
XTXAppCOM

控件接口说明文档

(V2.4)



北京数字认证股份有限公司

2014 年 4 月

目 录

欢迎使用..... 1

版权声明..... 2

阅读指南..... 3

 约定..... 4

第 1 章 产品简介..... 5

 1.1 产品架构..... 5

 1.2 功能特点..... 6

第 2 章 XTXAPP 接口说明..... 8

 2.1 应用接口..... 8

 2.1.1 获取控件版本号..... 8

 2.1.2 设置签名算法..... 8

 2.1.3 获取签名算法..... 9

 2.1.4 设置加密算法..... 9

 2.1.5 获取加密算法..... 9

 2.1.6 获取证书用户列表..... 10

 2.1.7 导出用户证书..... 10

 2.1.8 导出用户加密证书..... 10

 2.1.9 校验证书口令..... 11

 2.1.10 验证证书有效性..... 11

 2.1.11 登出证书口令..... 12

 2.1.12 获取口令重试次数..... 12

 2.1.13 修改证书口令..... 13

 2.1.14 获取证书基本信息..... 13

 2.1.15 获取证书扩展信息..... 15

 2.1.16 获取证书唯一实体标识..... 15

 2.1.17 数据签名..... 16

 2.1.18 数据验签..... 16

 2.1.19 文件签名..... 16

2.1.20 文件验签.....	17
2.1.21 数字信封加密数据.....	18
2.1.22 数字信封解密数据.....	18
2.1.23 数据签名(P7 格式).....	19
2.1.24 数据验签(P7 格式).....	19
2.1.25 解析签名(P7 格式).....	20
2.1.26 产生随机数.....	20
2.1.27 公钥加密数据.....	21
2.1.28 私钥解密数据.....	21
2.1.29 对称加密数据.....	22
2.1.30 对称解密数据.....	22
2.1.31 对称加密文件.....	22
2.1.32 对称解密文件.....	23
2.1.33 Base64 编码.....	24
2.1.34 Base64 解码.....	24
2.1.35 数据摘要.....	24
2.1.36 文件摘要.....	25
2.1.37 文件 Base64 编码.....	25
2.1.38 Base64 解码扩展.....	26
2.1.39 获取最后一次出错码.....	26
2.1.40 获取最后一次出错信息.....	27
2.1.41 OTP 获取挑战码.....	27
2.1.42 计算 HMAC.....	27
2.1.43 明文私钥签名.....	28
2.1.44 对摘要数据签名.....	28
2.1.45 对摘要数据验签.....	29
2.1.46 XML 数据签名.....	29
2.1.47 XML 验签.....	29
2.1.48 解析 XML 签名信息.....	30
2.1.49 门限拆分.....	31

2.1.50 门限恢复.....	31
2.2 发证接口.....	32
2.2.1 获取设备数量.....	32
2.2.2 获取设备数量扩展.....	32
2.2.3 获取所有的设备序列号.....	33
2.2.4 获取所有的设备序列号扩展.....	33
2.2.5 根据索引获取设备序列号.....	33
2.2.6 判断设备是否存在.....	34
2.2.7 获取设备详细信息.....	34
2.2.8 修改管理员口令.....	35
2.2.9 解锁用户口令.....	36
2.2.10 解锁用户口令 Ex.....	36
2.2.11 产生密钥对.....	36
2.2.12 导出公钥.....	37
2.2.13 导出 PKCS10 证书请求.....	38
2.2.14 导入签名证书.....	38
2.2.15 导入加密证书和加密密钥对.....	39
2.2.16 导入加密证书和加密密钥对扩展.....	39
2.2.17 枚举文件.....	40
2.2.18 读文件.....	40
2.2.19 读文件 Ex.....	40
2.2.20 写文件.....	41
2.2.21 写文件 Ex.....	41
2.2.22 获取容器数量.....	42
2.2.23 获取所有容器名称.....	42
2.2.24 判断容器是否存在.....	43
2.2.25 删除容器.....	43
2.2.26 删除最旧的一个容器.....	43
2.2.27 格式化设备.....	44
2.2.28 格式化设备 Ex.....	44

2.2.29 获取当前在用容器名称(读取 ENVSN).....	45
2.2.30 设置当前在用容器名称(写入 ENVSN).....	45
2.2.31 更新证书.....	45
2.3 二进制数据接口.....	47
2.3.1 数据签名.....	47
2.3.2 数据验签.....	47
2.3.3 数字信封加密数据.....	48
2.3.4 数字信封解密数据.....	48
2.3.5 数据签名(P7 格式).....	49
2.3.6 数据验签(P7 格式).....	49
2.3.7 公钥加密数据.....	50
2.3.8 私钥解密数据.....	50
2.3.9 Base64 编码.....	51
2.3.10 Base64 解码.....	51
2.3.11 数据摘要.....	51
2.4 其他接口.....	52
2.4.1 检测软设备环境.....	52
2.4.2 创建一个软设备.....	52
2.4.3 删除一个软设备.....	53
2.4.4 禁用或启用软设备.....	53
2.4.5 软设备备份.....	54
2.4.6 软设备恢复.....	54
2.4.7 将密钥和证书导入到软设备中.....	54
2.4.8 导入 P12 证书到软设备中.....	55
2.4.9 设置用户配置信息.....	55
2.4.10 选择文件.....	57
2.4.11 打开指定的路径.....	57
2.5 调用示例.....	58
2.5.1 JavaScript 调用方法.....	58
2.5.2 VC 调用方法.....	60

2.5.3 C#调用方法..... 65

2.5.4 Delphi 调用方法..... 67

2.5.5 PowerBuilder 调用方法..... 70

第 3 章 附录.....75

3.1 签名算法定义..... 75

3.2 加密算法定义..... 75


3.3 摘要算法..... 77


3.4 错误码定义..... 78

3.5 关于日志..... 81

欢迎使用

欢迎您使用 BJCA 证书应用环境,XTXAppCOM 控件接口说明文档,如果本手册能为您提供帮助,带来便利,我们将深感欣慰。如果您在使用过程中,遇到了问题,或对我们产品有好的建议,可以:

 致电客户服务热线 (010) 58515511

 或访问公司网站: www.bjca.org.cn

与我们联系,对您提出的问题或建议,我们表示衷心的感谢。

版权声明

本手册著作权属北京数字认证股份有限公司所有，在未经本公司许可的情况下，任何单位或个人不得以任何方式对本手册的部分或全部内容擅自进行增删、改编、节录、翻印、改写。

北京数字认证股份有限公司




©2013

阅读指南

本手册可以辅助您快速了解和掌握 XTXAppCOM 控件的各项功能和具体的调用方法。

- ◆ 本手册主要包括四个部分，本节阅读指南引导您了解本手册的主要内容、快速使用说明、阅读中的注意事项以及手册约定；
 - ◆ 第 1 章为产品简介，向您介绍本产品的基本功能、用途、产品特点和相关技术情况，帮助您对本产品具有直观的认识和了解；
 - ◆ 第 2 章为接口使用说明，分别向您介绍本控件支持的接口，以及接口的详细参数说明、返回值说明等；
 - ◆ 第 3 章是本控件的算法对应的值、错误码和日志等进行说明。
- ☞ 如果您是集成实施人员，本手册第 2 章“接口使用说明”的内容可以对您有所帮助。

约定

约定标识	说明
[]	方括号包含可选的参数。
【 】	表示按钮符号，如 【确定】 即指界面上的确定按钮。
...	在语法行中，省略号指示可能包括更多相同格式的项目。
	此符号代表警告提示，需要读者特别注意该内容。
	此符号代表对内容进行特别说明或解释。
	此符号在界面中所对应的输入框或选择框为必填项或必选项

第 1 章 产品简介

本组件是基于微软的 COM 技术生成的，它可以被多种语言如 C#、VC、JavaScript、ASP 和 Delphi 方便的调用。它是 BJCA 自主研发的证书的客户端部分，主要提供了生成随机数、对称算法加解密数据和文件、加解密数据、对数据和文件签名认证、数字信封、验证解析证书、时间戳和 XML 签名等功能。用户通过使用本组件和数字证书，可以方便、灵活的实现安全登录和对信息的保护。

组件名为 XTXAppCOM.dll，包含一个应用接口类为 XTXApp，符合证书应用综合服务接口——国密局规范。

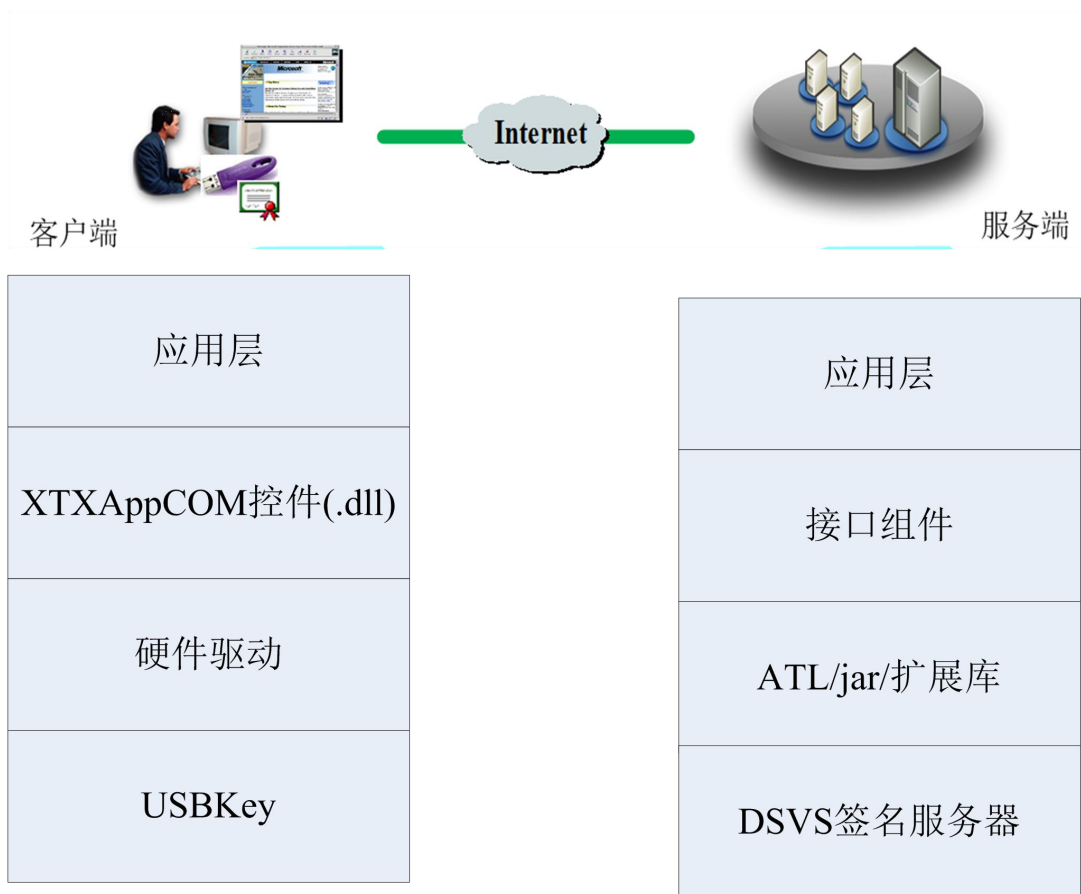
XTXAppCOM.dll 通过证书应用环境 V2.4 安装。

XTXApp 对应的 CLSID:3F367B74-92D9-4C5E-AB93-234F8A91D5E6，对应的 ActiveX 对象名称为“XTXAppCOM.XTXApp”，设备插拔时响应 OnUsbkeyChange 事件。

1.1 产品架构

XTXAppCOM 在结构上需要和服务端配合使用，分别为业务系统的客户端和服务端提供相应的安全服务。

客户端接口既可以内嵌到 Web 页面中，也可以被专用 Client 程序调用，对用户的使用是透明的；服务器端(DSVS)接口部署在应用服务器上，接受并处理由客户端发送过来的安全认证、数据加解密和签名验证等系列安全处理请求，为应用系统提供安全保护。



图表 1 XTXAppCOM 架构图

如上图所示，XTXAppCOM 其安全机制实现如下：

- 1) 用户和应用服务器需要从公信第三方机构，如北京数字证书认证中心，获取代表各自身份的数字证书（客户端数字证书和服务器数字证书）；
- 2) 用户和应用服务器两端安全组件和安全控件互相验证各自证书，确认各自身份的真实性；
- 3) 采用 PKI 技术体系下的数字信封技术和数字签名技术保证用户提交数据的安全性和应用服务器返回信息的机密性，防止信息泄密、篡改及抵赖。

1.2 功能特点

它通过与服务端（DSVS）的配合，实现安全登录、信息的完整性、机密性和

不可抵赖性，从而确保信息安全。客户端（通常是用户）和服务端需要从公信第三方（如北京数字证书认证中心）获取代表各自身份的数字证书；两端互相验证各自证书，来确认各自身份的真实性；通过 PKI 体系下的数字信封技术和数字签名技术，分别保证用户提交数据的安全性和 WEB 服务器返回信息的机密性，防止信息泄密、篡改及抵赖。

产品特点：

- **合法性：**采用权威第三方认证机构颁发的数字证书，符合《电子签名法》
- **规范性：**本接口的实现依照国密局颁发的《证书综合应用服务接口规范》

第 2 章 XTXApp 接口说明

XTXApp 接口实现了国密定义的证书应用组合服务接口，并且自定义了发证过程中需要的接口。

2.1 应用接口

2.1.1 获取控件版本号

接口原型	BSTR SOF_GetVersion()	
功能描述	获取控件的版本号。	
参数	无	
返回值	控件版本号	
备注		

2.1.2 设置签名算法

接口原型	LONG SOF_SetSignMethod(LONG SignMethod)	
功能描述	设置签名算法。	
参数	SignMethod	[IN] 签名算法
返回值	成功返回 0，失败返回其他	
备注	目前可支持的签名算法 详见附录。如果不调用本接口则RSA类型的证书默认使用SHA1_RSA算法，SM2类型的证书默认使用SM3_SM2算法。如果调用	

	了本接口，则不判断证书类型直接使用已经设置的算法。
--	---------------------------

2.1.3 获取签名算法

接口原型	LONG SOF_GetSignMethod()
功能描述	获取设置的签名算法。
参数	无
返回值	成功返回签名算法，失败返回 0
备注	如果之前没有设置过签名算法，则返回0

2.1.4 设置加密算法

接口原型	LONG SOF_SetEncryptMethod(LONG EncryptMethod)	
功能描述	设置加密算法。	
参数	EncryptMethod	[IN] 加密算法
返回值	成功返回 0，失败返回其他	
备注	目前可支持的加密算法 详见附录。如果不调用本接口，则默认算法为3DES 算法。	

2.1.5 获取加密算法

接口原型	LONG SOF_GetEncryptMethod()
功能描述	获取设置的加密算法。
参数	无

返回值	成功返回加密算法，失败返回 0
备注	如果之前没有设置过加密算法，则返回0

2.1.6 获取证书用户列表

接口原型	BSTR SOF_GetUserList()
功能描述	获取已安装的证书用户列表。
参数	无
返回值	证书用户列表
备注	<p>返回的证书用户列表格式为：用户名1 CertID1&&&用户名2 CertID2&&&用户名3 CertID3&&&...</p> <p>CertID是证书操作唯一标识，格式为 容器名称/设备序列号。通过CertID可以找到唯一的签名证书、加密证书，并使用对应密钥。</p>

