERNET_FLAG_TRANSFER_ASCII：ASCII 模式（&H1）。
- INTERNET_FLAG_TRANSFER_BINARY：Binary 模式（&H2）。

► FTP 上傳功能

欲自用戶端電腦上傳檔案至 FTP 伺服器端，可利用 `FtpPutFile` 函數，基本上與 `FtpGetFile` 類似，宣告方式：

```
[DllImport("wininet.dll")]
static extern bool FtpPutFile(int hFtpSession,
    string lpszLocalFile, string lpszNewRemoteFile,
    int dwInternetFlags, int dwContext);
```

其中參數：

1. `hFtpSession`：由 `InternetConnect` 所產生的 FTP 服務 handle 值。
2. `lpszLocalFile`：欲上傳至 FTP 伺服器端的 Local 端檔案名稱。
3. `lpszNewRemoteFile`：上傳至 FTP 伺服器端的檔案名稱。
4. `dwInternetFlags`：設定下載的檔案格式是以 ASCII（文字模式）或 Binary（二進位模式）傳輸。
5. `lpdwConnection`：回傳值，以代表目前使用 `InternetDial` 函數所指定之撥號連線，藉由此值，可使用 `InternetHangUp` 函數中斷此連線。
6. `dwContext`：預留值（0）。

欲上傳檔案至 FTP 伺服器端，FTP 伺服器端必須開放目錄寫入（Write）的權限，以允許修改 FTP 伺服器端目錄下檔案，否則上傳檔案將執行失敗。

► 新增 FTP 伺服器目錄

欲新增 FTP 目錄，可使用 FTP 指令 `mkdir`，相對應的 Win32 Internet API 函數為 `FtpCreateDirectory`，其宣告方式：

```
[DllImport("wininet.dll")]
static extern bool FtpCreateDirectory(int hFtpSession,
    string lpszDirectory);
```

其中參數：

1. `hFtpSession` 由 `InternetConnect` 所產生的 FTP 服務 handle 值。
2. `lpszDirectory`：欲新增之 FTP 伺服器目錄。

欲建立 FTP 伺服器目錄，FTP 伺服器必須開放目錄寫入（Write）的權限，否則建立伺服器目錄將執行失敗。

► 刪除 FTP 伺服器目錄

欲刪除 FTP 目錄，可使用 FTP 指令 `rmdir`，相對應的 Win32 Internet API 函數為 `FtpRemoveDirectory`，其宣告方式：

```
[DllImport("wininet.dll")]
static extern bool FtpRemoveDirectory(int hFtpSession,
    string lpszDirectory);
```

其中參數：

1. `hFtpSession` 由 `InternetConnect` 所產生的 FTP 服務 handle 值。
2. `lpszDirectory`：欲刪除之 FTP 伺服器目錄。

欲刪除 FTP 伺服器目錄，FTP 伺服器必須開放目錄寫入（Write）的權限，否則刪除伺服器目錄將執行失敗。

► 刪除 FTP 伺服器檔案

欲刪除 FTP 伺服器檔案，可使用 FTP 指令 `delete`，相對應的 Win32 Internet API 函數為 `FtpDeleteFile`，其宣告方式：

```
[DllImport("wininet.dll")]
static extern bool FtpDeleteFile(int hFtpSession,
    string lpszFileName);
```

其中參數：

1. `hFtpSession` 由 `InternetConnect` 所產生的 FTP 服務 handle 值。
2. `lpszFileName`：欲刪除之 FTP 伺服器檔案。

欲刪除 FTP 伺服器檔案，FTP 伺服器必須開放目錄寫入（Write）的權限，否則刪除伺服器檔案將執行失敗。

► 重新命名 FTP 伺服器檔案

欲重新命名 FTP 伺服器檔案，可使用 FTP 指令 `rename`，相對應的 Win32 Internet API 函數為 `FtpRenameFile`。其宣告方式：

```
[DllImport("wininet.dll")]
static extern bool FtpRenameFile(int hFtpSession,
    string lpszExisting, string lpszNew);
```

其中參數：

1. `hFtpSession` 由 `InternetConnect` 所產生的 FTP 服務 handle 值。
2. `lpszExisting`：欲重新命名之 FTP 伺服器檔案。
3. `lpszNew`：新的 FTP 伺服器檔案名稱。

欲重新命名 FTP 伺服器檔案，FTP 伺服器必須開放目錄寫入（Write）的權限，否則重新命名伺服器檔案將執行失敗。

✳ 【參考資料】

[1] Microsoft Developer Network.