

TFT 系列脱机通讯开发包 开发手册

版本: 6.14

日期: 2009 年12月

2009 ZKSoftware Inc.。版权所有，翻印必究。

- 本说明文件中的信息如有更改，恕不另行通知。
- ZKSoftware Inc. 为中控科技的司标，Biokey 为中控科技的核心技术徽标，ZKSoftware ,Biokey 已经在中国及美国注册。
- 本说明文件中述及其他商标和产品名称是指拥有相应商标和产品名称的公司或者制造的产品。

ZKSoftware Inc. 对其他公司的商标和产品名称不拥有任何所有权。

- 开发中使用问题，非开发包问题，请自行解决。
- 使用该开发包，由于人为或程序本身造成数据丢失，本公司概不负责。
- 该文档基于 6.0.3.5 版 SDK

目录

1. SDK 介绍.....	1
2. 快速开始.....	2
2.1 SDK 安装.....	2
2.2 常用词介绍.....	2
2.3 常用工作流程介绍.....	3
2.3.1 考勤记录下载流程:.....	3
2.3.2 操作记录下载流程:.....	4
2.3.3 门禁设置流程:.....	4
2.3.4 下载用户信息,指纹模板等流程.....	5
2.3.5 实时事件接收流程.....	5
2.3.6 联机登记流程 (上传用户信息,指纹模板等流程).....	6
2.3.7 短消息上传流程.....	7
3. 属性相关介绍.....	7
3.1 AccGroup.....	7
3.2 AccTimeZones.....	7
3.3 BASE64.....	7
3.4 CardNumber.....	8
3.5 CommPort.....	8
3.6 ConvertBIG5.....	8
3.7 PINWidth.....	8
3.8 GetStrCardNumber.....	8
3.9 SetStrCardNumber.....	9
4. 实时事件相关.....	9
4.1 获取实时事件方法.....	9
4.1.1 RegEvent.....	9
4.1.2 ReadRTLog.....	10
4.1.3 GetRTLog.....	10
4.2 实时事件介绍.....	10
4.2.1 OnConnected.....	10
4.2.2 OnDisConnected.....	10
4.2.3 OnAlarm.....	11
4.2.4 OnDoor.....	11
4.2.5 OnAttTransactionEx.....	11
4.2.6 OnEnrollFingerEx.....	12
4.2.7 OnFinger.....	12
4.2.8 OnFingerFeature.....	12
4.2.9 OnHIDNum.....	12
4.2.10 OnNewUser.....	12
4.2.11 OnVerify.....	13
4.2.12 OnWriteCard.....	13
4.2.13 OnEmptyCard.....	13
4.2.14 OnEMData.....	13
5 常用函数介绍.....	13
5.1 连接机器相关函数.....	13
5.1.1 Connect_Net.....	13
5.1.2 Connect_Com.....	14
5.1.3 Disconnect.....	14
5.2 数据管理相关函数.....	15

5.2.1 考勤记录数据.....	15
5.2.1.1 ReadGeneralLogData	15
5.2.1.2 ReadAllGLogData.....	15
5.2.1.3 SSR_GetGeneralLogData	15
5.2.1.4 ClearGLog	16
5.2.2 操作记录相关.....	17
5.2.2.1 ReadSuperLogData	17
5.2.2.2 ReadAllSLogData	17
5.2.2.3 GetSuperLogData.....	17
5.2.2.4 GetAllSLogData.....	19
5.2.2.5 ClearSLog	19
5.2.2.6 GetSuperLogData2.....	19
5.2.3 用户信息相关.....	20
5.2.3.1 ReadAllUserID.....	20
5.2.3.2 SSR_EnableUser	20
5.2.3.3 SetUserInfoEx	21
5.2.3.4 GetUserInfoEx	21
5.2.3.5 DeleteUserInfoEx.....	22
5.2.3.6 SSR_GetAllUserInfo.....	22
5.2.3.7 SSR_GetUserInfo.....	23
5.2.3.8 SSR_SetUserInfo	23
5.2.4 登记数据（同时包括用户信息和指纹）	24
5.2.4.1 SSR_DeleteEnrollData.....	24
5.2.4.2 SSR_DeleteEnrollDataExt	24
5.2.5 指纹模板相关.....	25
5.2.5.1 ReadAllTemplate.....	25
5.2.5.2 SSR_GetUserTmp	25
5.2.5.3 SSR_GetUserTmpStr	26
5.2.5.4 SSR_SetUserTmp.....	26
5.2.5.5 SSR_SetUserTmpStr	26
5.2.5.6 SSR_DelUserTmp	27
5.2.5.7 SSR_SetUserTmpExt	27
5.2.5.8 SSR_DelUserTmpExt	28
5.2.5.9 SetUserTmpEx	28
5.2.5.10 SetUserTmpExStr.....	29
5.2.5.11 GetUserTmpEx.....	30
5.2.5.12 GetUserTmpExStr	30
5.2.6 其他.....	31
5.2.6.1 短消息相关.....	31
5.2.6.1.1 SetSMS.....	31
5.2.6.1.2 SSR_SetUserSMS	31
5.2.6.1.3 GetSMS	32
5.2.6.1.4 DeleteSMS	32
5.2.6.1.5 SSR_DeleteUserSMS	33
5.2.6.1.6 ClearUserSMS.....	33
5.2.6.1.7 ClearSMS	33
5.2.6.2 WorkCode 相关.....	34
5.2.6.2.1 SSR_GetWorkCode.....	34
5.2.6.2.2 SSR_SetWorkCode.....	34
5.2.6.2.3 SSR_DeleteWorkCode	34

