aardio 范例: RSA 签名

```
//RSA 签名
import win.ui;
/*DSG{{*/
var winform = win.form(text="RSA签名";right=797;bottom=561)
winform.add(
btnExportPrivatePkcs1Raw={cls="button";text="导出 PKCS#1 私钥";left=29;top=196;right=166;bottom=232;dl=1;dt=1;z=5};
btnExportPrivatePkcs8={cls="button";text="导出 PKCS#8 私钥";left=29;top=87;right=166;bottom=123;dl=1;dt=1;z=4};
btnExportPublicPkcs1Raw={cls="button";text="导出 PKCS#1 公钥";left=29;top=141;right=166;bottom=177;dl=1;dt=1;z=3};
btnExportPublicX509={cls="button";text="导出 SPKI 公钥";left=29;top=32;right=166;bottom=68;dl=1;dt=1;z=2};
btnImportKey={cls="button";text="导入公钥或私钥(自动识别)";left=418;top=257;right=768;bottom=293;db=1;dr=1;z=6};
btnSign={cls="button";text="签名(使用私钥)";left=435;top=509;right=569;bottom=545;db=1;dr=1;z=8};
btnVerify={cls="button";text="验证(使用公钥)";left=586;top=509;right=720;bottom=545;db=1;dr=1;z=9};
editKey={cls="richedit";left=190;top=22;right=770;bottom=252;db=1;dl=1;dr=1;dt=1;edge=1;font=LOGFONT(name='新宋体');hscroll=1;multiline=1;vscroll=1;z=1};
editSign={cls="edit";left=190;top=302;right=770;bottom=368;db=1;dl=1;dr=1;edge=1;hscroll=1;multiline=1;z=10};
editText={cls="richedit";text="测试数据( UTF-8 编码)";left=29;top=376;right=770;bottom=504;db=1;dl=1;dr=1;edge=1;hscroll=1;multiline=1;vscroll=1;z=7};
static={cls="static";text="签名:";left=78;top=316;right=174;bottom=337;align="right";db=1;dl=1;transparent=1;z=11}
)
/*}}*/

import crypt.rsa;
var rsa = crypt.rsa();
rsa.genSignatureKey();

winform.btnExportPublicX509.oncommand = function(id,event){
    //导出通用的 SPKI (Subject Public Key Info) 格式公钥
    winform.editKey.text = rsa.exportPublicKeyX509ToPem();
}

winform.btnExportPublicPkcs1Raw.oncommand = function(id,event){
    winform.editKey.text = rsa.exportPublicKeyPkcs1RawToPem();
}

winform.btnExportPrivatePkcs8.oncommand = function(id,event){
    winform.editKey.text = rsa.exportPrivateKeyPkcs8ToPem();
}

winform.btnExportPrivatePkcs1Raw.oncommand = function(id,event){
    winform.editKey.text = rsa.exportPrivateKeyPkcs1RawToPem();
}

winform.btnImportKey.oncommand = function(id,event){
    var header = rsa.importPemKey(winform.editKey.text);
    if(header) {
        winform.msgbox("已导入: " + header);
    }
    else winform.msgboxErr("错误的密钥格式")
}

winform.btnSign.oncommand = function(id,event){

    //SHA256withRSA
    rsa.createHashBySha256(winform.editText.text);
    var sign =  rsa.signToBase64();

    if(!sign) return winform.msgboxErr("签名失败,请检查是否导入了正确的私钥");
    winform.editSign.text = sign;

}

winform.btnVerify.oncommand = function(id,event){

    //SHA256withRSA
    rsa.createHashBySha256(winform.editText.text);
    if( !rsa.verifyFromBase64( winform.editSign.text ) ){
        winform.msgboxErr("签名是错误的,数据已被篡改,或未导入正确的公钥");
    }
    else {
        winform.msgbox("签名是正确的,数据未被篡改");
    }
}

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

Markdown 格式