2.1.7 导出用户证书

接口原型	BSTR SOF_ExportUserCert(BSTR CertID)		
功能描述	根据证书操作唯一标识获取 Base64 编码格式的证书。		
参数	<table><tr><td>CertID</td><td>[IN] 证书操作唯一标识，也支持只输入设备序列号</td></tr></table>	CertID	[IN] 证书操作唯一标识，也支持只输入设备序列号
CertID	[IN] 证书操作唯一标识，也支持只输入设备序列号		
返回值	成功返回 Base64 编码的证书，失败返回空		
备注	默认导出签名证书，无签名证书时导出加密证书		

2.1.8 导出用户加密证书

接口原型	BSTR SOF_ExportExChangeUserCert(BSTR CertID)
------	--

功能描述	根据证书操作唯一标识获取 Base64 编码格式的加密证书。	
参数	CertID	[IN] 证书操作唯一标识，也支持只输入设备序列号
返回值	成功返回 Base64 编码的证书，失败返回空	
备注		

2.1.9 校证书口令

接口原型	boolean SOF_Login(BSTR CertID, BSTR PassWd)	
功能描述	校证书口令	
参数	CertID	[IN] 证书操作唯一标识，也支持只输入设备序列号。
	PassWd	[IN] 证书口令。
返回值	正确返回 TRUE 失败返回 FALSE	
备注	本接口登录成功后，如果可以正确加载 DSVS 组件，则调用 DSVS 的更新证书接口对将要到期的证书进行更新。	

2.1.10 验证证书有效性

接口原型	LONG SOF_ValidateCert(BSTR Base64EncodeCert)	
功能描述	验证证书有效性	
参数	Base64EncodeCert	[IN] Base64 编码证书
返回值	正确返回 0 失败返回其他 -1 证书不被信任 -2 超过有效期范围	

	<p>-3 证书已作废</p> <p>-4 证书已冻结</p> <p>-5 证书未生效</p> <p>-6 其他错误</p>
备注	<p>本接口验证用户证书有效性，可信任的 CA 和 CRL 列表保存在控件本身所在的目录下 trust.pem 文件中，trust.pem 文件是 PEM 格式，可以存放多张 CA 证书和多个 CRL。如果可信的 CA 列表没有存放在 trust.pem 文件中，也可以通过 SetUserConfig 接口进行添加可信的 CA 或 CRL 等。详见设置用户配置信息接口说明。</p>

2.1.11 登出证书口令

接口原型	boolean SOF_Logout (BSTR CertID)	
功能描述	登出证书口令，再进行签名或者其他需要验证口令的接口时需要重新校证书口令。	
参数	CertID	[IN] 证书操作唯一标识，也支持只输入设备序列号。
返回值	正确返回 TRUE 失败返回 FALSE	
备注		

2.1.12 获取口令重试次数

接口原型	LONG SOF_GetPinRetryCount(BSTR CertID)
功能描述	获取证书口令重试次数

参数	CertID	[IN] 证书操作唯一标识，也支持只输入设备序列号。
返回值	>0 表示重试次数，=0 表示证书口令已被锁死，必须解锁后才能使用，<0 表示获取重试次数失败	
备注		

2.1.13 修改证书口令

接口原型	boolean SOF_ChangePassWd(BSTR CertID, BSTR OldPassWd , BSTR NewPassWd)	
功能描述	修改证书口令	
参数	CertID	[IN] 证书操作唯一标识，也支持只输入设备序列号。
	OldPassWd	[IN] 原始口令，长度必须是 6-16 位，并且是可见字符
	NewPassWd	[IN] 新口令，长度必须是 6-16 位，并且是可见字符
返回值	成功返回 TRUE，失败返回 FALSE	
备注		

2.1.14 获取证书基本信息

接口原型	BSTR SOF_GetCertInfo(BSTR Cert, short Type)	
功能描述	获取证书基本信息	
参数	Cert	[IN] Base64 编码的证书。
	Type	[IN] 信息类型
返回值	成功返回证书的基本信息，失败返回空	
备注	目前可支持的证书信息类型参数如下：	

1	证书版本号 返回"V1" 或"V2"或 "V3"
2	证书序列号
3	证书类型 返回"RSA"或"SM2"
4	颁发者国家名(C) 多个用逗号(,)分割
5	颁发者组织名(O) 多个用逗号(,)分割
6	颁发者部门名(OU) 多个用逗号(,)分割
7	颁发者省州名(ST) 多个用逗号(,)分割
8	颁发者通用名(CN) 多个用逗号(,)分割
9	颁发者城市名(L) 多个用逗号(,)分割
10	颁发者 EMAIL(E) 多个用逗号(,)分割
11	证书有效开始日期 格式 YYYYMMDDHHMMSS
12	证书有效结束日期 格式 YYYYMMDDHHMMSS
13	使用者国家名(C) 多个用逗号(,)分割
14	使用者组织名(O) 多个用逗号(,)分割
15	使用者部门名(OU) 多个用逗号(,)分割
16	使用者省州名(ST) 多个用逗号(,)分割
17	使用者通用名(CN) 多个用逗号(,)分割
18	使用者城市名(L) 多个用逗号(,)分割
19	使用者 EMAIL(E) 多个用逗号(,)分割
20	证书公钥 Base64 编码格式
21	颁发者 UID
22	使用者 UID

	23	密钥用法 证书中如果没有密钥用法扩展项 返回"0"，密钥用法为加密返回"1"，密钥用法为签名返回"2"，密钥用法既有加密也有签名返回"3"
	33	使用者 DN
	34	颁发者 DN

2.1.15 获取证书扩展信息

接口原型	BSTR SOF_GetCertInfoByOid(BSTR Cert, BSTR Oid)	
功能描述	根据 OID 获取证书的私有扩展信息	
参数	Cert	[IN] Base64编码的证书
	Oid	[IN] OID串 如"2.16.840.1.113732.2"
返回值	成功返回对应的扩展信息，失败返回空	
备注		

2.1.16 获取证书唯一实体标识

接口原型	BSTR SOF_GetCertEntity(BSTR Cert)	
功能描述	获取证书的唯一实体标识	
参数	Cert	[IN] Base64编码的证书
返回值	成功返回证书唯一实体标识，失败返回空	
备注		

2.1.17 数据签名

接口原型	BSTR SOF_SignData(BSTR CertID, BSTR InData)	
功能描述	对数据进行数字签名。	
参数	CertID	[IN] 证书操作唯一标识，也支持只输入设备序列号。
	InData	[IN] 签名原文
返回值	成功返回 Base64 编码的签名值，失败返回空	
备注	当算法为 RSA 时，遵循 PKCS#1；当算法为 SM2 时，遵循《SM2 密码使用规范》，调用本接口之前必须已登录	

2.1.18 数据验签

接口原型	boolean SOF_VerifySignedData(BSTR Cert, BSTR InData, BSTR SignValue)	
功能描述	验证数据签名。	
参数	Cert	[IN] Base64编码的证书
	InData	[IN] 签名原文
	SignValue	[IN] Base64编码的签名值
返回值	成功返回 TRUE，失败返回 FALSE	
备注	本接口与 SOF_SignData 对应使用	

2.1.19 文件签名

接口原型	BSTR SOF_SignFile(BSTR CertID, BSTR InFile)
功能描述	对文件进行数字签名。

参数	CertID	[IN] 证书操作唯一标识，也支持只输入设备序列号。
	InFile	[IN] 文件全路径
返回值	正确返回 Base64 编码的签名值，失败返回空	
备注	当算法为 RSA 时，遵循 PKCS#1；当算法为 SM2 时，遵循《SM2 密码使用规范》，调用本接口之前必须已登录	

2.1.20 文件验签

接口原型	boolean SOF_VerifySignedFile(BSTR Cert, BSTR InFile, BSTR SignValue)	
功能描述	验证文件签名。	
参数	Cert	[IN] Base64编码的证书
	InFile	[IN] 文件全路径
	SignValue	[IN] Base64编码的签名值
返回值	成功返回 TRUE，失败返回 FALSE	
备注	本接口与 SOF_SignFile 对应使用	

2.1.21 数字信封加密数据

接口原型	BSTR SOF_EncryptData(BSTR Cert, BSTR Indata)	
	BSTR SOF_EncryptDataEx(BSTR Cert, BSTR Indata)	
功能描述	对数据进行数字信封加密。	
参数	Cert	[IN] Base64编码的证书。
	Indata	[IN] 原始数据

返回值	成功返回 Base64 编码的密文数据，失败返回空
备注	使用临时产生的对称密钥加密数据,然后使用数字证书加密对称密钥(数字信封)。公钥算法为 RSA 时，遵循 PKCS#7；当公钥算法为 SM2 时，遵循《SM2 算法加密签名消息语法规范》。其中 SOF_EncryptDataEx 接口加密的数字信封和 SecX 互通。 SOF_EncryptData 是标准的数字信封。

2.1.22 数字信封解密数据

接口原型	BSTR SOF_DecryptData(BSTR CertID, BSTR Indata)	
功能描述	对数据进行数字信封解密。	
参数	CertID	[IN] 证书操作唯一标识，也支持只输入设备序列号。
	Indata	[IN] Base64编码的密文数据
返回值	成功返回明文数据，失败返回空	
备注	本接口与 SOF_EncryptData 或 SOF_EncryptDataEx 对应使用，调用本接口之前必须已登录。本接口 RSA 的设备仅支持 3DES 算法的数字信封解密，SM2 设备仅支持 SM1 和 SSF33 算法的解密。解密之前调用一下 SOF_SetEncryptMethod 设置一下算法。	

2.1.23 数据签名(P7 格式)

接口原型	BSTR SOF_SignMessage(short dwFlag, BSTR CertID, BSTR InData)	
功能描述	对数据进行数字签名。	
参数	dwFlag	[IN] 标识是否Detached。0表示Attached(带原文)，1表示

		Detached(不带原文)。
	CertID	[IN] 证书操作唯一标识，也支持只输入设备序列号。
	InData	[IN] 待签名的原始数据
返回值	成功返回 Base64 编码的签名值，失败返回空	
备注	对字符串数据进行数字签名，公钥算法为 RSA 时，遵循 PKCS#7 (P7 签名)； 当公钥算法为 SM2 时，遵循《SM2 算法加密签名消息语法规范》，调用本接口之前必须已登录	

2.1.24 数据验签(P7 格式)

接口原型	boolean SOF_VerifySignedMessage(BSTR MessageData, BSTR InData)	
功能描述	对数据进行验证签名。	
参数	MessageData	[IN] Base64编码的签名值。
	InData	[IN] 原始数据，如果签名中包含原文，忽略本参数
返回值	成功返回 TRUE，失败返回 FALSE	
备注	本接口与 SOF_SignMessage 对应使用。	

2.1.25 解析签名(P7 格式)

接口原型	BSTR SOF_GetInfoFromSignedMessage(BSTR SignedMessage, short type)	
功能描述	解析签名包内的信息，包括：原文、签名值、签名证书等信息。	
参数	SignedMessage	[IN] Base64编码的签名值。
	type	[IN] 类型

		取值范围： 1：原文 2：签名者证书 3：签名值
返回值	成功返回解析的信息，失败返回空	
备注		

2.1.26 产生随机数

接口原型	BSTR SOF_GenRandom(LONG RandomLen)	
功能描述	产生随机数。	
参数	RandomLen	[IN] 随机数的长度。
返回值	成功返回 Base64 编码的随机数，失败返回空	
备注		

2.1.27 公钥加密数据

接口原型	BSTR SOF_PubKeyEncrypt(BSTR sCert, BSTR sInData)	
功能描述	公钥加密数据	
参数	sCert	[IN] Base64 编码证书
	sInData	[IN] 原文数据
返回值	正确返回 Base64 编码的密文数据，失败返回空	
备注	如果 RSA 类型的证书 密钥长度为 1024 位的，明文数据不能超过 117 字节、	

	密钥长度为 2048 位的，明文长度不能超过 245 字节 如果 SM2 类型的证书 长度不能超过 1024 字节
--	--

2.1.28 私钥解密数据

接口原型	BSTR SOF_PriKeyDecrypt(BSTR CertID, BSTR sInData)	
功能描述	私钥解密数据	
参数	CertID	[IN] 证书操作唯一标识，也支持只输入设备序列号
	sInData	[IN] Base64 编码的密文数据
返回值	正确返回明文数据，失败返回空	
备注	调用本接口之前必须已登录。本接口只实现了 RSA 类型的容器解密接口， SM2 类型的设备不支持。	

2.1.29 对称加密数据

接口原型	BSTR SOF_SymEncryptData(BSTR sKey, BSTR indata)	
功能描述	对称加密数据	
参数	sKey	[IN] Base64 编码的对称密钥
	indata	[IN] 原文数据
返回值	正确返回 Base64 编码的密文数据，失败返回空	
备注		

2.1.30 对称解密数据

接口原型	BSTR SOF_SymDecryptData(BSTR sKey, BSTR indata)	
功能描述	对称解密数据	
参数	sKey	[IN] Base64 编码的对称密钥
	indata	[IN] Base64 编码的密文
返回值	正确返回明文数据，失败返回空	
备注		

2.1.31 对称加密文件

接口原型	boolean SOF_SymEncryptFile(BSTR sKey, BSTR inFile, BSTR outFile)	
功能描述	对称加密文件	
参数	sKey	[IN] Base64 编码的对称密钥
	inFile	[IN] 明文文件全路径
	outFile	[IN] 加密后的文件全路径
返回值	成功返回 TRUE，失败返回 FALSE	
备注	<p>如果密文文件所在的目录不存在，本接口返回错误，不会自动创建目录。</p> <p>考虑访问本地文件存在安全问题，控件会保留文件名，但是把加密后的文件路径重定向到 系统盘:\FakePath 目录下。例如加密后的文件全路径是 "D:\1.txt"，系统盘是 C 盘，则控件写文件全路径重定向为 "C:\FakePath\1.txt"。该设置可以通过 2.4.9 节 SetUserConfig 来设置控件写文件时不重定向</p>	

2.1.32 对称解密文件

接口原型	boolean SOF_SymDecryptFile(BSTR sKey, BSTR inFile, BSTR outFile)	
功能描述	对称解密文件	
参数	sKey	[IN] Base64 编码的对称密钥
	inFile	[IN] 密文文件全路径
	outFile	[IN] 解密后的文件全路径
返回值	成功返回 TRUE，失败返回 FALSE	
备注	<p>如果解密后的文件所在的目录不存在，本接口返回错误，不会自动创建目录。</p> <p>考虑访问本地文件存在安全问题，控件会保留文件名，但是把解密后的文件路径重定向到 系统盘\FakePath 目录下。例如解密后的文件全路径是 "D:\1.txt"，系统盘是 C 盘，则控件写文件全路径重定向为 "C:\FakePath\1.txt"。该设置可以通过 2.4.9 节 SetUserConfig 来设置控件写文件时不重定向</p>	

2.1.33 Base64 编码

接口原型	BSTR SOF_Base64Encode(BSTR sIndata)	
功能描述	对数据进行 Base64 编码	
参数	sIndata	[IN] 原始数据
返回值	正确返回 Base64 编码的数据，失败返回空	
备注		