5.2.6.2.4SSR_ClearWorkCode	35
5.2.6.3 节假日相关.....	35
5.2.6.3.1 SSR_GetHoliday	35
5.2.6.3.2 SSR_SetHoliday.....	35
5.2.6.4 夏令时相关.....	36
5.2.6.4.1 SetDaylight.....	36
5.2.6.4.2 GetDaylight	36
5.2.6.5 快捷键定义.....	37
5.2.6.5.1 SSR_SetShortcut	37
5.2.6.5.2 SSR_GetShortcut.....	38
5.2.6.6 指纹模板转换相关.....	38
5.2.6.6.1 GetFPTempLength	38
5.2.6.6.2 GetFPTempLengthStr.....	39
5.2.6.6.3 FPTempConvert.....	39
5.2.6.6.4 FPTempConvertStr	40
5.2.6.6.5 FPTempConvertNew	40
5.2.6.6.6 FPTempConvertNewStr.....	40
5.2.7 系统数据管理相关.....	41
5.2.7.1 ClearKeeperData	41
5.2.7.2 ClearData.....	41
5.2.7.3 GetDataFile	42
5.2.7.4 SendFile.....	42
5.2.7.5 RefreshData.....	43
5.3 门禁相关（时间段，组，开锁组合）	43
5.3.1 GetUserGroup.....	43
5.3.2 SetUserGroup.....	43
5.3.3 GetTZInfo.....	44
5.3.4 SetTZInfo.....	44
5.3.5 SSR_SetUnLockGroup.....	45
5.3.6 SSR_GetUnLockGroup.....	45
5.3.7 SSR_SetGroupTZ.....	46
5.3.8 SSR_GetGroupTZ.....	46
5.3.9 GetUserTZs.....	47
5.3.10 SetUserTZs.....	47
5.3.11 GetUserTZStr.....	48
5.3.12 SetUserTZStr	48
5.3.13 ACUnlock.....	49
5.3.14 GetACFun.....	49
5.3.15 GetDoorState.....	49
5.3.16 UseGroupTimeZone.....	50
5.4 机器管理.....	50
5.4.1 IsTFTMachine.....	50
5.4.2 GetDeviceStatus.....	50
5.4.3 GetDeviceInfo.....	51
5.4.4 SetDeviceInfo.....	53
5.4.5 SetDeviceTime.....	54
5.4.6 SetDeviceTime2.....	54
5.4.7 GetDeviceTime.....	54
5.4.8 GetSerialNumber	55
5.4.9 GetProductCode.....	55

5.4.10	GetFirmwareVersion.....	55
5.4.11	GetSDKVersion.....	56
5.4.12	GetDeviceIP.....	56
5.4.13	SetDeviceIP.....	56
5.4.14	GetDeviceMAC.....	57
5.4.15	SetDeviceMAC.....	57
5.4.16	GetWiegandFmt.....	58
5.4.17	SetWiegandFmt.....	58
5.4.18	GetCardFun.....	58
5.4.19	SetDeviceCommPwd.....	59
5.4.20	SetCommPassword.....	59
5.4.21	QueryState.....	59
5.4.22	GetVendor.....	60
5.4.23	GetDeviceStrInfo.....	60
5.4.24	GetPlatform.....	61
5.4.25	ReadAOptions.....	61
5.4.26	GetSysOption.....	61
5.4.27	SetSysOption.....	62
5.5	其他.....	62
5.5.1	机器控制.....	62
5.5.1.1	ClearAdministrators.....	62
5.5.1.2	EnableDevice.....	63
5.5.1.3	EnableClock.....	63
5.5.1.4	DisableDeviceWithTimeOut.....	63
5.5.1.5	PowerOffDevice.....	64
5.5.1.6	RestartDevice.....	64
5.5.1.7	SleepDevice.....	64
5.5.2	连机登记相关操作.....	64
5.5.2.1	StartEnrollEx.....	64
5.5.2.2	StartVerify.....	65
5.5.2.3	StartIdentify.....	65
5.5.2.4	CancelOperation.....	65
5.5.3	卡操作相关.....	66
5.5.3.1	WriteCard.....	66
5.5.3.2	EmptyCard.....	66
5.5.4	其他.....	67
5.5.4.1	GetLastError.....	67
5.5.4.2	GetHIDEEventCardNumAsStr.....	67
5.5.4.3	CaptureImage.....	67
5.5.4.4	UpdateFirmware.....	68
5.5.4.5	BeginBatchUpdate.....	68
5.5.4.6	BatchUpdate.....	69
5.5.4.7	CancelBatchUpdate.....	69
5.5.4.8	PlayVoice.....	69
5.5.4.9	PlayVoiceByIndex.....	70
6.	常见问题解答.....	70
6.1	如何下载考勤记录.....	70
6.2	如何联机创建用户.....	70
6.3	U 盘数据导入及下载.....	70
6.4	BIOKEY 采集指纹模板写入脱机.....	72

