

## aardio 范例: COM 接口 - 响应事件

```
//COM 接口 - 响应事件
import win.ui;
/*DSG{*/
var winform = win.form(text="响应COM事件";right=759;bottom=469)
winform.add(
edit={cls="edit";left=9;top=13;right=749;bottom=457;db=1;dl=1;dr=1;dt=1;edge=1;multiline=1;z=1}
)
/*}*/

//创建 COM 对象,注意服务器系统没这个控件
var wmPlayer = com.CreateObject("WMPlayer.OCX"); //注意 win.ui 自动导入了 com 库。
/*
关于 WMPlayer.OCX 控件的更多用法,请参考:
范例 > 音频视频 > 音频 > lrc 歌词解析
*/

//定义事件对象
var ocxEvents1 = {
    StatusChange = function() {
        winform.edit.print("StatusChange",wmPlayer.Status)
    };
    MediaChange = function(item){
        winform.edit.print("ocxEvents1.MediaChange",item.sourceURL)
    };
}

//第一种最简单的写法: 挂接默认的事件监听器
//参考文档: https://www.aardio.com/zh-cn/doc/library-guide/builtin/com/event.html
com.Connect(wmPlayer, ocxEvents1);
/*
参数 wmPlayer 在作用域内事件才会有效, 如果 wmPlayer 被回收则会自动解除事件绑定。
第一个参数指定临时变量会在稍后被回收, 例如: com.Connect( execl.ActiveWorkbook , ... ) 这样写是错误的。
正确写法是先写 var book = execl.ActiveWorkbook 然后再 com.Connect( book, ... )
*/

//下面是第二种复杂的写法: 挂接指定接口的事件监听器
var ocxEvents2 = {
    MediaChange = function(item) {
        winform.edit.print("ocxEvents2.MediaChange",item.sourceURL)
    }
}
var eventSink = com.ImplInterface(ocxEvents2,"WMPlayer.OCX.7","_WMPOCXEvents")
var cookie = com.AddConnection( wmPlayer,eventSink );//挂接事件监听器

//释放前面挂接的事件监听器
com.ReleaseConnection(wmPlayer,eventSink,cookie);

//使用 COM 对象打开指定的音频
wmPlayer.url = "http://download.aardio.com/v10.files/demo/mp3/lrc.mp3"

/*
请仔细阅读下面这段说明,
实际上我们使用 COM 控件上基本遇不到下面的要讲的问题,
甚至基本不用自己绑定事件监听器, 但这背后的基本规则必须有所了解。
不然这个范例后面的几个范例你是看不懂的。

注意绑定事件的 com.Connect() 或 com.AddConnection()
会将参数@1指定的 COM 对象, 以及参数@2指定的事件表对象强绑定,
这里如果不小心建立了循环引用, 会导致2个对象都无法自动释放。
注意: 普通 aardio 表对象之间循环引用不存在这个问题。

所以一般应当避免绑定自身作为事件监听器, 例如:
com.Connect( comObject, comObject) 这样写是错的。

或者绑定引用了事件监听器的成员作为事件监听器, 例如:
com.Connect( eventTable.comObject, eventTable)
这样写也是错的。

但我们可能希望通过同一个对象访问 COM 对象并指定事件回调函数,
aardio 提供了 com.ConnectWeak() 以及 com.CreateEmbedEx() 可以实现这样的功能。
aardio 可以利用元表轻松地将一个表代理到另一个表、或建立弱引用表解决这种问题。
有兴趣可以看看这几个函数的源码( 位于标准库: builtin.com )。

实际上我们很少需要自己绑定事件并使用 com.Connect() 这些函数,
aardio 中嵌入窗口控件的 winform.createEmbed(), winform.createEmbedEx()
```

已经自动绑定返回的 COM 容器对象作为事件监听器，  
这几个函数虽有前述的循环引用，但是已经在宿主窗口销毁前  
自动解除事件监听并释放对象，所以能正常释放。

```
*/
```

```
/*
```

为了简化，这里创建的 WMPlayer.OCX 没有嵌入宿主窗口，  
但要注意如果要愉快的听歌，需要下面的 win.loopMessage()  
关闭窗口退出程序以后，COM 控件也会随之退出。

```
*/
```

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

[Markdown 格式](#)