2.1.34 Base64 解码

接口原型	BSTR SOF_SOF_Base64Decode(BSTR sIndata)	
功能描述	对 Base64 编码的数据进行解码	
参数	sIndata	[IN] Base64 编码的数据
返回值	正确返回明文数据，失败返回空	
备注		

2.1.35 数据摘要

接口原型	BSTR SOF_HashData(LONG hashAlg, BSTR sInData)	
功能描述	对数据做摘要	
参数	hashAlg	[IN] 摘要算法，详见附录
	sIndata	[IN] 原始数据
返回值	正确返回 Base64 编码的数据，失败返回空	
备注		

2.1.36 文件摘要

接口原型	BSTR SOF_HashFile(LONG hashAlg, BSTR inFile)	
功能描述	对文件做摘要	
参数	hashAlg	[IN] 摘要算法，详见附录
	inFile	[IN] 文件全路径

返回值	正确返回 Base64 编码的数据，失败返回空
备注	

2.1.37 文件 Base64 编码

接口原型	BSTR Base64EncodeFile(BSTR inFile)	
功能描述	对文件进行 Base64 编码	
参数	inFile	[IN] 文件全路径
返回值	正确返回 Base64 编码的文件数据，失败返回空	
备注		

2.1.38 Base64 解码扩展

接口原型	boolean Base64DecodeFile(BSTR sInData, BSTR outFile)	
功能描述	对输入数据进行 Base64 编码并保存到文件	
参数	sInData	[IN] Base64 编码的数据
	outFile	[IN] 文件全路径
返回值	成功返回 TRUE，失败返回 FALSE	
备注	考虑访问本地文件存在安全问题，控件会保留文件名，但是把写路径重定向到系统盘:\FakePath 目录下。例如文件全路径是"D:\1.txt"，系统盘是 C 盘，则控件写文件全路径重定向为"C:\FakePath\1.txt"。该设置可以通过 2.4.9 节 SetUserConfig 来设置控件写文件时不重定向	

2.1.39 获取最后一次出错码

接口原型	LONG SOF_GetLastError()
功能描述	获取最后一次出错的错误代码
参数	无
返回值	错误码
备注	如果从未出错，返回 0

2.1.40 获取最后一次出错信息

接口原型	BSTR SOF_GetLastErrMsg()
功能描述	获取最后一次出错的错误描述
参数	无
返回值	错误描述
备注	如果从未出错，返回空

2.1.41 OTP 获取挑战码

接口原型	BSTR OTP_GetChallengeCode(BSTR CertID)	
功能描述	获取挑战码，该接口仅 OTP-KEY 支持	
参数	CertID	[IN] 证书操作唯一标识，也支持只输入设备序列号
返回值	成功返回 Base64 编码格式的挑战码，失败返回空	
备注	非 OTP 设备，该接口返回空	

2.1.42 计算 HMAC

接口原型	BSTR SOF_HMAC(LONG hashid, BSTR key, BSTR indata)	
功能描述	计算数据的 HMAC 值	
参数	hashid	[IN] 摘要算法，详见附录
	key	[IN] Base64 编码的密钥
	indata	[IN] 原文数据
返回值	成功返回 Base64 编码格式的 HMAC 值，失败返回空	
备注		

2.1.43 明文私钥签名

接口原型	BSTR SOF_SignDataByPriKey(BSTR sPriKey,BSTR sCert,BSTR sInData)	
功能描述	利用传入的私钥对数据做签名	
参数	sPriKey	[IN] Base64 编码格式的明文私钥
	sCert	[IN] 签名私钥对应的证书，可以为空
	sInData	[IN] 原文数据
返回值	成功返回 Base64 编码格式的签名值，失败返回空	
备注		

2.1.44 对摘要数据签名

接口原型	BSTR SOF_SignHashData(BSTR CertID, BSTR b64ashData, LONG hashAlg)
------	---

功能描述	对原文的摘要值做签名	
参数	CertID	[IN] 证书操作唯一标识，也支持只输入设备序列号。
	b64ashData	[IN] Base64编码格式的摘要值
	hashAlg	[IN] 摘要算法，详见附录算法定义
返回值	成功返回 Base64 编码格式的签名值，失败返回空	
备注		

2.1.45 对摘要数据验签

接口原型	boolean SOF_VerifySignedHashData(BSTR Cert,BSTR b64ashData, BSTR SignValue, LONG hashAlg)	
功能描述	验证数据签名。	
参数	Cert	[IN] Base64编码的证书
	b64ashData	[IN] Base64编码格式的摘要值
	SignValue	[IN] Base64编码的签名值
	hashAlg	[IN] 摘要算法，详见附录算法定义
返回值	成功返回 TRUE ，失败返回 FALSE	
备注	本接口与 SOF_SignHashData 对应使用	

2.1.46 XML 数据签名

接口原型	BSTR SOF_SignDataXML(BSTR CertID, BSTR InData)
功能描述	对数据进行数字签名。

参数	CertID	[IN] 证书操作唯一标识，也支持只输入设备序列号。
	InData	[IN] 待签名的XML数据
返回值	成功返回 XML 签名值，失败返回空	
备注	XML 签名标准遵循 RFC3275	

2.1.47 XML 验签

接口原型	boolean SOF_VerifySignedDataXML(BSTR InData)	
功能描述	对 XML 签名数据进行验证签名。	
参数	InData	[IN] XML签名数据
返回值	成功返回 TRUE，失败返回 FALSE	
备注	本接口与 SOF_SignDataXMLe 对应使用。	

2.1.48 解析 XML 签名信息

接口原型	BSTR SOF_GetXMLSignatureInfo(BSTR XMLSignedData, SHORT type)	
功能描述	解析 XML 签名包内的信息，包括：原文、摘要、签名值、签名证书、摘要算法、签名算法等信息。	
参数	XMLSignedData	[IN] XML签名数据。
	type	[IN] 类型 取值范围： 1：原文 2：摘要

		3: 签名值 4: 签名证书 5: 摘要算法 6: 签名算法
返回值	成功返回解析的信息，失败返回空	
备注		

2.1.49 门限拆分

接口原型	BSTR SOF_SecertSegment(BSTR Secert, SHORT m, SHORT n, SHORT k)	
功能描述	门限拆分。	
参数	Secert	[IN] 待拆分的数据（不大于64字节）
	m	[IN] 拆分的份数（不能小于3，不能大于8）
	n	[IN] 最少恢复份数（不能小于2，不能小于 $m/2 + 1$ ，不能大于m，一般比m小）
	k	[IN] 恢复时必须的份数（大于或等于0并且不能大于n）
返回值	成功返回拆分的值 格式为 sec1&&&sec2&&&sec3&&&...secn&&& 失败返回空串	
备注	恢复时必须的值放到返回值的前半部分	

2.1.50 门限恢复

接口原型	BSTR SOF_SecertRecovery([in] BSTR Seg)
功能描述	门限恢复。

参数	Seg	[IN] 待恢复的数据 格式为sec1&&&sec2...secn&&&
返回值	成功返回恢复的数据，失败返回空串	
备注	恢复时必须的值放到返回值的前半部分	

2.2 发证接口

2.2.1 获取设备数量

接口原型	LONG GetDeviceCount()	
功能描述	获取当前存在的可支持的设备数量	
参数	无	
返回值	设备数量	
备注		

2.2.2 获取设备数量扩展

接口原型	LONG GetDeviceCountEx(LONG type)	
功能描述	获取当前存在的可支持的指定设备（硬设备或软设备）数量	
参数	type	<p>[IN] 设备类型</p> <p>取值范围为：</p> <p>1 表示获取硬设备的数量；</p> <p>2 表示获取软设备的数量；</p> <p>3 表示获取硬设备和软设备的数量，功能和 GetDeviceCount 一致。</p>
返回值	指定类型的设备数量	

备注	
----	--

2.2.3 获取所有的设备序列号

接口原型	BSTR GetAllDeviceSN()
功能描述	获取当前存在的可支持的所有设备的序列号，每个设备序列号以分号(;)结尾
参数	无
返回值	所有设备序列号的组合
备注	当前存在所有设备序列号，例如"11111;22222;"

2.2.4 获取所有的设备序列号扩展

接口原型	BSTR GetAllDeviceSNEx(LONG type)	
功能描述	获取当前存在的可支持的所有设备的序列号，每个设备序列号以分号(;)结尾	
参数	type	<p>[IN] 设备类型</p> <p>取值范围为：</p> <p>1 表示获取所有硬设备的序列号；</p> <p>2 表示获取所有软设备的序列号；</p> <p>3 表示获取硬设备和软设备的序列号，功能和 GetAllDeviceSN 一致。</p>
返回值	所有设备序列号的组合	
备注	当前存在所有指定设备序列号，例如"11111;22222;"	

2.2.5 根据索引获取设备序列号

接口原型	BSTR GetDeviceSNByIndex(LONG iIndex)	
功能描述	根据索引获取设备的序列号	
参数	iIndex	[IN] 索引 从 0 开始，不能大于当前存在的设备数量-1
返回值	成功返回设备序列号，失败返回空	
备注	当前存在所有设备序列号，例如"11111;22222;"	

2.2.6 判断设备是否存在

接口原型	boolean IsDeviceExist(BSTR sDeviceSN)	
功能描述	判断设备是否存在	
参数	sDeviceSN	[IN] 设备序列号
返回值	存在返回 TRUE，不存在返回 FALSE	
备注		

2.2.7 获取设备详细信息

接口原型	BSTR GetDeviceInfo(BSTR sDeviceSN, LONG iType)	
功能描述	获取设备详细信息	
参数	sDeviceSN	[IN] 设备序列号
	iType	[IN] 信息类型
返回值	成功返回设备信息，失败返回空	
备注	目前可支持的类型参数如下：	

	0x00000001	设备标签
	0x00000002	设备空余空间
	0x00000003	设备序列号
	0x00000004	设备类型 返回"RSA"或"SM2"
	0x00000005	获取证书密码重试次数
	0x00000007	设备类型 返回"HARD"或"SOFT"
	0x00000008	对应动态库的名称
	0x00000009	CSP 名称（只针对 RSA 类型的设备，国密接口的设备 返回为空
	0x00000073	设备类型 RSA 设备返回"10" SM2 设备返回"20"
	0x00000074	设备的 VID_PID （16 进制数据）

2.2.8 修改管理员口令

接口原型	boolean ChangeAdminPass(BSTR sDeviceSN, BSTR sOldPass, BSTR sNewPass)	
功能描述	修改设备管理员口令	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
	sOldPass	[IN] 原始管理员口令，长度必须是 6-24 位，并且是可见字符
	sNewPass	[IN] 新管理员口令，长度必须是 6-24 位，并且是可见字符
返回值	成功返回 TRUE，失败返回 FALSE	

备注	
----	--

2.2.9 解锁用户口令

接口原型	boolean UnlockUserPass(BSTR sDeviceSN, BSTR sAdminPass, BSTR NewUserPass)	
功能描述	解锁用户口令	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
	sAdminPass	[IN] 管理员口令，长度必须是 6-16 位，并且是可见字符
	NewUserPass	[IN] 新用户口令，长度必须是 6-16 位，并且是可见字符
返回值	成功返回 TRUE，失败返回 FALSE	
备注		

2.2.10 解锁用户口令 Ex

接口原型	boolean UnlockUserPassEx(BSTR sDeviceSN, BSTR sAdminPass, BSTR NewUserPass)	
功能描述	解锁用户口令	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
	sAdminPass	[IN] 管理员口令，长度必须是 6-16 位，并且是可见字符
	NewUserPass	[IN] 编码后的新用户口令，一般与管理员口令一致。
返回值	成功返回 TRUE，失败返回 FALSE	
备注		

2.2.11 产生密钥对

接口原型	boolean GenerateKeyPair(BSTR sDeviceSN, BSTR sContainerName, LONG iKeyType, boolean bSign)	
功能描述	产生密钥对	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
	sContainerName	[IN] 容器名称，必须为可见字符，不能包含 ‘/’ 字符 长度在 1-32 位之间
	iKeyType	[IN] 密钥类型 1: RSA 1024 2: RSA 2048 3 SM2 256
	bSign	[IN] 签名或加密 TRUE 标识签名 FALSE 标识加密
返回值	成功返回 TRUE，失败返回 FALSE	
备注	调用本接口前必须已登录 如果容器已经存在，本接口返回失败	

2.2.12 导出公钥

接口原型	BSTR ExportPubKey(BSTR sDeviceSN, BSTR sContainerName, boolean bSign)	
功能描述	导出公钥	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
	sContainerName	[IN] 容器名称
	bSign	[IN] 签名或加密 TRUE 标识签名 FALSE 标识加密
返回值	成功返回 Base64 编码的公钥，失败返回空	
备注		

2.2.13 导出 PKCS10 证书请求

接口原型	BSTR ExportPKCS10(BSTR sDeviceSN, BSTR sContainerName, BSTR sDN, boolean bSign)	
功能描述	导出 PKCS10 证书请求	
参数	sDeviceSN	设备序列号，也支持输入 CertID，内部会做解析
	sContainerName	[IN] 容器名称
	sDN	[IN] DN 信息，形如 "CN=name, OU=BJCA R&D, O=BJCA, L=\\, 北京, ST=bj, C=CN, E=mail@bjca.com," 如果 DN 项中有逗号(,)，前边加上\\进行转义
	bSign	[IN] TRUE 表示用签名密钥产生 PKCS10 FALSE 表示用加密密钥产生 PKCS10
返回值	成功返回 Base64 编码的证书请求，失败返回空	
备注	调用本接口前必须已登录	

2.2.14 导入签名证书

接口原型	boolean ImportSignCert(BSTR sDeviceSN, BSTR sContainerName, BSTR sCert)	
功能描述	导入签名证书	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
	sContainerName	[IN] 容器名称
	sCert	[IN] Base64 编码的签名证书
返回值	成功返回 TRUE，失败返回 FALSE	
备注	调用本接口前必须已登录	

2.2.15 导入加密证书和加密密钥对

接口原型	boolean ImportEncCert(BSTR sDeviceSN,BSTR sContainerName, BSTR sCert, BSTR sPriKeyCipher)	
功能描述	导入加密证书和加密密钥对	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
	sContainerName	[IN] 容器名称
	sCert	[IN] Base64 编码的加密证书
	sPriKeyCipher	[IN] Base64 编码的加密密钥密文
返回值	成功返回 TRUE，失败返回 FALSE	
备注	调用本接口前必须已登录 加密证书和加密密钥对不能全部为空	

2.2.16 导入加密证书和加密密钥对扩展

接口原型	boolean ImportEncCertEx(BSTR sDeviceSN,BSTR sContainerName, BSTR sCert, BSTR sPriKeyCipher, LONG ulSymAlg)	
功能描述	导入加密证书和加密密钥对	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
	sContainerName	[IN] 容器名称
	sCert	[IN] Base64 编码的加密证书
	sPriKeyCipher	[IN] Base64 编码的加密密钥密文
	ulSymAlg	[IN] 对称算法，具体值见附录
返回值	成功返回 TRUE，失败返回 FALSE	
备注	调用本接口前必须已登录	

	加密证书和加密密钥对不能全部为空
--	------------------

2.2.17 枚举文件

接口原型	BSTR EnumFilesInDevice(BSTR sDeviceSN)	
功能描述	枚举设备内的所有文件	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
返回值	成功返回所有文件名称，失败或设备内不存在容器时返回空 返回格式为 file1&&&file2&&&file3&&&...fileN&&&	
备注		

