最近的项目中由于要在C++代码中调用PHP的URL，所以不得不借助libcurl这个库，由于第一次用，所以很多地方很是纠结，特此写在这里，方便给同样刚入门的朋友指引。  
分两篇介绍，第一篇是理论知识，第二篇是实例。[快速链接--libcurl的使用总结（一）](http://www.vimer.cn/2010/03/libcurl%E7%9A%84%E4%BD%BF%E7%94%A8%E6%80%BB%E7%BB%93%EF%BC%88%E4%BA%8C%EF%BC%89.html)

**一.下载安装**  
    1.到<http://curl.haxx.se/download.html>上下载最新版本，由于公司的机器安装rpm有依赖关系，所以直接下载了source  
    2.编译。解压后进入curl的目录，直接执行 make all 就行。  
    3.等待编译结束后，可以查看目录结构。  
        curl/include/curl ： 头文件目录 （一般只要包含curl.h即可）  
        curl/lib/.lib/      ： lib文件目录（有libcurl.a和libcurl.so，注意，如果这两个文件在同一目录下，-lcurl默认是链接.so滴）

**二.函数简要说明**  
在基于LibCurl的程序里，主要采用callback function （回调函数）的形式完成传输任务，用户在启动传输前设置好各类参数和回调函数，当满足条件时libcurl将调用用户的回调函数实现特定功能。下面是利用libcurl完成传输任务的流程：

1.       调用curl\_global\_init()初始化libcurl  
2.       调用 curl\_easy\_init()函数得到 easy **interface**型指针  
3.       调用curl\_easy\_setopt设置传输选项  
4.       根据curl\_easy\_setopt设置的传输选项，实现回调函数以完成用户特定任务  
5.       调用curl\_easy\_perform（）函数完成传输任务  
6.       调用curl\_easy\_cleanup（）释放内存

在整过过程中设置curl\_easy\_setopt()参数是最关键的，几乎所有的libcurl程序都要使用它。

**1）CURLcode curl\_global\_init(long flags);**  
描述：  
这个函数只能用一次。(其实在调用curl\_global\_cleanup 函数后仍然可再用)  
如果这个函数在curl\_easy\_init函数调用时还没调用，它讲由libcurl库自动完成。  
参数：flags  
CURL\_GLOBAL\_ALL                      //初始化所有的可能的调用。  
CURL\_GLOBAL\_SSL                      //初始化支持 安全套接字层。  
CURL\_GLOBAL\_WIN32            //初始化win32套接字库。  
CURL\_GLOBAL\_NOTHING         //没有额外的初始化。

**2）void curl\_global\_cleanup(void);**  
描述：在结束libcurl使用的时候，用来对curl\_global\_init做的工作清理。类似于close的函数。

3.char \*curl\_version( );  
描述: 打印当前libcurl库的版本。

**4）CURL \*curl\_easy\_init( );**  
描述:  
curl\_easy\_init用来初始化一个CURL的指针(有些像返回FILE类型的指针一样). 相应的在调用结束时要用curl\_easy\_cleanup函数清理.  
一般curl\_easy\_init意味着一个会话的开始. 它的返回值一般都用在easy系列的函数中.

**5）void curl\_easy\_cleanup(CURL \*handle);**  
描述:  
这个调用用来结束一个会话.与curl\_easy\_init配合着用.  
参数:  
CURL类型的指针.

**6）CURLcode curl\_easy\_setopt(CURL \*handle, CURLoption option, parameter);**  
描述: 这个函数最重要了.几乎所有的curl 程序都要频繁的使用它.它告诉curl库.程序将有如何的行为. 比如要查看一个网页的html代码等.(这个函数有些像ioctl函数)参数:  
1 CURL类型的指针  
2 各种CURLoption类型的选项.(都在curl.h库里有定义,man 也可以查看到)  
3 parameter 这个参数 既可以是个函数的指针,也可以是某个对象的指针,也可以是个long型的变量.它用什么这取决于第二个参数.  
CURLoption 这个参数的取值很多.具体的可以查看man手册.