6.5 取得所有用户所有信息.....	72
6.6 机器连接.....	72
6.7 在 SetUserInfo 后密码不能用问题.....	72
6.8 联机模板转换为脱机模板.....	73
6.9 Demo 程序不能接机器问题.....	73
6.10 脱机指纹机连接后处于工作状态问题.....	73
6.11 上传非英文姓名或短消息到机器内出现乱码，或不能正常显示	73
6.12 卡管理问题.....	73
6.13 连接穿越防火墙或路由器问题.....	74
6.14 ZKFinger10.0 算法和 ZKFinger9.0 算法的区别以及相应模板的比较	74
6.15 大容量指纹上传.....	74
6.16 高速上传和普通上传的区别.....	74
6.17 如何判断机器使用的是 10.0 算法还是 9.0 算法	75
6.18 如何支持 ZKFinger10.0 算法模板的上传、下载、删除	75
6.19 ZKFinger9.0 算法模板的上传、下载、删除	76



1. SDK 介绍

脱机通讯开发包是一个与脱机指纹机、门禁机、卡机等数据进行数据通讯的接口。可以方便地对机器进行用户信息、指纹的管理，下载考勤记录、管理记录、用户信息、指纹模板，对机器进行各种设置及门禁配置管理的开发包。主要功能有：

- 1、下载机器考勤记录。
- 2、下载、上传用户信息、卡信息、指纹信息。
- 3、设置门禁机的门禁规则。
- 4、进行各种配置，如设置机器时间、匹配阈值等。
- 5、可实时激发机器的各种事件，如按指纹、验证通过等。
- 6、可直接联机登记用户
- 7、可设置用户“短消息”、“workcode”（只能在标示有此功能的机种使用）。
- 8、可进行个性化的提示音，功能键等设置

2. 快速开始

2.1 SDK 安装

- 1、 Demo
- 3、 setup.exe

如图,SDK 包包含上述文件,运行 setup.exe,即可安装 SDK 到您的电脑

说明

以下都采用 C 语法说明函数协议。函数以功能分类。SDK 默认为脱机通信开发包。

2.2 常用词介绍

1. 实时事件

在应用程序与机器通信成功后,部分在机器上的操作,如连接机器,验证,登记新用户等,会实时触发相应的事件,并将数据传输到 PC 端(上位机),这些触发的事件被称为实时事件。用户可以通过实时事件来实时监控机器目前所处的状态以及用户进行的实时操作。

2. FP

Fingerprint 的缩写,即指纹

3. 指纹算法

指生成指纹模板和验证指纹模板的算法,当前我司最新的指纹算法为 ZKFinger 10.0,该算法为高速算法,各方面性能大大提高,具体可参考 FAQ 说明

4. 高速缓冲区

指 SDK 在使用过程中在 PC 端申请的一部分内存,在上传下载数据过程中,数据首先被保存在缓冲区内,然后才会被相应的处理

5. 时间段、组、开锁组合

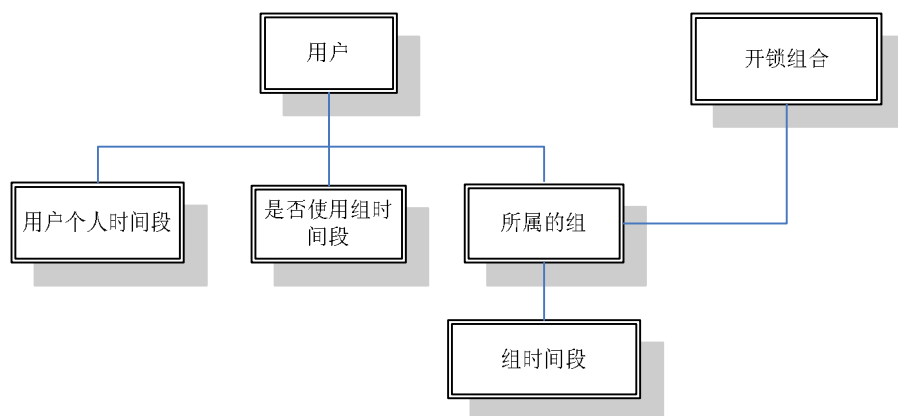
这三者为门禁功能中最重要的概念

时间段就是一个时间范围,一个时间段包含一周的时间信息,该周内每天有一个时间范围,如下面时间段,表示一周 7 天,每天的时间范围都是 00:00 至 22:11。可以表示为

000022110000221100002211000022110000221100002211,一般机器内可以定义 50 个时间段组为一个类别,当多数用户都具有相同的门禁权限时,可以将这些用户都分在一个组内,并设置这些用户都使用组时间段,然后给该组设置相应的时间段即可

开锁组合指需要哪些组一起才能进行开锁操作,若开锁组合内只有一个组,则表示该组内所有用户都可独立验证开锁,如开锁组合内包含 2 个或 2 个以上组时,则表示若想要开锁的话,必须同时验证通过才能开锁,如开锁组合包含组 A 和组 B,则只有组 A 中的一个成员验证通过 + 组 B 中一个成员验证通过,才能开锁。