2.2.18 读文件

接口原型	BSTR ReadFile(BSTR sDeviceSN, BSTR sFileName)	
功能描述	读取文件	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
	sFileName	[IN] 文件名称
返回值	成功返回读取的文件内容，失败返回空	
备注		

2.2.19 读文件 Ex

接口原型	BSTR ReadFileEx(BSTR sDeviceSN, BSTR sFileName)	
功能描述	读取文件，返回文件的 Base64 编码后的数据，如果文件内容为二进制数据，	

	推荐调用本接口而不是调用 ReadFile 接口	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
	sFileName	[IN] 文件名称
返回值	成功返回读取的文件内容做 Base64 编码后的数据，失败返回空	
备注		

2.2.20 写文件

接口原型	boolean WriteFile(BSTR sDeviceSN, BSTR sFileName, BSTR sContent, boolean bPrivate)	
功能描述	写文件	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
	sFileName	[IN] 文件名称
	sContent	[IN] 写入的内容
	bPrivate	[IN] 是否私有区文件，目前仅支持公有区文件
返回值	成功返回 TRUE，失败返回 FALSE	
备注	bPrivate 为 true 时 需要登录	

2.2.21 写文件 Ex

接口原型	boolean WriteFileEx(BSTR sDeviceSN, BSTR sFileName, BSTR sContent)	
功能描述	写文件，传入的数据必须是 Base64 编码格式的。如果写入文件的内容为二进制数据，推荐调用本接口而不是调用 WriteFile 接口	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析

	sFileName	[IN] 文件名称
	sContent	[IN] 写入的内容 经过 Base64 编码的数据
返回值	成功返回 TRUE，失败返回 FALSE	
备注		

2.2.22 获取容器数量

接口原型	LONG GetContainerCount(BSTR sDeviceSN)	
功能描述	获取容器数量	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
返回值	成功返回容器数量，失败返回负数	
备注		

2.2.23 获取所有容器名称

接口原型	BSTR SOF_GetAllContainerName(BSTR sDeviceSN)	
功能描述	获取所有容器名称	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
返回值	成功返回所有容器名称，失败或设备内不存在容器时返回空 返回格式为 container1&&&container2&&&container3&&&...	
备注		

2.2.24 判断容器是否存在

接口原型	boolean IsContainerExist(BSTR sDeviceSN, BSTR sContainerName)	
功能描述	判断容器是否存在	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
	sContainerName	[IN] 容器名称
返回值	容器存在返回 TRUE，容器不存在返回 FALSE	
备注		

2.2.25 删除容器

接口原型	boolean DeleteContainer(BSTR sDeviceSN, BSTR sContainerName)	
功能描述	删除容器	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
	sContainerName	[IN] 容器名称
返回值	成功返回 TRUE，失败返回 FALSE	
备注	调用本接口前必须已登录	

2.2.26 删除最旧的一个容器

接口原型	boolean DeleteOldContainer(BSTR sDeviceSN)	
功能描述	删除最旧的一个容器	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析

返回值	成功返回 TRUE，失败返回 FALSE
备注	调用本接口前必须已登录

2.2.27 格式化设备

接口原型	boolean InitDevice(BSTR sDeviceSN, BSTR sAdminPass)	
功能描述	格式化设备，调用本接口后会清除设备内的所有证书、密钥和文件信息	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
	sAdminPass	[IN] 管理员口令
返回值	成功返回 TRUE，失败返回 FALSE	
备注		

2.2.28 格式化设备 Ex

接口原型	boolean InitDeviceEx(BSTR sDeviceSN, BSTR sAdminPass, BSTR sUserPin, BSTR sKeyLabel, LONG adminPinMaxRetry, LONG userPinMaxRetry)	
功能描述	格式化设备，调用本接口后会清除设备内的所有证书、密钥和文件信息	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
	sAdminPass	[IN] 管理员口令
	sUserPin	[IN] 用户口令
	sKeyLabel	[IN] 设备标签
	adminPinMaxRetry	[IN] 管理员口令最多重试次数
	userPinMaxRetry	[IN] 用户口令最多重试次数

返回值	成功返回 TRUE，失败返回 FALSE
备注	

2.2.29 获取当前在用容器名称(读取 ENVSN)

接口原型	BSTR GetENVSN(BSTR sDeviceSN)	
功能描述	获取当前在用的容器名称	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
返回值	成功返回容器名称，失败返回空	
备注	本接口不判断容器名称在设备内是否存在	

2.2.30 设置当前在用容器名称(写入 ENVSN)

接口原型	boolean SetENVSN(BSTR sDeviceSN, BSTR sENVSN)	
功能描述	设置当前在用的容器名称	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
	sENVSN	[IN] 容器名称
返回值	成功返回 TRUE，失败返回 FALSE	
备注	本接口判断容器名称在设备内是否存在，如果容器不存在返回 FALSE	

2.2.31 更新证书

接口原型	LONG SOF_UpdateCert(BSTR CertID, LONG type)
功能描述	更新用户即将到期的证书

参数	CertID	[IN] CertID
	type	[IN] 是否弹框提示用户 1 表示提示用户 0 表示不提示用户
返回值	成功返回 0 失败返回非 0 -1 表示证书未即将过期，不用更新 -2 表示调用 CSS 控件接口失败 -3 表示其他错误，可通过获取最后一次出错错误码获取到错误信息	
备注	如果 type 值设置为 1，则通过系统的 MessageBox 进行错误提示，提示的内容包括： 1.证书在有效期之内不用更新 2.提交了证书请求，等待证书签发成功 3.调用 CSS 接口失败 4.打开设备，打开应用等失败 5.创建新容器、生成密钥对失败 6.公钥、证书等编解码失败 7.新证书尚未签发完成，请稍后几天重试 8.解析 CSS 服务器返回的数据失败 9.对 CSS 返回的数据解码失败 10.导入签名或加密证书失败 11.修改管理员口令 12.记录当前在用容器失败	

2.3 二进制数据接口

2.3.1 数据签名

接口原型	BSTR SOF_SignBinaryData(BSTR CertID, VARIANT InData)	
功能描述	对数据进行数字签名。	
参数	CertID	[IN] 证书操作唯一标识，也支持只输入设备序列号。
	InData	[IN] 签名原文
返回值	成功返回 Base64 编码的签名值，失败返回空	
备注	当算法为 RSA 时，遵循 PKCS#1；当算法为 SM2 时，遵循《SM2 密码使用规范》，调用本接口之前必须已登录	

2.3.2 数据验签

接口原型	boolean SOF_VerifySignedBinaryData(BSTR Cert,VARIANT InData,BSTR SignValue)	
功能描述	验证数据签名。	
参数	Cert	[IN] Base64编码的证书
	InData	[IN] 签名原文
	SignValue	[IN] Base64编码的签名值
返回值	成功返回 TRUE，失败返回 FALSE	
备注	本接口与 SOF_SignBinaryData 对应使用	

2.3.3 数字信封加密数据

接口原型	BSTR SOF_EncryptBinaryData(BSTR Cert, VARIANT Indata)	
	BSTR SOF_EncryptBinaryDataEx(BSTR Cert, VARIANT Indata)	
功能描述	对数据进行数字信封加密。	
参数	Cert	[IN] Base64编码的证书。
	Indata	[IN] 原始数据
返回值	成功返回 Base64 编码的密文数据，失败返回空	
备注	使用临时产生的对称密钥加密数据,然后使用数字证书加密对称密钥(数字信封)。公钥算法为 RSA 时,遵循 PKCS#7;当公钥算法为 SM2 时,遵循《SM2 算法加密签名消息语法规范》。其中 SOF_EncryptDataEx 接口加密的数字信封和 SecX 互通。 SOF_EncryptData 是标准的数字信封。	

2.3.4 数字信封解密数据

接口原型	VARIANT SOF_DecryptBinaryData(BSTR CertID, BSTR Indata)	
功能描述	对数据进行数字信封解密。	
参数	CertID	[IN] 证书操作唯一标识，也支持只输入设备序列号。
	Indata	[IN] Base64编码的密文数据
返回值	成功返回明文数据，失败返回空	
备注	本接口与 SOF_EncryptBinaryData 或 SOF_EncryptBinaryDataEx 对应使用，调用本接口之前必须已登录。本接口 RSA 的设备仅支持 3DES 算法的数字信封解密，SM2 设备仅支持 SM1 和 SSF33 算法的解密。解密之前调用一下	

	SOF_SetEncryptMethod 设置一下算法。
--	------------------------------

2.3.5 数据签名(P7 格式)

接口原型	BSTR SOF_SignBinaryMessage(short dwFlag, BSTR CertID, VARIANT InData)	
功能描述	对数据进行数字签名。	
参数	dwFlag	[IN] 标识是否Detached。0表示Attached(带原文), 1表示Detached(不带原文)。
	CertID	[IN] 证书操作唯一标识, 也支持只输入设备序列号。
	InData	[IN] 待签名的原始数据
返回值	成功返回 Base64 编码的签名值, 失败返回空	
备注	对字符串数据进行数字签名, 公钥算法为 RSA 时, 遵循 PKCS#7 (P7 签名); 当公钥算法为 SM2 时, 遵循《SM2 算法加密签名消息语法规则》, 调用本接口之前必须已登录	

2.3.6 数据验签(P7 格式)

接口原型	boolean SOF_VerifySignedBinaryMessage(BSTR MessageData, VARIANT InData)	
功能描述	对数据进行验证签名。	
参数	MessageData	[IN] Base64编码的签名值。
	InData	[IN] 原始数据, 如果签名中包含原文, 忽略本参数
返回值	成功返回 TRUE, 失败返回 FALSE	
备注	本接口与 SOF_SignBinaryMessage 对应使用。	

2.3.7 公钥加密数据

接口原型	BSTR SOF_PubKeyBinaryEncrypt(BSTR sCert, VARIANT sInData)	
功能描述	公钥加密数据	
参数	sCert	[IN] Base64 编码证书
	sInData	[IN] 原文数据
返回值	正确返回 Base64 编码的密文数据，失败返回空	
备注	如果 RSA 类型的证书 密钥长度为 1024 位的，明文数据不能超过 117 字节、 密钥长度为 2048 位的，明文长度不能超过 245 字节 如果 SM2 类型的证书 长度不能超过 1024 字节	

2.3.8 私钥解密数据

接口原型	VARIANT SOF_PriKeyBinaryDecrypt(BSTR CertID, BSTR sInData)	
功能描述	私钥解密数据	
参数	CertID	[IN] 证书操作唯一标识，也支持只输入设备序列号
	sInData	[IN] Base64 编码的密文数据
返回值	正确返回明文数据，失败返回空	
备注	调用本接口之前必须已登录。本接口只实现了 RSA 类型的容器解密接口， SM2 类型的设备不支持。	

2.3.9 Base64 编码

接口原型	BSTR SOF_Base64BinaryEncode(VARIANT sldata)	
功能描述	对数据进行 Base64 编码	
参数	sldata	[IN] 原始数据
返回值	正确返回 Base64 编码的数据，失败返回空	
备注		

2.3.10 Base64 解码

接口原型	VARIANT SOF_SOF_Base64BinaryDecode(BSTR sldata)	
功能描述	对 Base64 编码的数据进行解码	
参数	sldata	[IN] Base64 编码的数据
返回值	正确返回明文数据，失败返回空	
备注		

2.3.11 数据摘要

接口原型	BSTR SOF_HashBinaryData(LONG hashAlg, VARIANT sldata)	
功能描述	对数据做摘要	
参数	hashAlg	[IN] 摘要算法，详见附录
	sldata	[IN] 原始数据
返回值	正确返回 Base64 编码的数据，失败返回空	

备注	
----	--

2.4 其他接口

2.4.1 检测软设备环境

接口原型	boolean CheckSoftDeviceEnv()
功能描述	检测软设备的运行环境是否正常
参数	无
返回值	成功返回 TRUE，失败返回 FALSE
备注	

2.4.2 创建一个软设备

接口原型	boolean CreateSoftDevice(BSTR sDeviceSN, BSTR sLabel)	
功能描述	创建一个软设备	
参数	sDeviceSN	[IN] 设备序列号，必须为可见字符，长度不能大于 128 位并且不能和硬件设备的序列号相同，调用时需要确保与硬件设备序列号不同，如果和某个硬件的设备序列号相同。在软证书和该硬件设备一起使用时会存在问题
	sLabel	[IN] 设备标签，长度为 1-32 字节的可见字符

返回值	成功返回 TRUE，失败返回 FALSE
备注	

2.4.3 删除一个软设备

接口原型	boolean DeleteSoftDevice (BSTR sDeviceSN, BSTR sPasswd)	
功能描述	删除一个软设备	
参数	sDeviceSN	[IN] 设备序列号
	sPasswd	[IN] 用户口令
返回值	成功返回 TRUE，失败返回 FALSE	
备注		

2.4.4 禁用或启用软设备

接口原型	boolean EnableSoftDevice(VARIANT_BOOL enable, BSTR sDeviceSN)	
功能描述	启用或禁用软设备	
参数	enable	[IN] TRUE 表示启用软设备 FALSE 表示禁用软设备
	sDeviceSN	[IN] 设备序列号，保留参数输入为空即可
返回值	成功返回 TRUE，失败返回 FALSE	
备注		

2.4.5 软设备备份

接口原型	BSTR SoftDeviceBackup(BSTR sDeviceSN, BSTR sPasswd)	
功能描述	备份一个软设备	
参数	sDeviceSN	[IN] 设备序列号
	sPasswd	[IN] 用户口令，也是备份口令。恢复软设备时需要用到
返回值	成功返回备份文件的路径，失败返回空	
备注		

2.4.6 软设备恢复

接口原型	boolean SoftDeviceRestore(BSTR sDeviceSN, BSTR sPasswd, BSTR sInFilePath)	
功能描述	备份一个软设备	
参数	sDeviceSN	[IN] 设备序列号
	sPasswd	[IN]备份口令
	sInFilePath	[IN] 备份文件的全路径
返回值	成功返回 TRUE，失败返回 FALSE	
备注		

2.4.7 将密钥和证书导入到软设备中

接口原型	boolean ImportKeyCertToSoftDevice(BSTR sDeviceSN, BSTR sContainerName, BSTR sPriKey, BSTR sCert, VARIANT_BOOL bSign)
功能描述	将明文私钥和证书导入到软设备中

参数	sDeviceSN	[IN] 设备序列号
	sContainerName	[IN] 容器名称
	sPriKey	[IN] Base64 编码格式的明文私钥
	sCert	[IN] Base64 编码格式的证书
	bSign	[IN] TRUE 表示签名位置 FALSE 表示加密位置
返回值	成功返回 TRUE，失败返回 FALSE	
备注		

2.4.8 导入 P12 证书到软设备中

接口原型	boolean ImportPfxToDevice(BSTR sDeviceSN, BSTR sContainerName, VARIANT_BOOL bSign, BSTR strPfx, BSTR strPfxPass)	
功能描述	导入P12证书到软设备中	
参数	sDeviceSN	[IN] 设备序列号，也支持输入 CertID，内部会做解析
	sContainerName	[IN] 容器名称
	bSign	[IN] 导入位置 TRUE：签名位置 FALSE：加密位置
	sPriKeyCipher	[IN] Base64 编码的 P12 证书
	strPfxPass	[IN] P12 证书的密码
返回值	成功返回 TRUE，失败返回 FALSE	
备注		