**7）CURLcode curl\_easy\_perform(CURL \*handle);**

描述:这个函数在初始化CURL类型的指针 以及curl\_easy\_setopt完成后调用. 就像字面的意思所说perform就像是个舞台.让我们设置的  
option 运作起来.参数:  
CURL类型的指针.

**三.重要函数详解**  
**1）curl\_easy\_setopt函数介绍**

本节主要介绍curl\_easy\_setopt中跟http相关的参数。注意本节的阐述都是以libcurl作为主体，其它为客体来阐述的。  
1.     CURLOPT\_URL  
设置访问URL  
2.       CURLOPT\_WRITEFUNCTION，CURLOPT\_WRITEDATA  
回调函数原型为：size\_t function( void \*ptr, size\_t size, size\_t nmemb, void \*stream); 函数将在libcurl接收到数据后被调用，因此函数多做数据保存的功能，如处理下载文件。CURLOPT\_WRITEDATA 用于表明CURLOPT\_WRITEFUNCTION函数中的stream指针的来源。  
3.      CURLOPT\_HEADERFUNCTION，CURLOPT\_HEADERDATA  
回调函数原型为 size\_t function( void \*ptr, size\_t size,size\_t nmemb, void \*stream); libcurl一旦接收到http 头部数据后将调用该函数。CURLOPT\_WRITEDATA 传递指针给libcurl，该指针表明CURLOPT\_HEADERFUNCTION 函数的stream指针的来源。  
4.       CURLOPT\_READFUNCTION CURLOPT\_READDATA  
libCurl需要读取数据传递给远程主机时将调用CURLOPT\_READFUNCTION指定的函数，函数原型是：size\_t function(void \*ptr, size\_t size, size\_t nmemb,void \*stream). CURLOPT\_READDATA 表明CURLOPT\_READFUNCTION函数原型中的stream指针来源。  
5.       CURLOPT\_NOPROGRESS，CURLOPT\_PROGRESSFUNCTION，CURLOPT\_PROGRESSDATA  
跟数据传输进度相关的参数。CURLOPT\_PROGRESSFUNCTION 指定的函数正常情况下每秒被libcurl调用一次，为了使CURLOPT\_PROGRESSFUNCTION被调用，CURLOPT\_NOPROGRESS必须被设置为false，CURLOPT\_PROGRESSDATA指定的参数将作为CURLOPT\_PROGRESSFUNCTION指定函数的第一个参数  
6.       CURLOPT\_TIMEOUT，CURLOPT\_CONNECTIONTIMEOUT:  
CURLOPT\_TIMEOUT 由于设置传输时间，CURLOPT\_CONNECTIONTIMEOUT 设置连接等待时间  
7.       CURLOPT\_FOLLOWLOCATION  
设置重定位URL  
CURLOPT\_RANGE: CURLOPT\_RESUME\_FROM:  
断点续传相关设置。CURLOPT\_RANGE 指定char \*参数传递给libcurl，用于指明http域的RANGE头域，例如：  
表示头500个字节：bytes=0-499  
表示第二个500字节：bytes=500-999  
表示最后500个字节：bytes=-500  
表示500字节以后的范围：bytes=500-  
第一个和最后一个字节：bytes=0-0,-1  
同时指定几个范围：bytes=500-600,601-999  
    CURLOPT\_RESUME\_FROM 传递一个long参数给libcurl，指定你希望开始传递的  
偏移量。  
**2）curl\_easy\_perform 函数说明（error 状态码）**

该函数完成curl\_easy\_setopt指定的所有选项，本节重点介绍curl\_easy\_perform的返回值。返回0意味一切ok，非0代表错误发生。主要错误码说明：  
1.    CURLE\_OK  
    任务完成一切都好  
2     CURLE\_UNSUPPORTED\_PROTOCOL  
    不支持的协议，由URL的头部指定  
3     CURLE\_COULDNT\_CONNECT  
    不能连接到remote 主机或者代理  
4     CURLE\_REMOTE\_ACCESS\_DENIED  
    访问被拒绝  
5     CURLE\_HTTP\_RETURNED\_ERROR  
    Http返回错误  
6     CURLE\_READ\_ERROR  
    读本地文件错误