另:三者关系如下



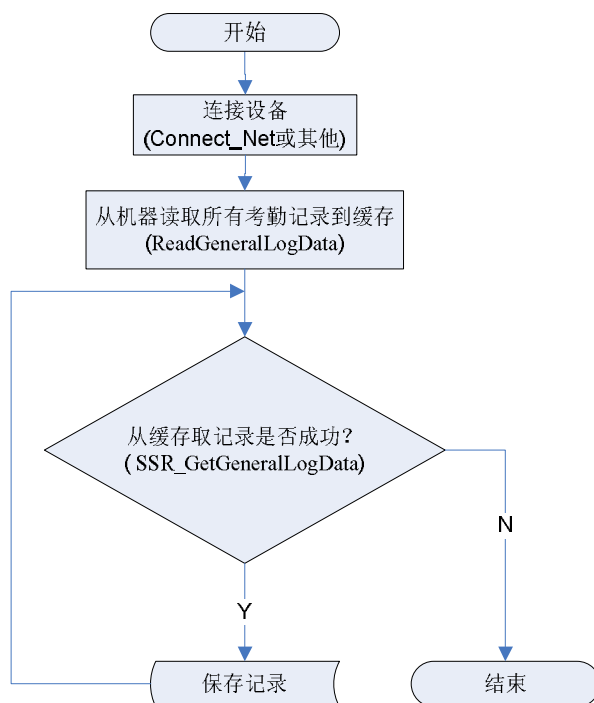
6. 操作记录

也称管理记录，为用户或管理员在操作机器时产生的操作记录，如开机，关机，登记用户等操作。

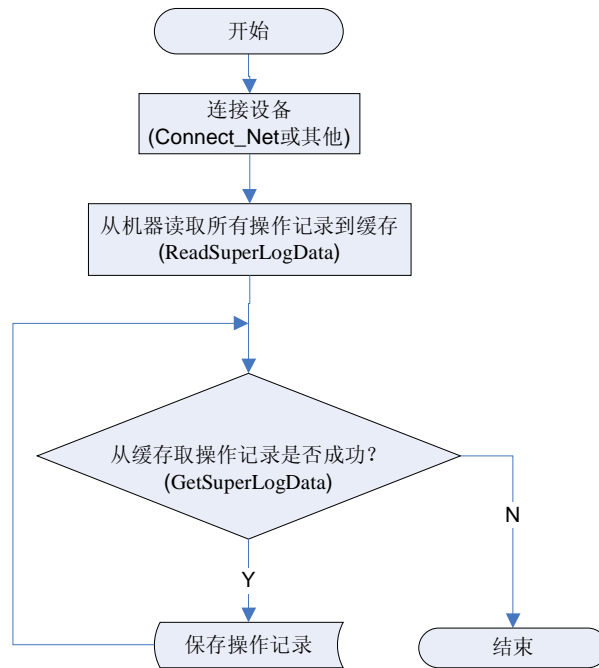
2.3 常用工作流程介绍

详细使用方法请参考对应的 demo 程序。

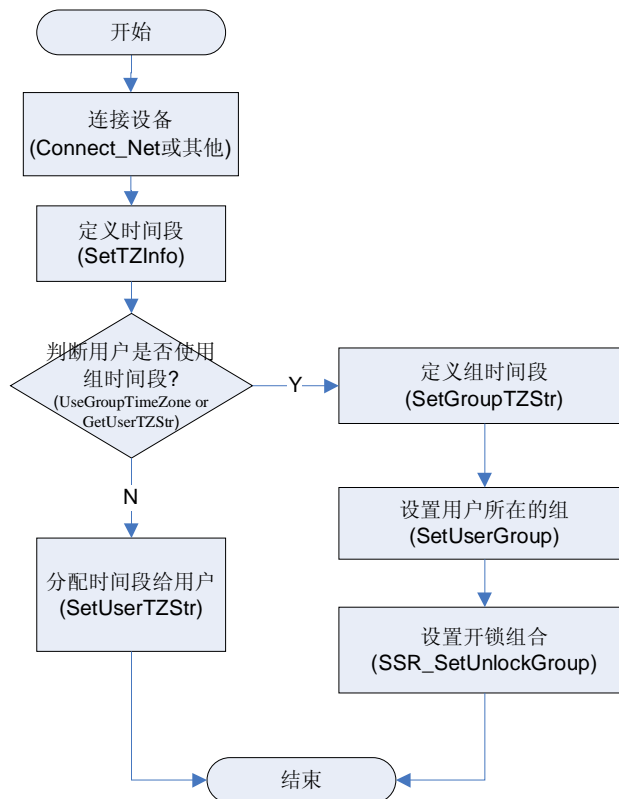
2.3.1 考勤记录下载流程：



2.3.2 操作记录下载流程:

