uo_blob 对象在 WINAPI 调用中的巧妙使用

许多人对 pbidea 中的 uo_blob 对象没有多少印象。这个对象具有强大功能。

下面用一个 WINAPI 调用的例子，再介绍一下它的使用。

一．GetSystemPowerStatus

首先看函数声明：

```
BOOL GetSystemPowerStatus(
   [out] LPSYSTEM_POWER_STATUS lpSystemPowerStatus
);
```

这个函数使用一个结构，我们再来看它的结构：

```
typedef struct _SYSTEM_POWER_STATUS {
  BYTE  ACLineStatus;
  BYTE  BatteryFlag;
  BYTE  BatteryLifePercent;
  BYTE  SystemStatusFlag;
  DWORD BatteryLifeTime;
  DWORD BatteryFullLifeTime;
} SYSTEM_POWER_STATUS, *LPSYSTEM_POWER_STATUS;
```

结构前 4 个是字节，后 2 个是 DWORD。**BYTE 类型是 1 个字节，DOWRD 是 4 个字节**，所以，**这个结构大小是 12 个字节**。

我们常规调用方法是：

1.先在 PB 里声明一个 Structure 对象 str_system_power_status,注意，高版本 PB 才有 BYTE 类型 ，PB9 声明时要使用 character，高版本必须用 BYTE。有些人在 PB 升级时不知道要把 character 改成 BYTE，结果肯定是错误的。

2.声明函数

```
function GetSystemPowerStatus(ref str_system_power_status s) library “kernel32.dll”
```

3.调用 ：

```
str_system_power_status status
GetSystemPowerStatus(status)
```

那么，使用 uo_blob 如何来调用这个函数呢？

由上面的分析，我们知道结构的大小是 12 个字节。GetSystemPowerStatus 是传引用参数，传引用也就是把地址传给它，而地址实际上就是一串数字表示内存里的一个位置。

所以，我们这样声明函数：

```
function GetSystemPowerStatus(ulong add) library “kernel32.dll” //直接用一个数字表示地址
```

至于结构，就不需要声明了。接下来使用 uo_blob 调用它:

```
uo_blob b;b = create uo_blob //对象实例化
b.resize(12) //分配大小空间，12 个字节，可以更大，但不能小于 12
GetSystemPowerStatus(b.ptr()) //这里就调用了
```

uo_blob 有个函数 ptr() 直接返回的是它在内存里的地址，指向已分配的空间。函数调用后，数据就放到那段内存空间里去了。

```
int  ACLineStatus,BatteryFlag,BatteryLifePercent,SystemStatusFlag  //声明变量，也就是结构里的前 4 个 BYTE 成员
```

long BatteryLifeTime,BatteryFullLifeTime //声明 2 个变量，也就是结构里的后 2 个 DWORD 成员
b.get(1,ACLineStatus,true) //从第 1 个字节读 1 个字节
b.get(2,BatteryFlag,true) //从第 2 个字节读 1 个字节
b.get(3,BatteryLifePercent,true) //从第 3 个字节读 1 个字节
b.get(4,SystemStatusFlag,true) //从第 4 个字节读 1 个字节
b.get(5,BatteryLifeTime) //从第 5 个字节读 4 个字节
b.get(9,BatteryFullLifeTime) //从第 9 个字节读 4 个字节
至此，函数调用完成，取得各个成员数据。避免多声明一个结构对象。
更多 uo_blob 功能，请参考 pbidea demo 。