

用 libcurl 撰寫 HTTP 存取程式：以 Yahoo! 奇摩字典為範例

Last Update : 09 / 17 / 2007 By DarkRanger .

[前言](#)[在 Dev-C++ 開發環境使用 libcurl 函式庫](#)[Yahoo! 奇摩字典的資料傳送分析](#)[程式碼的撰寫](#)[完整程式碼範例](#)[執行測試](#)[結語](#)[前言](#)

先來談談為什麼會有這篇文章：話說 PCMan 開發了一個名叫 [GNetDict](#)（最早叫做 YKDict）的線上字典查詢工具，使用 python + pygtk + gtkhtml2 開發，而支援的作業系統為 Linux。DR 認為這個工具實在是非常的方便，Windows 上也該要有一個，於是就打算利用 [wxWidgets](#) + [libcurl](#) 撰寫一個跨平台且依賴性較低的版本。然而寫著寫著，DR 在網路討論區上赫然發現已經有類似的工具了，那就是：[onlineDic](#)，而且 [onlineDic](#) 實做出來的功能已經比 DR 預想的目標還要好上許多。結果.....當然是有點小失望啦，不過換個想法，也不算是做了白工，已經研究過的部份倒是可以寫篇文章，於是這篇 [libcurl](#) 範例文章就誕生了。

在這篇文章中 DR 會示範如何利用 [libcurl](#) 向 Yahoo! 奇摩字典傳送字典查詢的要求，並接收查詢結果。

也許哪天寫點關於 [wxWidgets](#) 的東西.....

[在 Dev-C++ 開發環境使用 libcurl 函式庫](#)

流程如下：

(1) 首先在 [cURL 的官方網站](#) 下載 Win32 - Generic（2000/XP）的 [libcurl](#) 套件，由於要存取的 Yahoo! 奇摩字典網站並沒有使用 SSL（HTTPS），所以可以選擇不支援 SSL 的版本。

(2) 檔案下載後解開來，將 [include](#) 和 [lib](#) 目錄裡的所有東西分別丟到 Dev-C++ 的 [include](#) 和 [lib](#) 目錄裡。

(3) 將 [lib](#) 目錄中的 [libcurl-4.dll](#) 先複製一份到你所要存放 Project 的目錄。

(4) 啟動 Dev-C++ 並新增 Project，然後在 Dev-C++ 的選單：[Project](#)→[Project Options](#)→[Parameters](#)→[Linker](#) 中加入 [-lcurl](#)

(5) 最後在程式碼中加入標頭檔即可：

```
#include <curl/curl.h>
```

[Yahoo! 奇摩字典的資料傳送分析](#)

在開始進行程式碼撰寫之前，必須先瞭解要傳送什麼資料給網站，才能接受到正確的結果。所以請先至 [Yahoo! 奇摩字典](#) 瞭解一下它是怎麼運作的，可以先試著打一個單字，比方說：unix，然後按下『搜尋』，這時候網址列就會跑出以下結果：

<http://tw.dictionary.yahoo.com/search?ei=UTF-8&p=unix>

從網址列可以得知：負責字典搜尋的函式是『[search](#)』，而傳送的資料名稱是『[ei](#)』和『[p](#)』。看樣子 [ei](#) 的值應該是固定的，也就是 UTF-8，而 [p](#) 值就是欲查詢的單字了，你也可以把網址列的結果和網頁的原始碼對照確認一下。

瞭解其資料傳送的方式後，就可以進行程式碼的撰寫了。

[程式碼的撰寫](#)

首先宣告 CURL 變數並啟動 session：

```
CURL *curl;  
curl = curl_easy_init();
```

然後設定 URL：

```
curl_easy_setopt(curl, CURLOPT_URL, "http://tw.dictionary.yahoo.com/search");
```

```
curl_easy_setopt(curl, CURLOPT_URL, "tw.dictionary.yahoo.com/search");
```

向 URL 傳送資料，假設要搜尋「BSD」這個字：

```
curl_easy_setopt(curl, CURLOPT_POSTFIELDS, "ei=UTF-8&p=BSD");
```

接收回應資料，也就是搜尋結果：

```
curl_easy_perform(curl);
```

最後別忘了把 session 關閉：

```
curl_easy_cleanup(curl);
```

完整程式碼範例

以下是完整的程式碼，可以順利編譯與使用：

```
#include <stdio.h>
#include <stdlib.h>
#include <curl/curl.h>

int main(int argc, char *argv[])
{
    char POST[255] = "";
    CURL *curl;
    CURLcode res;

    curl = curl_easy_init();

    sprintf(POST, "ei=UTF-8&p=%s", argv[1]);

    curl_easy_setopt(curl, CURLOPT_URL, "tw.dictionary.yahoo.com/search");
    curl_easy_setopt(curl, CURLOPT_POSTFIELDS, POST);
    res = curl_easy_perform(curl);

    curl_easy_cleanup(curl);
    return 0;
}
```

執行測試

將以上的完整程式碼編譯好後，假設檔名為 **curl-test.exe**，欲搜尋的單字為 **blizzard**，只要執行以下指令：

```
curl-test.exe blizzard > result.html
```

再把 **result.html** 打開來就可看到單字的查詢結果了。

注意：請記得將 **libcurl-4.dll** 和執行檔放在同個目錄下，否則會無法執行。

結語

libcurl 為通訊協定的程式撰寫提供了一個跨平台且更為簡易的方案，有需要的開發者可以多多注意一下。

官方網站（以及套件裡）有許多的文件與參考範例。

→Back to DarkRanger.no-ip.org