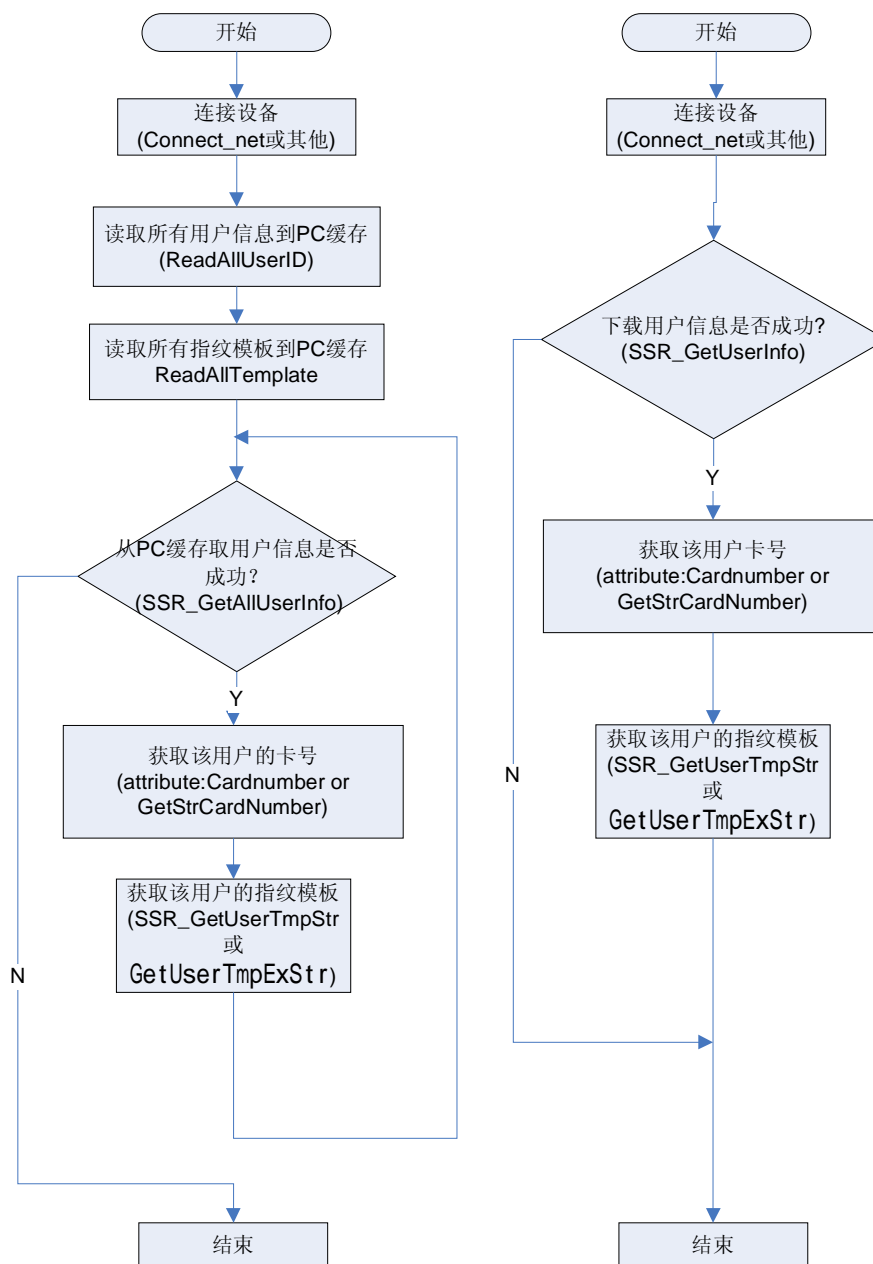


2.3.3 门禁设置流程:



