先解释一些术语或名词，SDK 是 Software Development Kit 的简写，也就是软件开发包的意思，其中就包含了我们写程序要用到的一些头文件，库，工具，帮助文档之类的。

Windows API 编程是指调用 Windows 的接口函数来进行程序的编写，例如 Message Box 就是一个 API 函数或者说接口函数。怎么说都可以，自己理解就行。如果你连这个都不太懂，我想也不会搜到这篇文章了吧~·

为什么做这个系列教程呢，请听我一一道来先，最近遇到一些事一些人，让我真的感觉在这方面的引导入门文章真的很是匮乏，加上 Windows SDK 头文件中那些复杂，庞大，‘烦人’的宏定义与数据类型定义，对于一个新手来说(我所说的新手不单只刚接触编程的，还特指那些在其他语言领域有比较高造诣的朋友) 一个纯 SDK 写的 helloworld 程序都算是一个有些困难和挑战的任务了吧。 本着帮助别人,高兴自己的原则，我有了这个打算，当然对自己以前所学，所经历做一次回忆，也是这次计划的一部分。

声明一下，本系列教程是面向广大初次接触 WIN32 SDK 程序编写的新手朋友们的，如果你是高手，一笑而过吧~当然，除了一笑而过，也多谢你们提出指正文章中的错误，以免我误人子弟啊~~谢谢。

废话不多说，进入正题，今天第一篇，讲什么?对于一个新人来说，第一次接触 SDK 编程或者说 API 编程，什么最迷惑你们的，我们讲它，我觉得 Windows SDK 中那'烦人'的数据类型定义和宏定义应该算这个很角色吧。。

其实微软的本意也是善良的，为了减轻程序员的负担，和为了编程的方便，才花了那么多心思与精力定义出了这么一大套数据类型与宏定义,这也是我为什么在之前说它烦人都是加上引号的原因，因为他不是真的烦人，熟练了，你不但不觉得它烦，反而离不开它了，呵呵，日久深情也就是这么来的。

呵呵 先看几个数据类型定义吧

```
typedef float FLOAT;

typedef long LONG;
```

typedef short SHORT

typedef int INT;

typedef char CHAR;

float, long, short, int, char 这几个数据类型都是大家熟悉的 C/C++的数据类型吧，微软将他们重新定义了一下，很简单，就是改变名字为大写了，这样做的目的大概是微软为了编码的方便吧，输入法大小写都不用切换了，多人性化呀 呵呵。

再看几个数据类型定义的例子

typedef unsigned int UINT;

typedef unsigned int UINT32;

typedef signed int INT32;

typedef unsigned long DWORD;

typedef unsigned short WORD;

这些数据类型的定义就稍微有实质性作用一些了，注意观察，他们都比较短了，不用写那么长，而且也还比较直观，如果我要定义一个无符号整形， 我就不用写 unsigned int a；这么长了，只需 UINT a; 多简单， 多明了，所以我说其实不烦人吧。

其中 DWORD 算是 SDK 程序中可以经常看见的一个数据类型了，经常被使用，很多新手也就不明白，这是什么数据类型啊，现在看到了吧，其实就是无符号长整形 unsigned long，给他取了个外号而已··没什么技术含量，所以不用怕，程序中究竟是写 unsigned long 还是 DWORD 都看你自己心情，因为他们都代表同一种数据类型。

下面介绍 2 个很重要的，经常使用到的，无处不在的数据类型 WPARAM,LPARAM，先看看他们定义吧

typedef LONG_PTR LPARAM;

typedef UINT_PTR WPARAM;

先告诉你，这 2 个数据类型很重要，不是危言耸听，以后你写 SDK 程序就知道了，看他们的定义如上，有些迷糊? 别，我们一步一步分析,我们分析 LPARAM。首先定义 LPARAM 为 LONG_PTR 也就是用 LPARAM 的地方也就可以写成 LONG_PTR，LONG_PTR 又是被定义成什么的呢?

typedef long LONG_PTR;

看到了吗?也就是 long 所以归根结底，LPARAM 就是 long 型，所有 LPARAM 型的变量，你都可以直接使用 long 数据类型代替。不过不推荐这样，至于为什么，各位思考思考呢~~

以上这些数据类型是参考 MSDN 中的说明，或者可以查看 WinDef.h 这个头文件查看这些 Windows 数据类型的定义，那么也请各位自己推推看 LARAM 和 WPARAM 的真面目吧~

各位朋友在推导的过程中可能发现 LONG_PTR 的定义是这样写的

#if defined(_WIN64)

typedef __int64 LONG_PTR;

#else

typedef long LONG_PTR;

#endif

这是什么意思呢，能看懂英文都能知道这在定义些什么,如果定义了 _WIN64 这个宏那么就定义 LONG_PTR 为 __int64，否则定义 LONG_PTR 为 long。 很简单吧 也就是说如果_WIN64 这个宏在前面被定义了，那么这里的 LONG_PTR 就被定义为__int64 这个在 64 位编程下的数据类型，否则就定义为 long 型，这样说应该比较好理解了吧。在这里，

各位就不必深究__int64 了，在目前的主流 32 位编程下很少使用它啦。理解就 ok 了。这样定义是微软为了程序员编写的程序能在 32 位与 64 位下都能编译而采用的伎俩。

有关这些 Windows 的数据类型，想查看他们的真面目，其实很简单，在 VC6.0,VS2008 这些集成开发环境里面，你只需要在一个数据类型上面点击右键，在弹出菜单中选择'Goto Defination' 或者是 '查看定义'就可以看到了，如果看到的还不是最终面目，在继续上面步骤。直到看到它的本质数据类型为止。通过这样，新手对于 Windows 的这些复杂的数据类型定义也就有了根本的认识，不再是迷迷糊糊，在以后的编程中也就不会出现不知道用哪种数据类型或者哪些数据类型之间可以相互转换的情况了。不过还需要多多观察与练习才是啊~~

下面再来看一看 windows 中定义的一些宏

```
#define   VOID       void

#define  CONST    const
```

2 个最简单的宏，也是只变成大写而已，难道又是为了方便程序员不切换输入法？还真的人性化呀。

Windows SDK 中的宏定义是最庞大的，最复杂的，但也是最灵活的，为什么这样说，先不告诉你，我会在以后的系列文章中一点一点的讲解，累积，因为太多了，也比较复杂，我们就采取在需要用到的时候才讲解它，目前看来还没这个必要了解那么多，就了解上面 2 个很简单的好了，像其他如：WINAPI CALLBACK GetWindowText 这些宏现在讲了不但记不住还会增加你们的负担。我们就在以后要用到的时候再做讲解。

到这里第一篇系列文章的内容也就差不多了。新手朋友们哪些地方迷惑的，提出来，我可以考虑是否加在后续的文章中进行解说。本 SDK 系列入门教程需要你们的支持。谢谢。