2.4.9 设置用户配置信息

接口原型	boolean SetUserConfig(LONG type, BSTR strConfig)
------	--

功能描述	设置用户信息		
参数	type	[IN] 设置的参数类型	
		1 表示禁用所有硬件设备	
	strConfig	2 表示启用或停用等待窗口	
		3 表示启用或停用验证 Ukey 驱动文件的代码签名	
		4 表示添加用户可信的 CA 证书	
		5 表示添加用户可信的 CRL	
		6 表示添加 P7B 证书链	
		7 表示启用或停用证书即将过期或已过期时的弹框提示	
		8 表示启用或停用写入本地文件时是否重定向	
		[IN] 设置的参数，具体传入的值和 type 参数有关系	
		type 值	对应参数
		1	只支持传入 "0" 或 "off" 其他参数无效
		2	传入"1"或"on"表示启用等待窗口 传入"0"或"off"表示停用等待窗口 默认是启用的
		3	传入"1"或"on"表示验证驱动库的代码签名 传入"0"或"off"表示不验证驱动库的代码签名 本配置目前不起作用
		4	Base64 编码格式的 CA 证书
		5	Base64 编码格式 CRL
		6	Base64 编码格式的 P7B 证书链

		7	<p>传入"1"或"on"表示证书即将过期或已过期时弹框提示</p> <p>传入"0"或"off"表示证书即将过期或已过期时不弹框提示</p> <p>默认是开启的</p>
		8	<p>传入"1"或"on"表示写入本地文件时，把文件路径重定向到"系统盘:\FakePath"目录下</p> <p>传入"0"或"off"表示写入本地文件时不做重定向</p> <p>默认是开启重定向</p> <p>影响到的接口包括：SymEncryptFile、SymDecryptFile、Base64DecodeFile</p>
返回值	成功返回 TRUE，失败返回 FALSE		
备注			

2.4.10 选择文件

接口原型	BSTR SelectFile()
功能描述	弹出选择文件的对话框，让用户选择一个文件。
参数	无
返回值	成功返回文件绝对路径，失败返回空
备注	

2.4.11 打开指定的路径

接口原型	OpenSpecifiedFolder(BSTR strFilePath)	
功能描述	打开指定的路径。	
参数	strFilePath	<p>[IN] 文件夹或文件全路径，</p> <p>1、如果指定的文件夹的全路径，只打开文件夹</p> <p>2、如果指定的是文件全路径，打开文件所在的文件夹并选中文件</p> <p>3、如果输入的参数为空，则会打开软证书备份的路径。软证书备份路径如果不存在，会弹出无法打开的提示。</p> <p>4、如果输入的路径或文件不存在，会弹出无法打开的提示。</p>
返回值	无返回值	
备注		

2.5 调用示例

2.5.1 JavaScript 调用方法

在此仅示例了产生随机数的功能，其他功能示例详见 Demo 中的 JSP、ASP 以及 ASP.NET 示例。

下面是简单的 HTML 示例，该示例中将加载控件放在了页面中。实际集成时一般将加载控件封装到一个 JavaScript 脚本中。详见 Demo。（本示例可兼容

Firefox、Chrome、Opera、Safari 等其他非 IE 浏览器)

```

<html>
<head>
<script language=JavaScript>
try {
    if (window.ActiveXObject || 'ActiveXObject' in window) {
        document.writeln("<OBJECT
classid='CLSID:3F367B74-92D9-4C5E-AB93-234F8A91D5E6' height=1 id=XTXAPP
style='HEIGHT: 1px; LEFT: 10px; TOP: 28px; WIDTH: 1px' VIEWASTEXT>");
        document.writeln("</OBJECT>");
        XTXAPP.SOF_GetVersion();
    } else {
        document.writeln("<embed id=XTXAPP0 type=application/x-xtx-axhost
clsid={3F367B74-92D9-4C5E-AB93-234F8A91D5E6}
event_OnUsbkeyChange=OnUsbKeyChange width=1 height=1 />");
        XTXAPP = document.getElementById("XTXAPP0");
    }
    XTXAPP.SOF_GetVersion();
}
catch(e) {
    alert("请检查证书应用环境是否正确安装!");
}

//非 IE 浏览器响应设备插拔执行的接口
function OnUsbKeyChange()
{
    alert("OnUsbKeyChange called!");
}
function GenRandom()
{
    var r = XTXAPP.SOF_GenRandom(24);
    alert(r); //打印随机数
}
</script>

//IE 浏览器中响应设备插拔事件
<SCRIPT LANGUAGE=javascript event=OnUsbKeyChange for=XTXAPP>
    alert("OnUsbKeyChange called!");
</SCRIPT>
</head>
<body>b
    <input type="button" value="产生随机数" onclick="return GenRandom()"
/><br><br>

```



```
</body>  
</html>
```

2.5.2 VC 调用方法

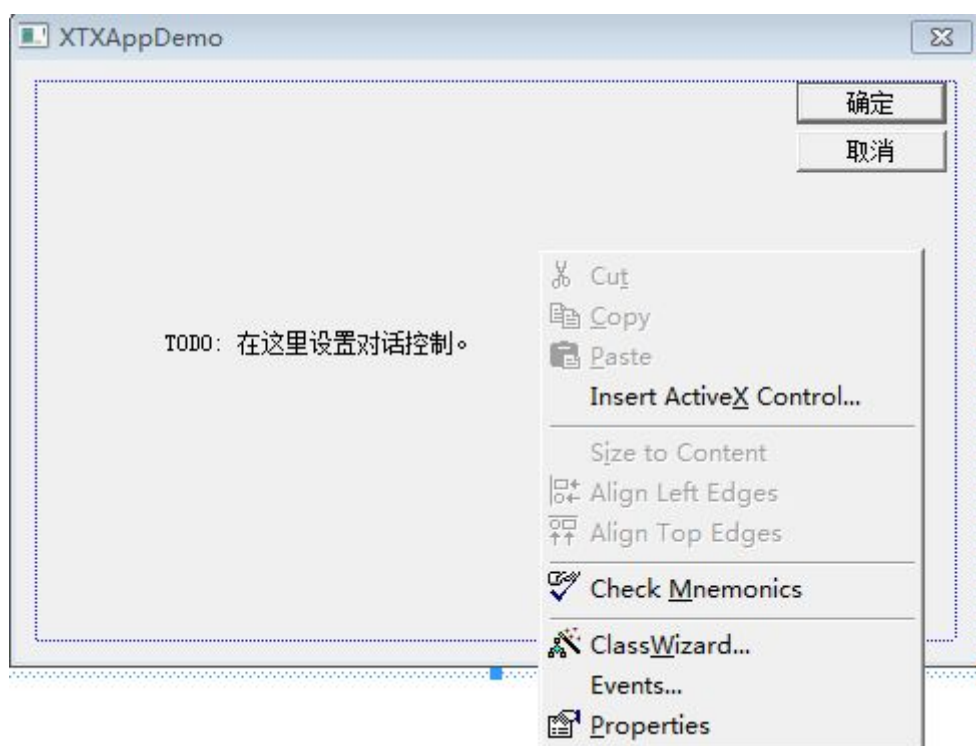
在此以 VC6.0 为例,介绍 VC 中如何调用 XTXApp 接口。其他 VS2005、2008、2010 等调用方法类似。

- 步骤一：新建 VC 工程

选择 MFC Appwizard(exe)->Dialog Based, 选择支持 ActiveX 控件

- 步骤二：在工程中加载 XTXApp 控件

在对话框编辑器中点击右键, 选择 Insert ActiveX Control..., 如下图



将 XTXApp 控件添加到工程中, 如下图, 选择 XTXApp Class, 单击 OK

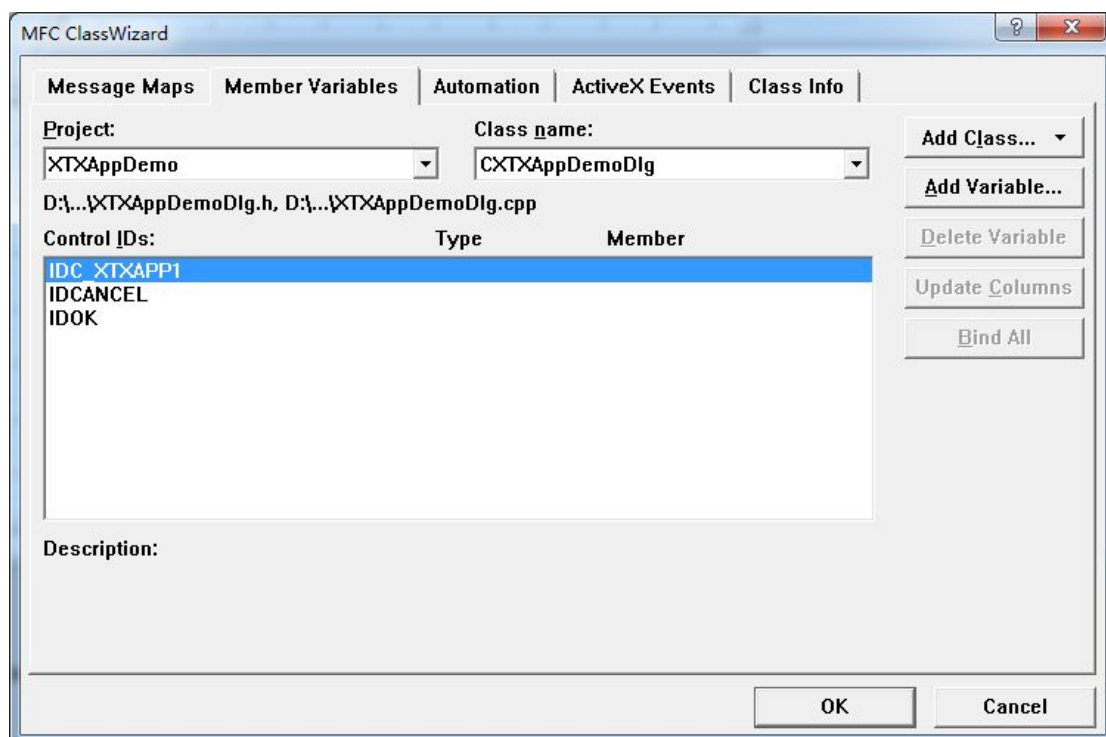


控件添加到工程如下图所示



- 步骤三：为 XTXApp 控件关联变量。

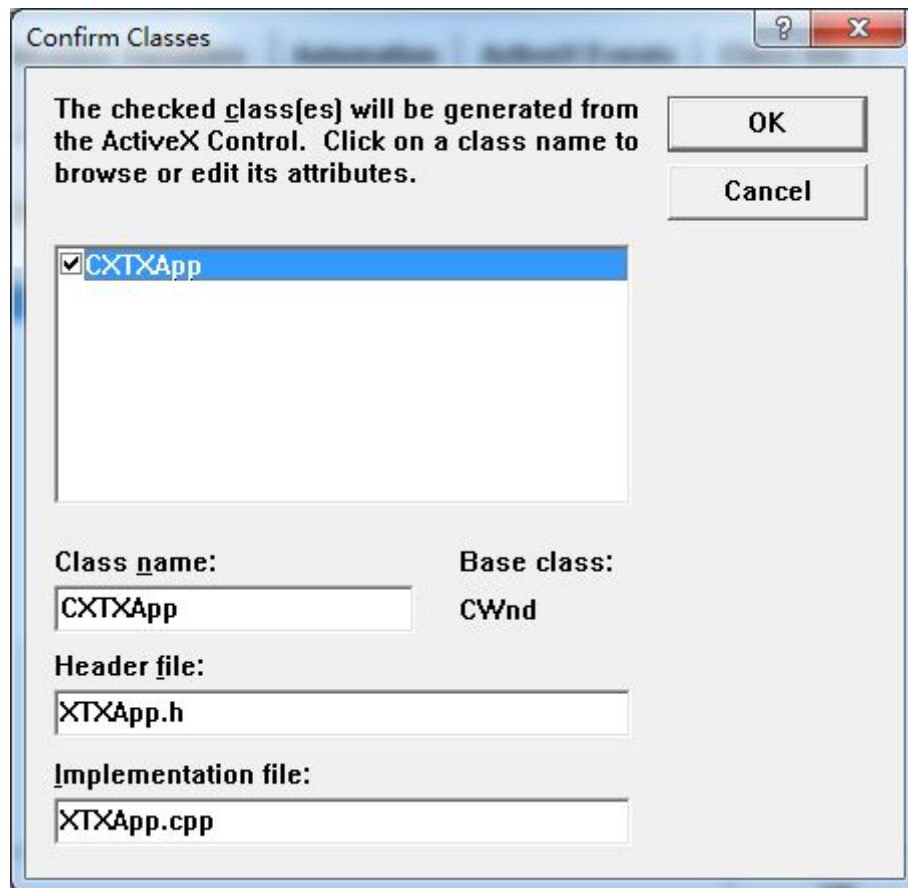
双击 XTXApp 控件或点击右键选择 ClassWizard, 如下图选择 Member Variable 选项卡



