

PB 的扩展 DLL 开发（超级篇）（五）

(PB 史上第一次开放的开发技术)

上一节里，我们说到参数获取与返回值，这一节里，我们来谈谈引用。

其实引用，也是一种返回数据的方式。

假如 PB 里有这样的声明方式：

Function boolean GetText(ref string as_text,ref int al_textlen) system library "my.dll" alias
for "GetText"

这个函数里，就涉及到 2 个引用型参数。我们在 DLL 内部是这样实现：

```
DLLEXPORT DWORD WINAPI GetText(POB_THIS obThis, int nArgs)
{
    //先声明2个参数
    POB_DATA argText = NULL;
    POB_DATA argTextLen = NULL;

    //先获取as_text
    {
        POT_LVALUE_INFO info = NULL;
        POB_DATA obRefArg = ot_get_next_lvalue_arg(obThis, &info);
        if (ot_is_a_reference_argument(obThis, obRefArg))
        {
            POT_REF_PAK v = (POT_REF_PAK)ob_get_data_ptr(obRefArg);
            argText = ot_access_ref_data(obThis, v);
        }
    }

    //再获取al_textlen
    {
        POT_LVALUE_INFO info = NULL;
        POB_DATA obRefArg = ot_get_next_lvalue_arg(obThis, &info);
        if (ot_is_a_reference_argument(obThis, obRefArg))
        {
            POT_REF_PAK v = (POT_REF_PAK)ob_get_data_ptr(obRefArg);
            argTextLen = ot_access_ref_data(obThis, v);
        }
    }

    //返回 as_text
    if (argText)
    {
        TCHAR* lpText = ob_dup_string(obThis, "this is text");
```

```

        ob_set_data_ptr(argText, lpText, STRING_TYPE, 1);
    }
    //返回 al_textlen
    if (argTextLen)
    {
        ob_set_data_int(argTextLen, 12, INT_TYPE, 1);
    }

    //设置函数返回值为 true
    OB_DATA obReturn = { 0 };
    ob_set_data_int(&obReturn, TRUE, BOOL_TYPE, 1);
    ot_set_return(obThis, &obReturn);
    return(1);
}

```

从代码中可以看出，一个引用参数，它实际是一个OB_DATA，它是一个指针，需要用ot_get_next_lvalue_arg这个函数取它的引用参数，指向一个POT_REF_PAK结构指针，这个结构指针内部又放了另一个OB_DATA，argText = ot_access_ref_data(obThis, v)得到我们需要的变量参数，这个才是我们需要真正操作数据的地方。有了这个参数，就当一般OB_DATA操作就可以了，比如ob_set_data_ptr、ob_set_data_int等。

我们来看看这个 OT_REF_PAK 结构：

```

typedef struct ot_ref_pak
{
    OT_REFPAK_STYLE      style;
    OB_GROUP_HNDL        group_hndl;
    OB_CLASS_ID          type;
    USHORT               flags;
    OT_REF_TAG_UNION     ref;
} OT_REF_PAK, FAR* POT_REF_PAK;

```

其中的 OB_CLASS_ID type 指明了这个引用是什么类型。更多的复杂内容，自己慢慢研究吧。

通常我们使用函数传入的引用参数，但也许我们呼叫自定义对象成员事件的时候，我们需要通过事件的引用函数获取我们需要的返回值，那我们就要在呼叫事件前，构造这样一个引用参数，然后再呼叫对象的事件。我们可以使用宏 ot_build_simple_refarg 来构建我们自己的引用参数。

如果你对 system library 相关开发方式感兴趣，可到 QQ 群 624409252 共享里大自在的专用目录下下载 DEMO。