

aardio 范例: AutoCAD - LISP 调用 aardio

```
//AutoCAD - LISP 调用 aardio
if(_STUDIO_INVOKED) error("请复制此代码到 aardio 工程的 main.aardio 内, 然后生成 EXE 文件再运行",2);
/*
LISP 调用 aardio 的三种途径
1、aardio 创建 ActiveX 控件, 在 LISP 中调用。
打开「aardio 工程向导 / 窗口程序 /ActiveX EXE」创建工程,
发布 EXE 以后, 鼠标双击 EXE 可自动注册该控件。
```

该工程目录下提供 test.lsp 包含 AutoLisp 调用示例。
注意要用 vlax-invoke 而不是 vlax-invoke-method 调用 aardio 函数。
并且 AutoLISP 会自动将函数名转为大写, aardio 大小写敏感, 所以在 aardio 里被调用函数名也要大写。

然后在 AutoCAD 中以如下 LISP 代码创建并调用 aardio 控件:
(setq aarObj (vlax-get-or-create-object "aardioTestControl.Sample"))(vlax-invoke aarObj 'Add 1 3)

2、执行以下 LISP 代码异步启动 aardio 编写的 EXE。
(startapp "d:/test/test.exe" (strcat "/hwnd" " " (itoa(vla-get-hwnd(vlax-get-acad-object)))))

在 aardio 中使用 _ARGV 得到解析后的 _ARGV.hwnd 参数。
或者用 _CMDLINE 得到解析前的命令行参数。

上面的代码主要演示传命令行参数的方法。
其实在 aardio 中用 process.getParent().getMainWindow() 可以直接取到 AutoCAD 窗口。

有了 AutoCAD 的句柄,
然后使用 com.cad.sendCopyData(hwnd,"") 就可以发送命令了。

3、下面示例演示的 startExe 函数阻塞启动 aardio 编写的 EXE。
*/

```
import fsys.dlg;
import process;

//获取父进程对象
var parentProcess = process.getParent();

//直接拿到 AutoCAD 窗口
var hwndCad;
if(parentProcess) hwndCad = parentProcess.getMainWindow();

//自标准输入读取 LISP 传入的值
var cmdline = io.stdin.read();

if(! (cmdline && parentProcess) ){
    import console
    console.showLoading("正在为 为 AutoCAD 安装 startExe( cmdline ) 函数。");

    import com.cad;
    var cad = com.cad()
    cad.Visible = true;

    //直接添加函数
    cad.SendCommand(`(defun startExe(cmdline)
    (setq wsc (vlax-get-or-create-object "WScript.Shell"))
    (setq handle (itoa(vla-get-hwnd(vlax-get-acad-object)))))
    (setq wsx (vlax-invoke wsc 'EXec <?

    //将当前 EXE 路径传入 lisp 代码, 模板参数放入数组可自动处理转义字符, 首尾自动加双引号。
    ={ io._exepath }
    ?>))
    (setq stat(vlax-get wsx 'status ))

    (setq stdin(vlax-get wsx 'StdIn ))
    (vlax-invoke stdin 'Write (strcat cmdline "\n"))

    (setq stdout(vlax-get wsx 'StdOut ))
    (setq out1 (vlax-invoke stdout 'Readall))
    out1
    `)`

    console.log("已为 AutoCAD 安装 startExe( cmdline ) 函数");
    console.pause();
    return;
}
```

```
import process;
import string.cmdline;

//解析命令行参数
var argv = string.cmdline.argv(` ` + cmdline);

//获取 AutoCAD 主窗口句柄
var hwndAutoCAD = parentProcess.getMainWindow();
//注意这时候 AutoCAD 是阻塞状态，不应也不必要设置所有者窗口为 hwndAutoCAD

//请在 AutoCAD 输入命令 (startExe "/c=dlg")
if(argv.c = "dlg"){
    var path = fsys.dlg.open();

    //把返回值写到标准输出
    if(path) io.stdout.write(path);
}
else {
    io.stdout.write("暂未支持此参数");
}
```

[Markdown 格式](#)