双击 IDC_XTXAPP1 或单击 Add Variable... 按钮，为控件关联变量。在关联变量之前，IDE 环境会有如下提示：



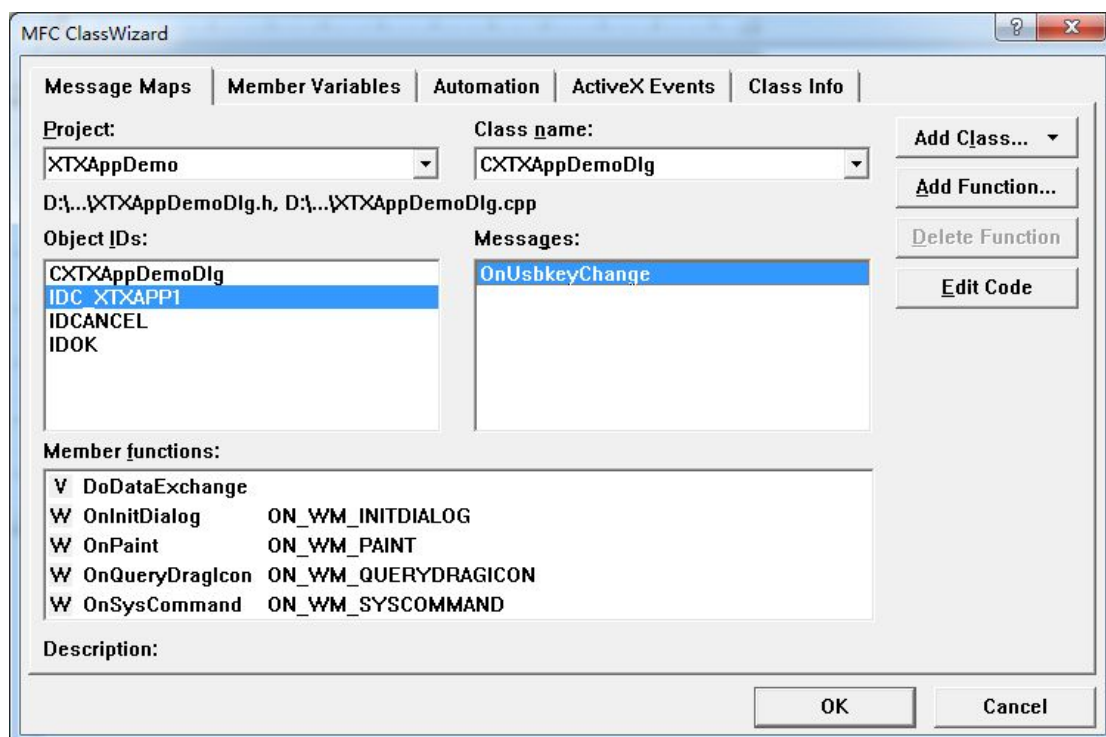
单击确定，如下图



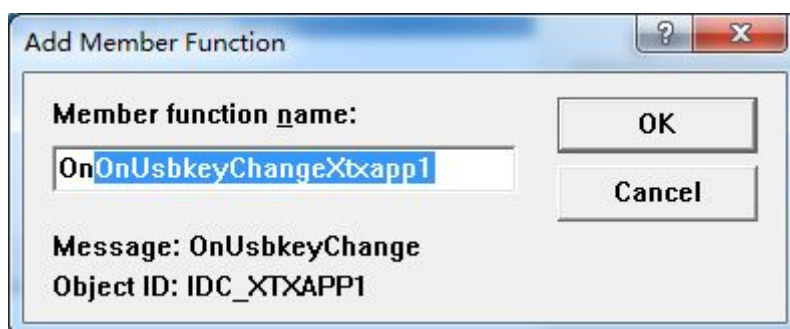
点击 OK 按钮，将 IDE 环境自动生成代码加入到工程中，并为控件关联变量。

- 步骤四：为响应设备插拔事件注册处理函数。

双击 XTXApp 控件或点击右键选择 ClassWizard，如下图选择 Message Maps 选项卡



Object IDs 选择 IDC_XTXAPP1, Messages 选择 OnUsbkeyChange, 单击 Add Function..., 如下图



点击 OK 为控件添加响应插拔的处理函数。

在响应设备插拔的函数中添加相应的处理。

- 步骤五：编写调用控件的代码。

在此仅给出了产生随机数的调用示例，其他示例详见 Demo。

```
CString r = m_xtxApp.SOF_GenRandom(24);

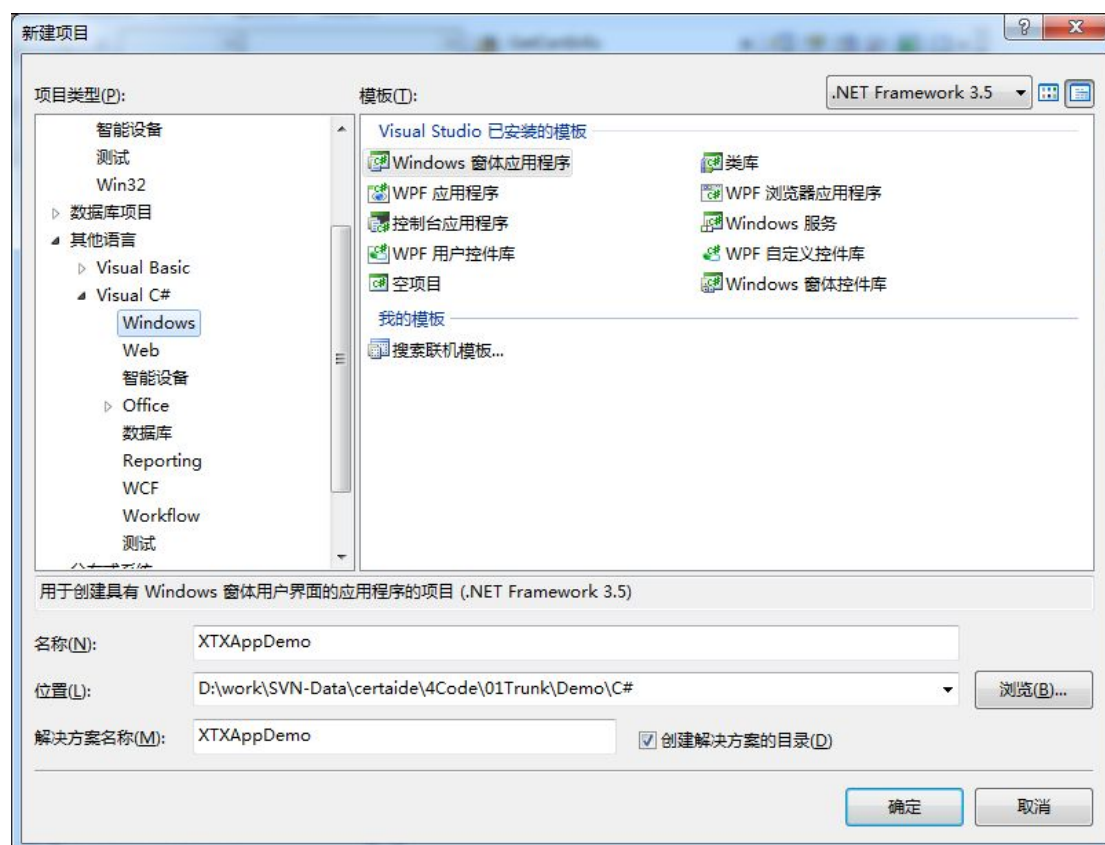
MessageBox(r, _T("产生随机数成功"), MB_OK | MB_ICONINFORMATION);
```

2.5.3 C#调用方法

在此以 Visual Studio 2010 为例，介绍 C#中如何调用 XTXApp 接口。其他 VS2005、2008 等调用方法类似(目前在 VS2005 和 VS2008 中不能响应设备插拔事件)。

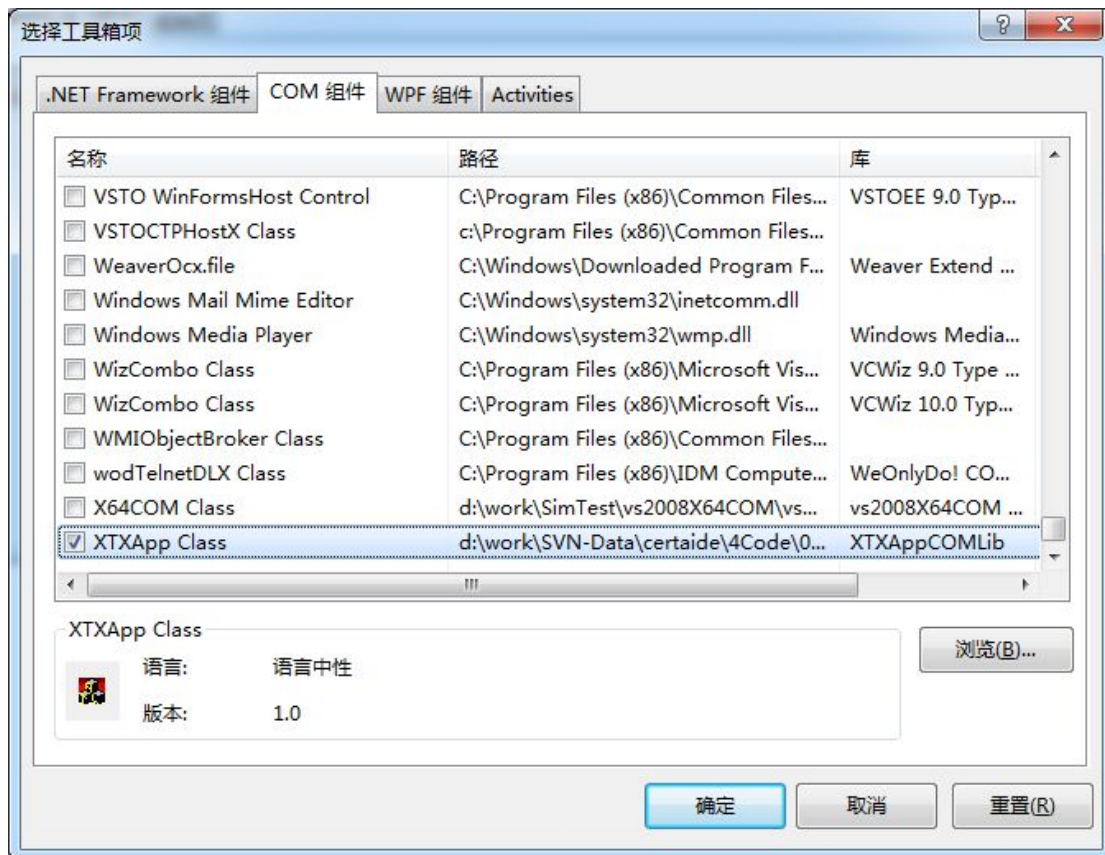
- 步骤一：新建 C#工程

选择[文件]->[新建]->[项目]->[Visual C#]->[Windows]->[Windows 窗体应用程序]

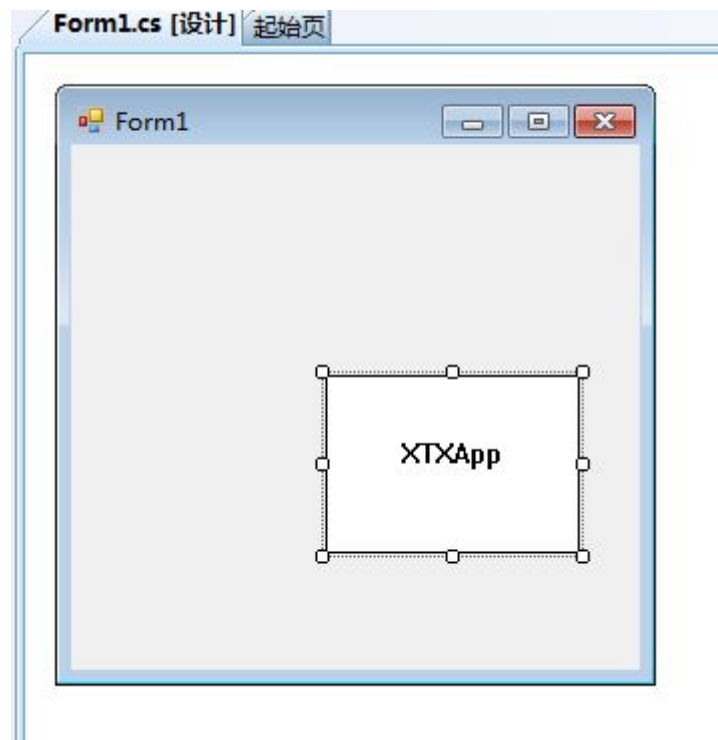


- 步骤二：在工程中加载 XTXApp 控件

在 工具箱->常规 选项卡下点击右键，选择“选择项”，然后选择“COM 组件”标签页，查找 XTXApp Class 项并勾选，最后点击确定按钮。如下图

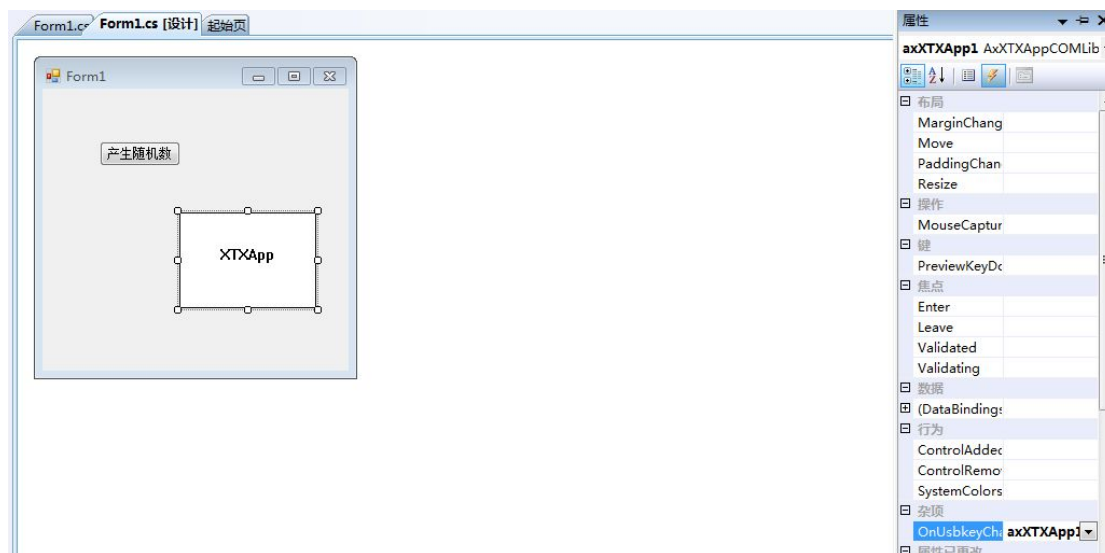


将 XTXApp 控件添加到工程中，在工具箱->常规 出现 XTXApp Class，把组件 XTXApp Class 拖到工程的 Form 中。如下图



- 步骤三：为响应设备插拔事件注册处理函数。

双击 XTXApp 控件或点击右键选择 [属性]->[事件]->[杂项]->[OnUsbkeyChange] 添加处理函数，如下图



在响应设备插拔的函数中添加相应的处理。

- 步骤四：编写调用控件的代码。

在此仅给出了产生随机数的调用示例 (XTXApp 控件对应的名称为系统默认的 axXTXApp1)，其他示例详见 Demo。

```
string r = axXTXApp1.SOF_GenRandom(24);

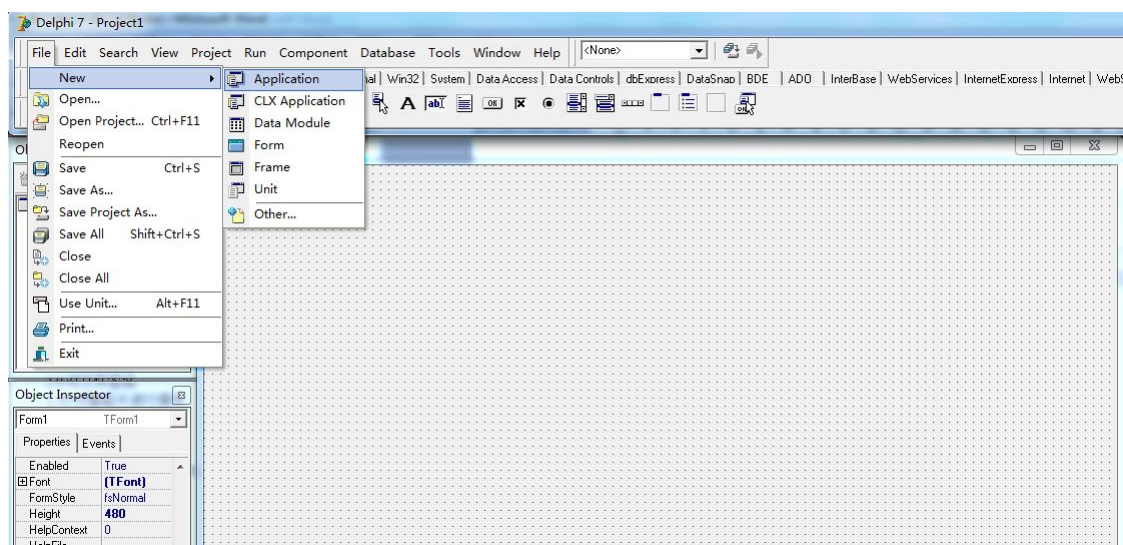
MessageBox.Show(r, "产生随机数成功", MessageBoxButtons.OK,
MessageBoxIcon.Information);
```

