

## aardio 范例: COM 接口 - 输出与引用参数

```
//COM 接口 - 输出与引用参数
import fonts.fontAwesome;
import win.ui;
/*DSG{ {*/
var winform = win.form(text="COM 输出参数";right=700;bottom=266)
winform.add(
edit={cls="edit";left=216;top=22;right=662;bottom=245;db=1;dl=1;dr=1;dt=1;edge=1;hscroll=1;multiline=1;vscroll=1;z=1};
static={cls="static";text="Static";left=20;top=24;right=199;bottom=234;transparent=1;z=2}
)
/*}}*/

import com.lite;
var dll = com.lite("~/example/Languages/VB/.vb6/Vb6Control.ocx")
var vbCtrl = dll.createEmbedEx(winform.static);
winform.show();

/*
对于大部分附带运行时 COM 类型库的 COM 对象，
如果使用了输出与引用参数，只要简单地在返回值接收输入参数就可以了。
*/

var ret,outStr = vbCtrl.GetOutStr("");
var ret,outStr = vbCtrl.GetOutStr(); //带类型库且不需要输入值时通常可以省略
winform.edit.print("调用 VB 控件函数返回值:",ret, outStr );

/*
有极少数 COM 对象，使用了输出参数，又不提供运行时类型库。
这时候我们就需要用 com.Variant 定义一个类型明确的 COM 变体对象作为参数。
并且指定参数 @3 为 true ，以声明这是一个输出与引用参数。
*/
var ptrOutStr = com.Variant("",8/*_VT_BSTR*/,true);//BSTR 类型可自动识别，参数 @2 可以省略为空。

//调用函数，输出参数仍然会添加到返回值列表（返回值是传值，不是引用输出参数）。
var ret = vbCtrl.GetOutStr(ptrOutStr);//传址（也就是传指针值）。

//COM 函数直接修改了输入参数的值。
winform.edit.print("输出参数新值:", ptrOutStr.value );

/*****
以下基于 com.Variant 实现的函数将参数 @2 指定为 true 可创建输出参数:
*****/

namespace com{
    float = lambda(v,ref) Variant(v,4/*_VT_R4*/,ref);
    double = lambda(v,ref) Variant(v,5/*_VT_R8*/,ref);
    byte = lambda(v,ref) Variant(v,0x10/*_VT_I1*/,ref);
    ubyte = lambda(v,ref) Variant(v,0x11/*_VT_UI1*/,ref);
    word = lambda(v,ref) Variant(v,2/*_VT_I2*/,ref);
    uword = lambda(v,ref) Variant(v,0x12/*_VT_UI2*/,ref);
    int = lambda(v,ref) Variant(v,3/*_VT_I4*/,ref);
    uint = lambda(v,ref) Variant(v,0x13/*_VT_UI4*/,ref);
    long = lambda(v,ref) Variant(v,0x14/*_VT_I8*/,ref);
    ulong = lambda(v,ref) Variant(v,0x15/*_VT_UI8*/,ref);
}
*****/

winform.show();
win.loopMessage();
```