

aardio 范例: 跨语言跨线程回调 消息 _WM_THREAD_CALLBACK

```
//跨语言跨线程回调
import win.ui;
/*DSG{*/
var winform = win.form(text="跨语言跨线程回调 消息 _WM_THREAD_CALLBACK";right=599;bottom=399)
winform.add(
edit={cls="edit";left=17;top=22;right=585;bottom=379;edge=1;multiline=1;z=1}
)
/*}*/

import web.json;
winform.onSendStruct = function(structParam){
    winform.edit.print("onSendStruct",structParam);
    structParam.x = 123;

    //可选返回修改后的值
    return structParam;
}

winform.onSendJson = function(jsonParam){
    winform.edit.print("onSendJson",jsonParam);

    import ide;
    ide.openDocument("~/example/Languages/Delphi/dll.aardio");
}

thread.invoke(
function(hwnd){
    /*
    _WM_THREAD_CALLBACK 使所有回调安全的转发到UI线程。
    _WM_THREAD_CALLBACK 可以跨线程跨编程语言使用。
    */
    var ret,structParam = ::User32.SendMessage(
        hwnd,0xACCE/* WM_THREAD_CALLBACK*/,
        //窗体函数名( 结构体原型声明 ); 结构体原型声明应使用aardio语法
        "onSendStruct( { int x;int y;int id } )",
        { int x = 1;int y = 2;id = thread.getId() }
    )

    import web.json;
    ::User32.SendMessage(
        hwnd,0xACCE/* WM_THREAD_CALLBACK*/,
        //如果结构体原型声明为 "{JSON}" 则下一个参数传入JSON文本
        "onSendJson( {JSON} )",
        //界面线程如果调用了 import web.json, 则会自动解析下面的JSON文本, 解析结果将作为被调用函数的调用参数
        //json必须使用一个数组包含多个参数
        web.json.stringify( { structParam } )
    )
},winform hwnd
)

/*
// WM_THREAD_CALLBACK 会将调用窗口函数的返回值转换为数值, 无法返回其他值。
//可在下面的事件内拦截调用窗口函数的返回值, 并返回新的返回值( 仍会转换为数值)。
winform.onThreadCallBackReturn = function(result){
    return result;
}
*/

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