2.5.4 Delphi 调用方法

在此以 Delphi7 为例，介绍 Delphi 中如何调用 XTXApp 接口。其他版本的 Delphi 调用方法类似。

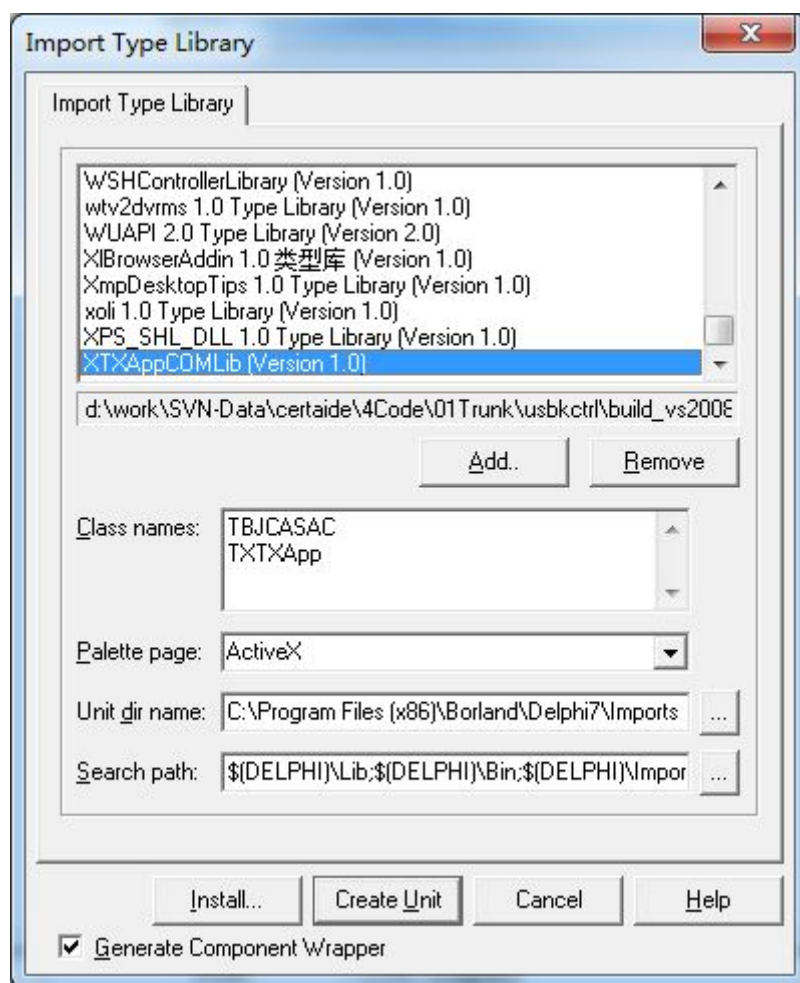
- 步骤一：新建 Delphi 工程

选择[File]->[New]->[Application]

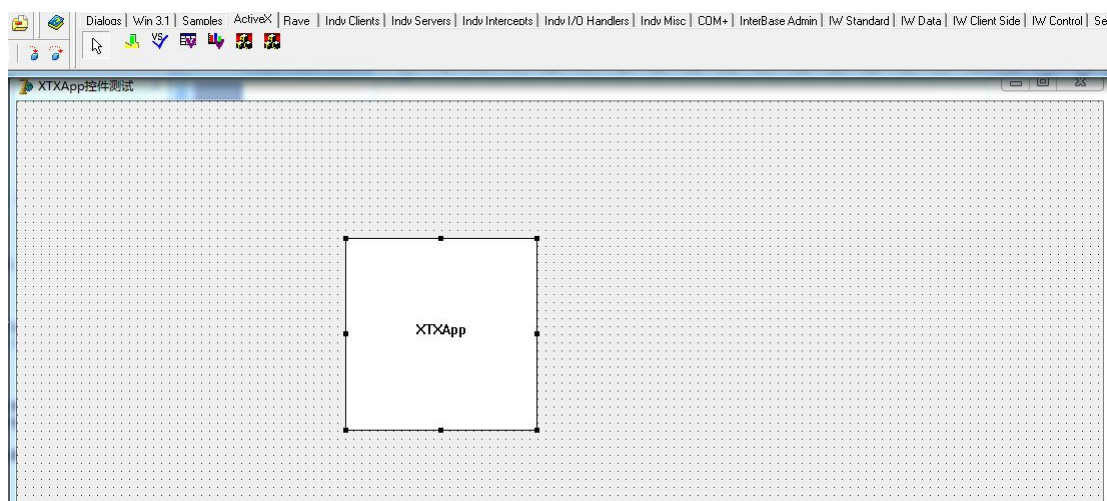


- 步骤二：在工程中加载 XTXApp 控件

选择菜单[project]->[Import Type Library]，查找 Import Type Library 列表中的 XTXAppCOMLib，然后点击[install...]，这样就加载了 XTXAppCOM.dll。如果 XTXAppCOM.dll 已经加载，这步可以跳过去。如下图



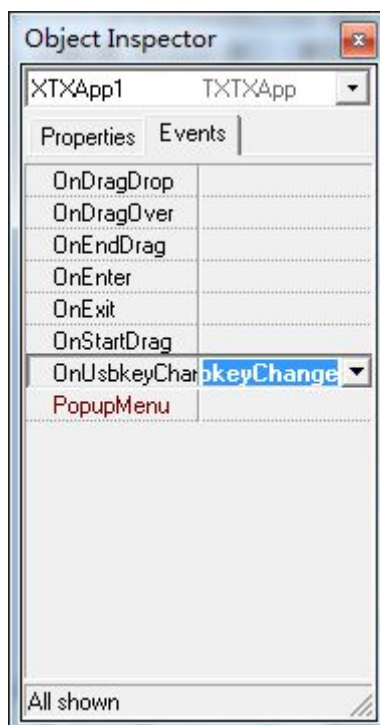
选择工具条中的 ActiveX 选项卡，双击 XTXApp 组件或将 XTXApp 组件拖到 Form 中，如下图



- 步骤三：为响应设备插拔事件注册处理函数。

选中 XTXApp 组件，打开 “Object Inspector” 窗口，选择 [Events] 选项卡，

找到 OnUsbKeyChange 项，双击即可添加设备插拔的响应处理函数，如下图



在响应设备插拔的处理函数中进行相应处理。

- 步骤四：编写调用控件的代码。

在此仅给出了产生随机数的调用示例 (XTXApp 控件对应的名称为系统默认的 XTXApp1)，其他示例详见 Demo。

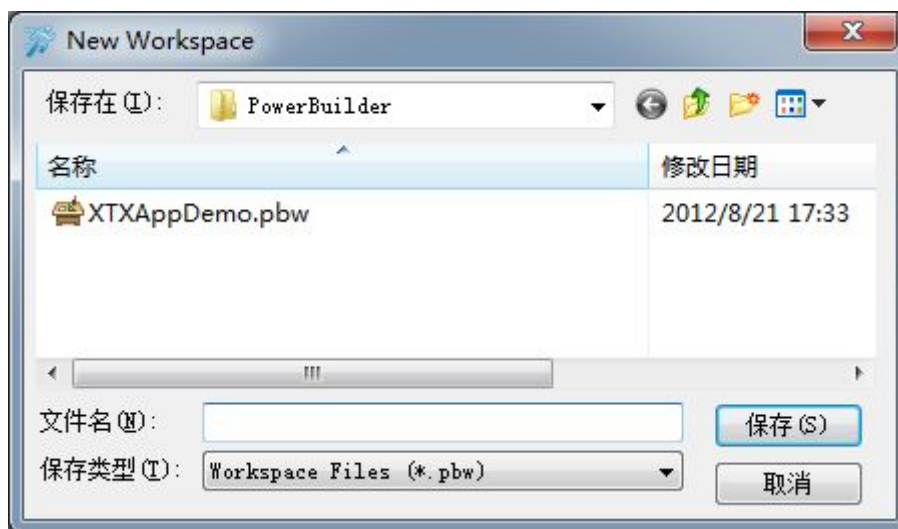
```
var r : string;  
begin  
    r := XTXApp1.SOF_GenRandom(24);  
    showmessage(r);  
end;
```

2.5.5 PowerBuilder 调用方法

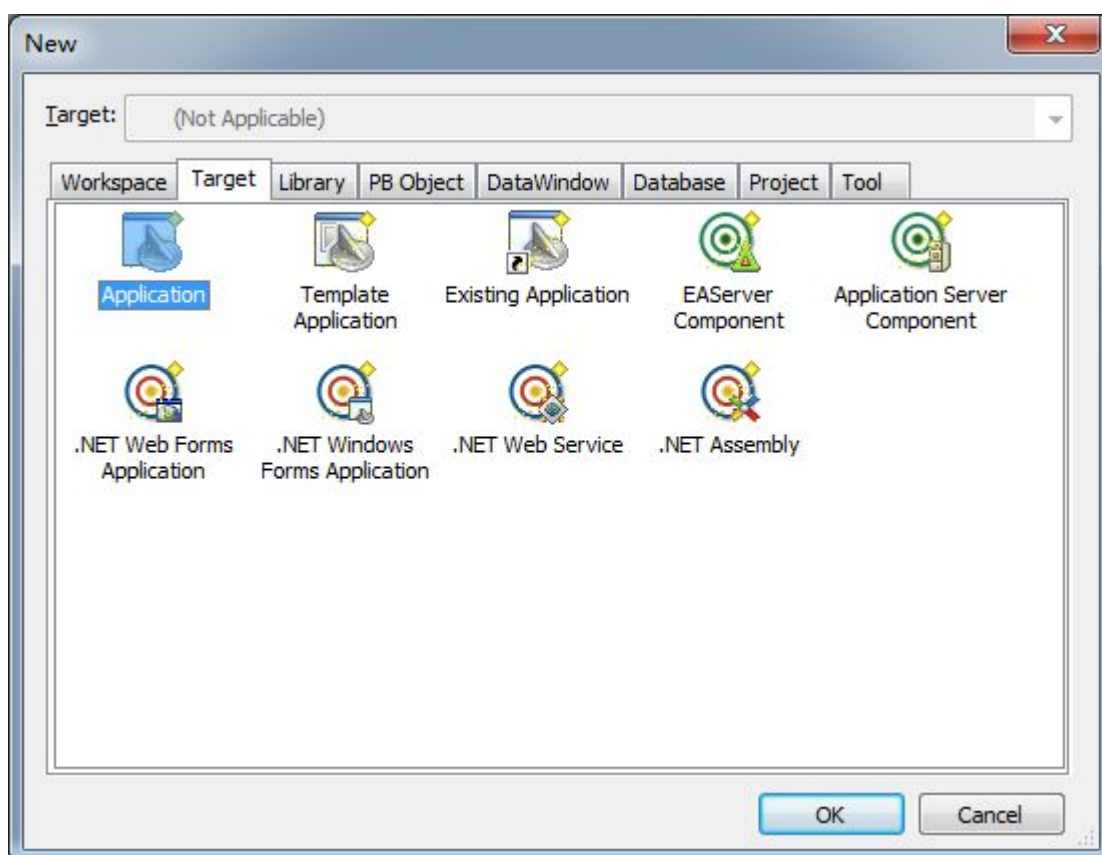
在此以 PowerBuilder12.0 为例，介绍 PowerBuilder 中如何调用 XTXApp 接口。其他版本的 PowerBuilder 调用方法类似。

- 步骤一：新建 Delphi 工程

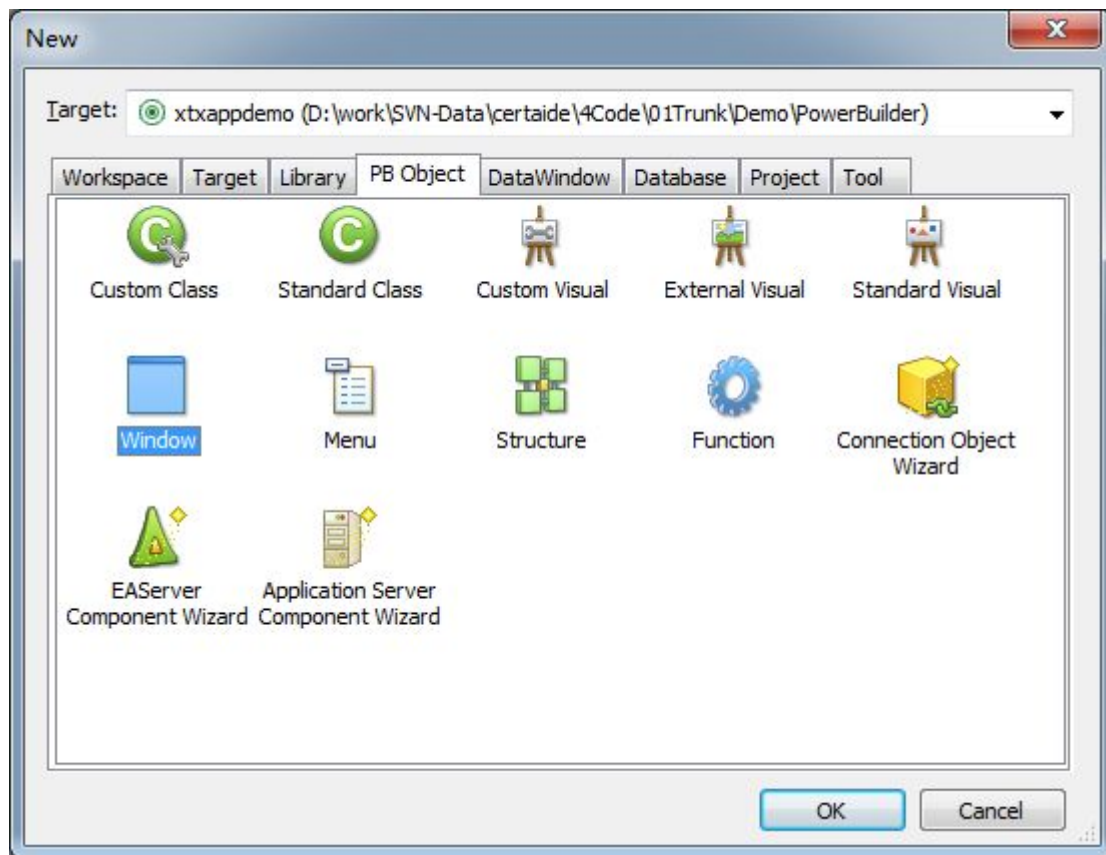
选择[File]->[New]->[WorkSpace], 假设名称为 XTXAppDemo, 如下图



