

## PB 的扩展 DLL 开发（超级篇）（六）

(PB 史上第一次开放的开发技术)

### 数组使用及返回引用数组

数组的定义是这样：

```
Typedef PVOID OB_ARRAY_ID, FAR* POB_ARRAY_ID;
```

可见它是一个指针。本节举一个数组复制的函数例子，PB 中声明如下：

```
Function int CopyArray(string fromArray[],ref string toArray[]) system library "my.dll" alias  
for "CopyArray"
```

```
DLLEXPORT DWORD WINAPI CopyArray(POB_THIS obThis, int nArgs)
{
    BOOL isnull = FALSE;
    //先声明2个数组指针变量
    OB_ARRAY_ID fromArray = NULL;
    OB_ARRAY_ID toArray = NULL;

    //取第一个参数
    fromArray = (OB_ARRAY_ID)ot_get_valptr_arg(obThis, &isnull);
    //取第二个参数,引用类型
    {
        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);
            POB_DATA obToArray = ot_access_ref_data(obThis, v);
            if (ot_is_array(obThis, obToArray))
            {
                //判断此数据是数组
                toArray = (OB_ARRAY_ID)ob_get_data_ptr(obToArray);
            }
        }
    }

    int nCopyIndex = 0;
    //判断,其中返回值的引用数组必须是一个 unbounded 数组
    if (fromArray && toArray && ot_is_array_unbounded(obThis, toArray))
    {
        //获取源数组中的项数
        int nItemCount = ot_array_num_items(obThis, fromArray);
```

```

//循环取出所有源项，并且复制
for (int i = 0; i < nItemCount; i++)
{
    //源数据项
    POB_DATA obFromItem = ot_array_index(obThis,fromArray, i);
    //目标数据项
    POB_DATA obToItem = ot_array_index(obThis, toArray, nCopyIndex);

    //进行数据项复制
    //注意第4个参数，这个参数表示是否引用，对于一般数据类型，直接是复制，但对于对象实例，不可能再创建一个对象实例，而应该是引用，即源数组和目标数组中的对象指针，指向同一个对象实例。
    rtDataCopy(obThis, obToItem, obFromItem, TRUE);

    //目标索引加1
    nCopyIndex++;
}
}

//设置函数返回值为复制的个数
OB_DATA obReturn = { 0 };
ob_set_data_int(&obReturn, nCopyIndex, INT_TYPE, 1);
ot_set_return(obThis, &obReturn);
}

```

以上是一个关于数组使用的函数，这个函数示例中都加了详细的中文注解，应该比较好理解。

数组的使用比较复杂，以上只是对一般使用做个介绍，能满足大多数普通应用。

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