在 XTXAppDemo 工作区, 点击右键[New]->[Target]->[Application], 如下图

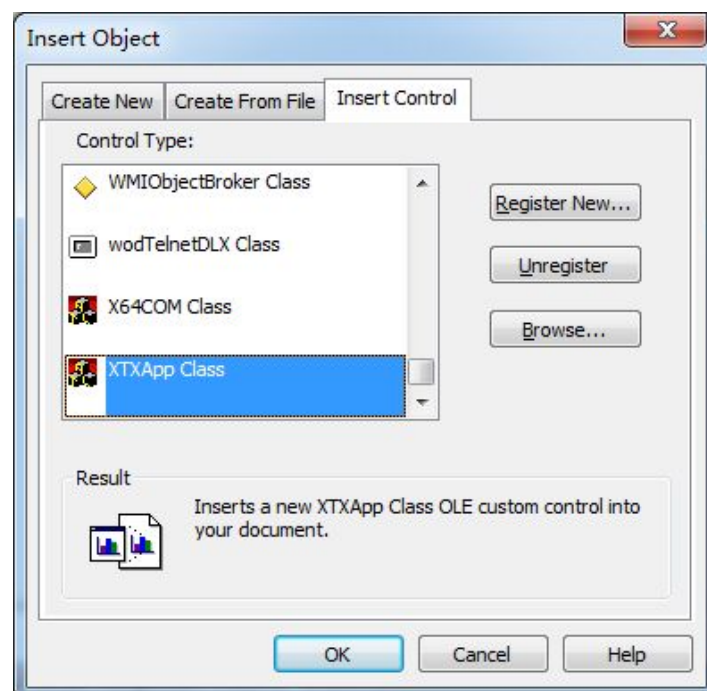


Application Name 假设为 XTXAppDemo, 然后在 XTXApp 应用右键, [New]->[Pb Object]->[Window], 添加一个窗口, 如下图

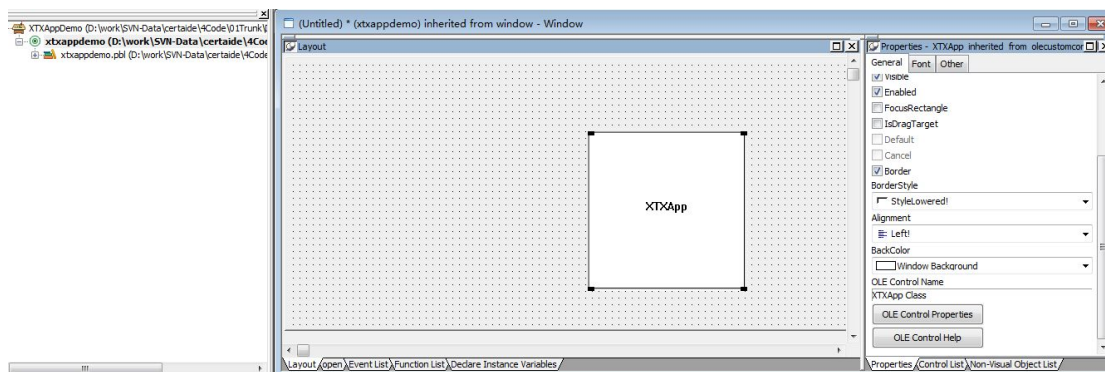


- 步骤二：在工程中加载 XTXApp 控件

选择菜单[Insert]->[Control]->[OLE...], 打开 InsertObject 窗口, 选择 Insert Control 选项卡, 找到 XTXApp Class, 点击 OK 按钮。

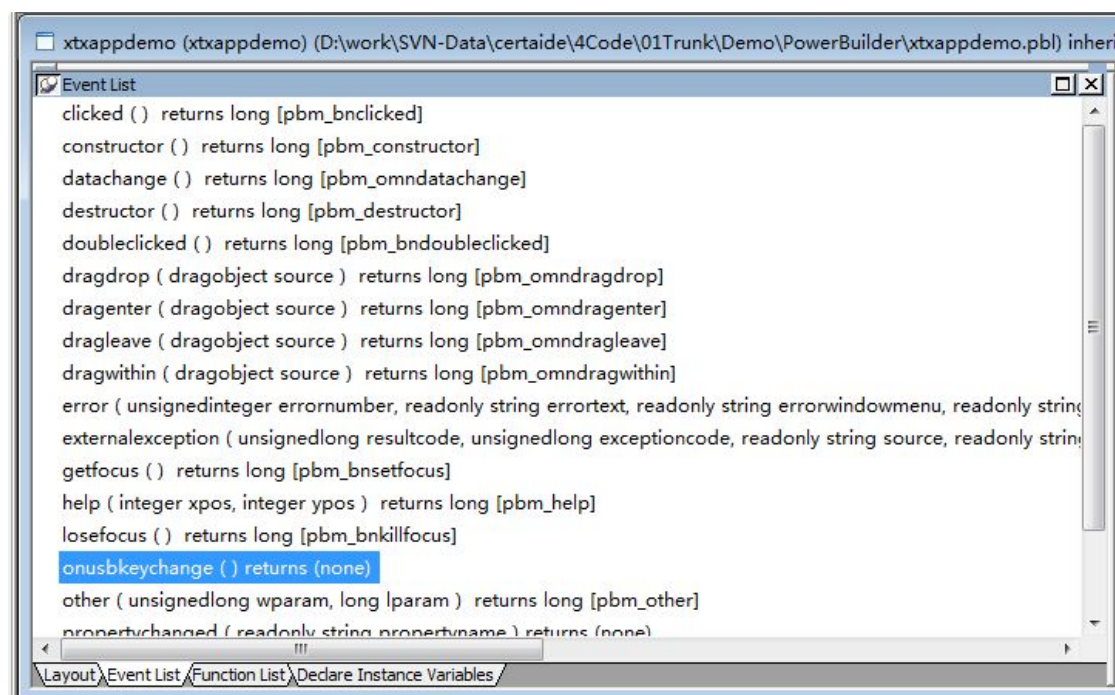


XTXApp 控件添加到窗口中如下图



● 步骤三：为响应设备插拔事件添加处理函数

选中 XTXApp 组件，打开“Object Inspector”窗口，选择[Events]选项卡，找到 onusbkeychange () 项，双击即可添加设备插拔的响应处理函数，如下图



在响应设备插拔的处理函数中进行相应处理。

● 步骤四：编写调用控件的代码。

在此仅给出了产生随机数的调用示例（XTXApp 控件对应的名称为 XTXApp），其他示例详见 Demo。

```
string a  
a = XTXAPP.object.SOF_GenRandom(24)  
MessageBox("产生随机数成功", a)
```

第 3 章 附录

3.1 签名算法定义

宏定义	值	备注
SGD_SM3_RSA	0x00010001	基于SM3算法和RSA算法的签名
SGD_SHA1_RSA	0x00010002	基于SHA1算法和RSA算法的签名
SGD_SHA256_RSA	0x00010004	基于SHA256算法和RSA算法的签名
SGD_SM3_SM2	0x00020101	基于SM2算法和SM3算法的签名

3.2 加密算法定义

宏定义	值	备注
SGD_SM1_ECB	0x00000101	SM1算法ECB加密模式
SGD_SM1_CBC	0x00000102	SM1算法CBC加密模式
SGD_SM1_CFB	0x00000104	SM1算法CFB加密模式
SGD_SM1_OFB	0x00000108	SM1算法OFB加密模式
SGD_SM1_MAC	0x00000110	SM1算法MAC运算
SGD_SSF33_ECB	0x00000201	SSF33算法ECB加密模式
SGD_SSF33_CBC	0x00000202	SSF33算法CBC加密模式
SGD_SSF33_CFB	0x00000204	SSF33算法CFB加密模式

SGD_SSF33_OFB	0x00000208	SSF33算法OFB加密模式
SGD_SSF33_MAC	0x00000210	SSF33算法MAC运算
SGD_SMS4_ECB	0x00000401	SMS4算法ECB加密模式
SGD_SMS4_CBC	0x00000402	SMS4算法CBC加密模式
SGD_SMS4_CFB	0x00000404	SMS4算法CFB加密模式
SGD_SMS4_OFB	0x00000408	SMS4算法OFB加密模式
SGD_SMS4_MAC	0x00000410	SMS4算法MAC运算
SGD_DES_ECB	0x00000801	DES算法ECB加密模式
SGD_DES_CBC	0x00000802	DES算法CBC加密模式
SGD_DES_CFB	0x00000804	DES算法CFB加密模式
SGD_DES_OFB	0x00000808	DES算法OFB加密模式
SGD_3DES_2KEY_E CB	0x00001001	3DES算法16字节密钥ECB加密模式
SGD_3DES_2KEY_C BC	0x00001002	3DES算法16字节密钥CBC加密模式
SGD_3DES_2KEY_C FB	0x00001004	3DES算法16字节密钥CFB加密模式
SGD_3DES_2KEY_O FB	0x00001008	3DES算法16字节密钥OFB加密模式
SGD_3DES_3KEY_E CB	0x00002001	3DES算法24字节密钥ECB加密模式
SGD_3DES_3KEY_C BC	0x00002002	3DES算法24字节密钥CBC加密模式
SGD_3DES_3KEY_C FB	0x00002004	3DES算法24字节密钥CFB加密模式

SGD_3DES_3KEY_OFB	0x00002008	3DES算法24字节密钥OFB加密模式
SGD_AES_128_ECB	0x00004001	AES算法16字节密钥ECB加密模式
SGD_AES_128_CBC	0x00004002	AES算法16字节密钥CBC加密模式
SGD_AES_128_CFB	0x00004004	AES算法16字节密钥CFB加密模式
SGD_AES_128_OFB	0x00004008	AES算法16字节密钥OFB加密模式
SGD_AES_192_ECB	0x00008001	AES算法24字节密钥ECB加密模式
SGD_AES_192_CBC	0x00008002	AES算法24字节密钥CBC加密模式
SGD_AES_192_CFB	0x00008004	AES算法24字节密钥CFB加密模式
SGD_AES_192_OFB	0x00008008	AES算法24字节密钥OFB加密模式
SGD_AES_256_ECB	0x00010001	AES算法32字节密钥ECB加密模式
SGD_AES_256_CBC	0x00010002	AES算法32字节密钥CBC加密模式
SGD_AES_256_CFB	0x00010004	AES算法32字节密钥CFB加密模式
SGD_AES_256_OFB	0x00010008	AES算法32字节密钥OFB加密模式

3.3 摘要算法

宏定义	值	备注
SGD_SM3	0x00000001	SM3摘要算法
SGD_SHA1	0x00000002	SHA1摘要算法
SGD_SHA256	0x00000004	SHA256摘要算法

3.4 错误码定义

错误码	描述
1	设备序列错误（索引值错误）
2	设备序列号长度错误
3	打开设备错误
4	打开应用错误
5	没有打开设备
6	Pin码长度错误
7	校验口令失败
8	获取Pin码重试次数错误
9	修改管理员口令失败
10	修改用户口令失败
11	解锁用户口令失败
12	容器名称长度错误
13	创建容器失败
14	产生密钥对失败
15	打开容器失败
16	导出公钥失败
17	尚未登录
18	导入证书失败
19	容器类型错误
20	ENVSN为空
21	写入ENVSN失败
22	读取ENVSN失败
23	获取容器数量失败
24	删除容器失败
25	判断容器是否存在失败

26	获取设备序列号失败
27	导入证书时，证书中的公钥和设备容器中的公钥不匹配
28	Base64 编码失败
29	Base64 解码失败
30	解析字符串失败
31	产生随机数失败
32	设置明文对称密钥失败
33	加密或解密初始化失败
34	对称加密失败
35	认证设备密钥失败
36	删除应用失败
37	创建应用失败
38	创建文件失败
39	设置设备标签失败
40	修改设备认证密钥失败
41	摘要运算失败
42	签名失败
43	验签失败
44	导出证书失败
45	导出 PKCS10 证书请求失败
46	参数错误
47	写入文件失败
48	读取文件失败
49	解析证书失败
50	不存在设备
51	容器已存在
52	获取设备信息失败
53	格式化设备失败
54	对称解密失败

55	PAD错误
56	私钥解密失败
57	公钥加密数据失败
58	OTP设备获取应答码失败
59	计算HMAC失败
60	验证证书失败
61	解析PKCS12文件失败
62	XML数据签名失败
63	XML数据验签失败
64	获取XML签名数据的信息失败
65	密钥分割失败
66	恢复分割的密钥失败
67	关闭OTP设备失败
68	产生时间戳请求失败
69	比较时间戳的Nonce失败
70	产生带时间戳的签名失败
71	解析带时间戳的签名失败
72	验证带时间戳的签名失败
73	OTP设备获取时间同步码失败
101	对输入的参数不支持
102	设备类型错误
103	不存在证书
104	获取签名算法错误
105	创建软设备失败
106	删除软设备失败
107	备份软设备失败
108	恢复软设备失败
109	启用或禁用软设备失败

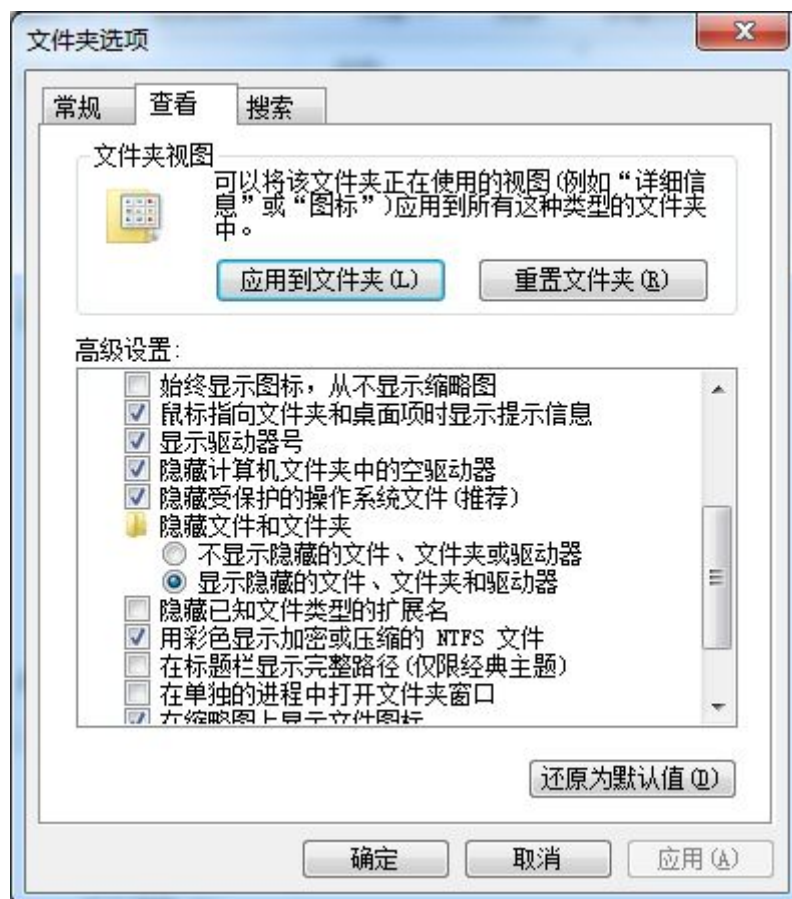
3.5 关于日志

在 Windows7或 Windows8系统上日志记录在

系统盘:\Users\当前用户\AppData\LocalLow\BJCA\log 目录下

默认 log 目录是不存在的，如果需要记录日志，需要手动创建 log 目录

AppData 目录是隐藏的 需要打开 文件夹选项 显示隐藏的文件 文件夹和驱动器



在 Windows2000或 WindowsXP 等系统上日志记录在

系统盘 :\Documents and Settings\当前用户\Local Settings\Application

Data\BJCA\log 目录下，默认 log 目录是不存在的，如果需要记录日志，需要手

动创建 log 目录。其中 Local Settings 目录是隐藏目录，需要设置。需要打开 文

件夹选项 显示隐藏的文件 文件夹和驱动器。