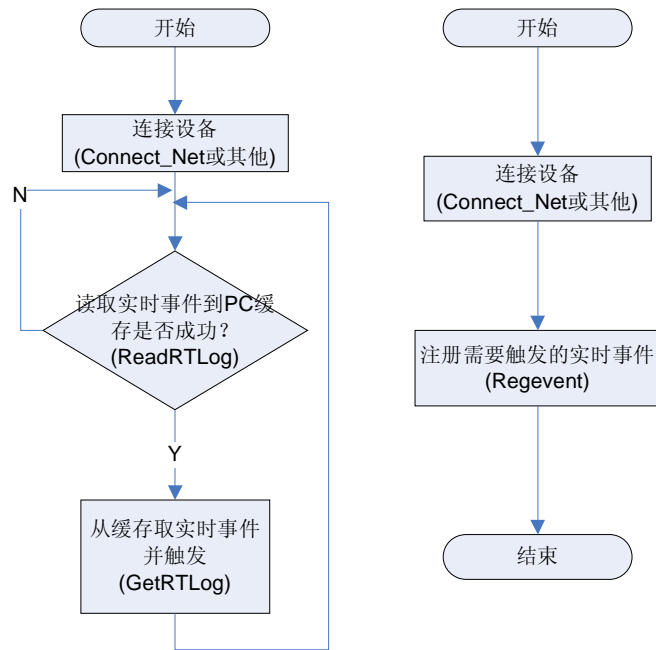
2.3.4 下载用户信息,指纹模板等流程

左边为下载所有用户，右边为下载指定用户



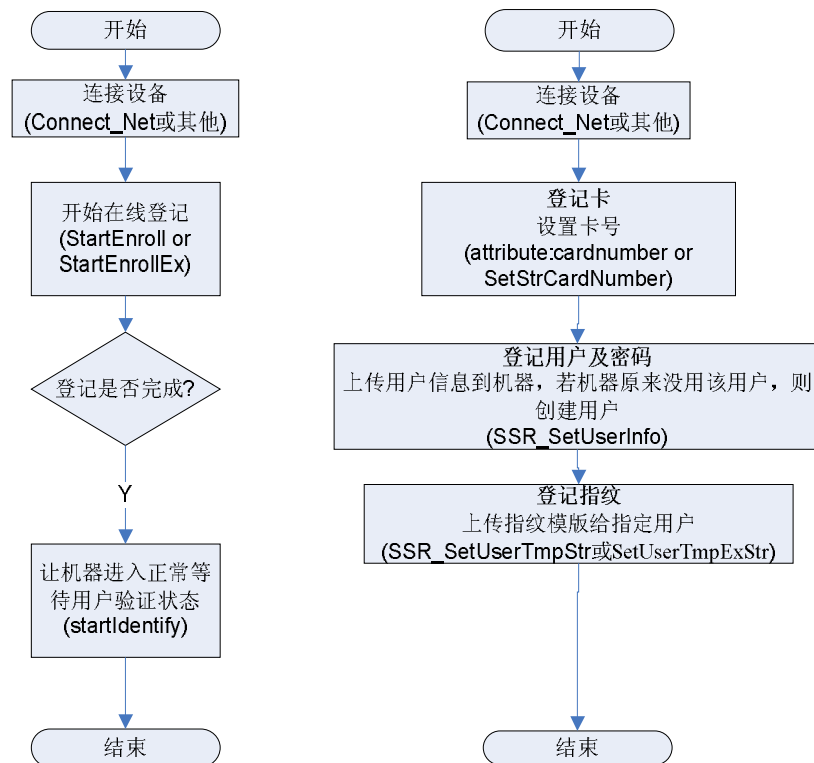
2.3.5 实时事件接收流程

实时事件接收有以下两种方式，建议用户使用第二种方式

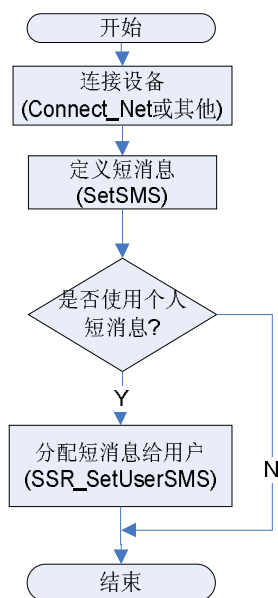


2.3.6 联机登记流程（上传用户信息, 指纹模板等流程）

联机登记用户分为两种方式，如下：左边为连接机器时机器进入登记界面进行登记，右边为直接在机器内创建用户，并给该用户上传相应的卡号、密码、指纹信息，即分别实现登记卡用户，密码用户，指纹用户等



2.3.7 短消息上传流程



3. 属性相关介绍

3.1 AccGroup

功能: 设置或获取用户所属的组。

在上传用户前如果设置了该属性，则在 SetUserInfo 等函数上传用户时，设置该用户所属的组，否则默认为 1 组。

类型: LONG ，可读写。

3.2 AccTimeZones

功能: 设置用户使用时间段。

在上传用户前如果设置了该属性，则在 SetUserInfo 等函数上传用户时，设置该用户所属的时间段。

类型: LONG * ,可以认为该属性为下标为 3 的一维数组 long 类型数组。可读写。

3.3 BASE64

功能: 当设置该属性为真值时，开发包在输出字符串模板时输出为 Base64 编码，否则是十六进制编码。

类型: LONG ，可读写。

3.4 CardNumber

功能：设置或读取用户的卡号，如果不能使用该属性，请使用 GetStrCardnumber 和 SetStrCardnumber 进行卡号的相关操作

类型：LONG,可读写。

3.5 CommPort

功能：设置串口或 485 连接时的端口号。

类型：LONG,可读写。

3.6 ConvertBIG5

功能：当设置该属性为真值时，开发包自动把简体字符转换为繁体字符，用于繁体版脱机开发，但是在多国语言系列产品中，该函数无效，请不要设置该属性

类型：LONG,可读写。

注意：在多国语言系列版本以后的版本中，请不要设置该属性，该属性无效。ZEM100 5.22， ZEM200 5.30 后的版本不需要修改该属性。

3.7 PINWidth

功能：标示用户编号（阿拉伯数字）最大长度。

类型：LONG,只读。

3.8 GetStrCardNumber

【函数定义】

VARIANT_BOOL GetStrCardNumber([out] BSTR* ACardNumber)

【功能】

获取 SDK 属性 cardnumber 的值，一般在获取到用户信息后即可利用该函数获得相应用户的卡号信息

【参数】

ACardNumber

卡号

【返回值】

成功返回 true,否则返回 false

【相关函数】

GetUserInfo

3.9 SetStrCardNumber

【函数定义】

VARIANT_BOOL SetStrCardNumber([in] BSTR ACardNumber)

【功能】

设置 SDK 属性 cardnumber 的值，一般在设置用户信息前用该函数设置相应用户的卡号信息

【参数】

ACardNumber

卡号

【返回值】

成功返回 true, 否则返回 false

【相关函数】

SetUserInfo

4. 实时事件相关

4.1 获取实时事件方法

4.1.1 RegEvent

【函数定义】

VARIANT_BOOL RegEvent([in] LONG dwMachineNumber, [in] LONG EventMask)

【功能】

注册需要触发的实时事件

【参数】

dwMachineNumber:

机器号

EventMask:

事件代号。具体含义如下

1	OnAttTransaction, OnAttTransactionEx
2 (1<<1)	OnFinger
4 (1<<2)	OnNewUser
8 (1<<3)	OnEnrollFinger, OnEnrollFingerEx
16 (1<<4)	OnKeyPress
256 (1<<7)	OnVerify
512 (1<<8)	OnFingerFeature
1024 (1<<9)	OnDoor, OnAlarm
2048 (1<<10)	OnHIDNum
4096 (1<<11)	OnWriteCard
8192 (1<<12)	OnEmptyCard
16384 (1<<13)	OnDeleteTemplate

使用时，如需注册多个实时事件，只需将相关事件代号的二进制进行或运算即可，如注册全部实时事件，EventMask 参数值为 65535

【返回值】

成功返回 True，否则返回 False

【相关函数】

ReadRTLog，GetRTLog

4.1.2 ReadRTLog

【函数定义】

VARIANT_BOOL ReadRTLog([in] LONG dwMachineNumber)

【功能】

读取实时事件到 PC 的缓冲区，该函数和 GetRTLog 搭配使用可以实现在连接机器成功后主动向机器获取实时事件的功能。

【参数】

dwMachineNumber:

机器号

【返回值】

成功返回真，否则返回假

【相关函数】

GetRTLog

4.1.3 GetRTLog

【函数定义】

VARIANT_BOOL GetRTLog(LONG dwMachineNumber)

【功能】

从 PC 的缓冲区取出事件，并使该事件触发，该函数和 ReadRTLog 搭配使用可以实现在连接机器成功后主动向机器获取实时事件的功能。

【参数】

dwMachineNumber:

机器号

【返回值】

成功返回真，否则返回假

【相关函数】

ReadRTLog

4.2 实时事件介绍

4.2.1 OnConnected

当成功连接机器时触发该事件，无返回值

4.2.2 OnDisconnected

当断开机器时触发该事件，无返回值

FP_OR_PW_OR_RF	0
FP	1
PIN	2
PW	3
RF	4
FP_OR_PW	5
FP_OR_RF	6
PW_OR_RF	7
PIN_AND_FP	8
FP_AND_PW	9
FP AND RF	10

PW_AND_RF	11
FP_AND_PW_AND_RF	12
PIN_AND_FP_AND_PW	13
FP_AND_RF_OR_PIN	14

Year/Month/Day/Hour/Minute/ Second 为验证通过时的时间

WorkCode 返回验证时 WorkCode 值，当机器无 Workcode 功能时，该值返回 0.

4.2.6 OnEnrollFingerEx

OnEnrollFingerEx (BSTR EnrollNumber, LONG FingerIndex, LONG ActionResult, LONG TemplateLength)

登记指纹时触发该事件

【返回值】

EnrollNumber 为当前登记指纹的用户 ID

FingerIndex 为当前指纹的索引号

ActionResult 为操作结果，成功则为 0，否则返回值大于 0

TemplateLength 返回该指纹模板的长度

4.2.7 OnFinger

当机器上指纹头上检测到有指纹时触发该消息，无返回值

4.2.8 OnFingerFeature

OnFingerFeature (LONG Score)

登记用户指纹时，当有指纹按下时触发该消息

【返回值】

Score 为该指纹的质量分数

4.2.9 OnHIDNum

OnHIDNum (LONG CardNumber)

当刷卡时触发该消息

【返回值】

CardNumber 为该卡的卡号 卡类型可以是 ID 卡，HID 卡，MIFARE 卡需要被作为 ID 卡使用时才会触发该事件

4.2.10 OnNewUser

OnNewUser (LONG EnrollNumber)

当成功登记新用户时触发该消息

【返回值】

EnrollNumber 为新登记用户的 UserID

4.2.11 OnVerify

OnVerify (LONG UserID)

当用户验证时触发该消息

【返回值】

当验证成功是 UserID 为该用户 ID，当卡验证通过时，该值返回卡号，验证失败时返回值为-1

4.2.12 OnWriteCard

OnWriteCard (LONG EnrollNumber, LONG ActionResult, LONG Length)

当机器进行写卡操作时触发该事件

【返回值】

EnrollNumber 为当前需写如卡内用户的用户 ID

ActionResult 为写卡操作的结果，0 为成功，其他值为失败

Length 为写入卡内总的文件大小

4.2.13 OnEmptyCard

OnEmptyCard (LONG ActionResult)

当清空 MIFARE 卡操作时触发该事件

【返回值】

ActionResult 为清卡操作的结果，0 为成功，其他值为失败

4.2.14 OnEMData

OnEMData (LONG DataType, LONG DataLen, CHAR* DataBuffer)

当机器向 SDK 发送未知事件时，触发该事件

【返回值】

DataType 为返回事件类型

DataLen 为整个数据长度

DataBuffer 为实际数据

5 常用函数介绍

5.1 连接机器相关函数

5.1.1 Connect_Net

【函数定义】

VARIANT_BOOL Connect_Net([in] BSTR IPAdd, [in] long Port)

【功能】

通过 IP 地址连接机器，和机器建立一个网络连接

【参数】

IPAdd:

机器的 IP 地址。

Port:

连接机器时使用的端口号，默认为 4370.

【返回值】

连接成功返回 True，否则返回 False

【相关函数】

Disconnect、Connect_Com

5.1.2 Connect_Com

【函数定义】

VARIANT_BOOL Connect_Com([in] long ComPort, [in] long MachineNumber, [in] long BaudRate)

【功能】

通过串口连接机器，即 RS232 或 RS485 连接

注:对于有的机器采用 USB client 与 PC 进行连接通讯也使用该函数,但需先安装 USB client 驱动以虚拟出串口出来，成功之后可在 PC 端设备管理器内查看到该串口号,也可程序自动寻找虚拟串口号，具体实现可参考 DEMO 中的 USBClient 部分

【参数】

ComPort:

需要连接机器的 PC 串口号

MachineNumber:

机器号

BaudRate:

波特率

【返回值】

连接成功返回 True，否则返回 False

【相关函数】

Disconnect、Connect_Net

5.1.3 Disconnect

【函数定义】

Disconnect(void)

【功能】

断开连接的机器，释放相关资源

【参数】

无

【返回值】

无

【相关函数】

Connect_Net、Connect_Com

5.2 数据管理相关函数

5.2.1 考勤记录数据

5.2.1.1 ReadGeneralLogData

【函数定义】

VARIANT_BOOL ReadGeneralLogData([in] long dwMachineNumber)

【功能】

读取考勤记录到 PC 的内部缓冲区，同 ReadAllGLogData

【参数】

dwMachineNumber:

机器号

【返回值】

成功返回 True，否则返回 False

【相关函数】

ReadAllGLogData、SSR_GetGeneralLogData、ClearGLog

5.2.1.2 ReadAllGLogData

【函数定义】

VARIANT_BOOL ReadAllGLogData ([in] long dwMachineNumber)

【功能】

读取考勤记录到 PC 的内部缓冲区，同 ReadGeneralLogData

【参数】

dwMachineNumber:

机器号

【返回值】

成功返回 True，否则返回 False

【相关函数】

ReadGeneralLogData、SSR_GetGeneralLogData、ClearGLog

5.2.1.3 SSR_GetGeneralLogData

【函数定义】

VARIANT_BOOL SSR_GetGeneralLogData([in] LONG dwMachineNumber, [out] BSTR* dwEnrollNumber, [out] LONG* dwVerifyMode, [out] LONG* dwInOutMode, [out] LONG* dwYear, [out] LONG* dwMonth, [out] LONG* dwDay, [out] LONG* dwHour, [out] LONG* dwMinute, [out] LONG* dwSecond, [out] LONG* dwWorkcode)

【功能】

从内部缓冲区中逐一读取考勤记录，使用该函数前，可使用 ReadAllGLogData 或 ReadGeneralLogData 将考勤记录从机器读到 PC 内部缓冲区。该函数每执行一次，指向考勤记录的指针指向下一条记录

【参数】

dwMachineNumber

机器号

dwEnrollNumber

指向 BSTR 型变量的指针，值接收该考勤记录的用户 ID 号，可最大支持 24 位

dwVerifyMode

指向 long 型变量的指针，其值接收记录的验证方式，具体含义如下：

一般情况下：0 为密码验证，1 为指纹验证，2 为卡验证

多种验证方式模式下：

FP_OR_PW_OR_RF	0
FP	1
PIN	2
PW	3
RF	4
FP_OR_PW	5
FP_OR_RF	6
PW_OR_RF	7
PIN_AND_FP	8
FP_AND_PW	9
FP_AND_RF	10
PW_AND_RF	11
FP_AND_PW_AND_RF	12
PIN_AND_FP_AND_PW	13
FP_AND_RF_OR_PIN	14

dwInOutMode

指向 long 型变量的指针，其值接收记录的考勤状态，具体含义如下：

默认 0—Check-In 1—Check-Out 2—Break-Out
3—Break-In 4—OT-In 5—OT-Out

dwYear, dwMonth, dwDay, dwHour, dwMinute, dwSecond,

都为指向 long 型变量的指针，其值分别接收考勤记录的日期和时间

dwWorkcode

指向 long 型变量的指针，其值接收记录的 Workcode 值

【返回值】

成功返回 True,否则返回 False

【相关函数】

ReadGeneralLogData

5.2.1.4 ClearGLog

【函数定义】

VARIANT_BOOL ClearGLog([in] long dwMachineNumber)

【功能】

清除机器内所有考勤记录

【参数】

dwMachineNumber

机器号

【返回值】

成功返回 True,否则返回 False

【相关函数】

ClearSLog、ClearKeeperData

5.2.2 操作记录相关

5.2.2.1 ReadSuperLogData

【函数定义】

VARIANT_BOOL ReadSuperLogData([in]long dwMachineNumber)

【功能】

读取操作记录到 PC 的内部缓冲区，同 ReadAllSLogData

【参数】

dwMachineNumber:

机器号

【返回值】

成功返回 True，否则返回 False

【相关函数】

ReadAllSLogData、GetAllSLogData、GetSuperLogData、ClearSLog

5.2.2.2 ReadAllSLogData

【函数定义】

VARIANT_BOOL ReadAllSLogData([in] long dwMachineNumber)

【功能】

读取操作记录到 PC 的内部缓冲区，同 ReadSuperLogData

【参数】

dwMachineNumber:

机器号

【返回值】

成功返回 True，否则返回 False

【相关函数】

ReadSuperLogData、GetAllSLogData、GetSuperLogData、ClearSLog

5.2.2.3 GetSuperLogData

【函数定义】

VARIANT_BOOL GetSuperLogData([in] long dwMachineNumber, [out] long* dwTMachineNumber, [out] long* dwSEnrollNumber, [out] long* Params4, [out] long* Params1, [out] long* Params2, [out] long* dwManipulation, [out] long* Params3, [out] long* dwYear, [out] long* dwMonth, [out] long* dwDay, [out] long* dwHour, [out] long* dwMinute)

【功能】

从内部缓冲区中逐一读取操作记录，使用该函数前，可使用 ReadAllSLogData 或 ReadSuperLogData 将操作记录从机器读到 PC 内部缓冲区。该函数每执行一次，指向操作记录的指针指向下一条记录，同 GetSuperLogData2 不同的仅是 GetSuperLogData2 能获得精确到秒的操作记录时间

【参数】

dwMachineNumber

机器号

dwTMachineNumber

指向 long 型变量的指针，其值接收操作记录的机器号
dwSEnrollNumber

指向 long 型变量的指针，其值接收该操作记录的管理者 ID 号
Params4

指向 long 型变量的指针，其值含义视 dwManipulation 不同而不同
Params1

指向 long 型变量的指针，其值含义视 dwManipulation 不同而不同
Params2

指向 long 型变量的指针，其值含义视 dwManipulation 不同而不同
dwManipulation

指向 long 型变量的指针，其值接收操作类型值，具体含义如下：

dwManipulation 值	dwManipulation 含义	Params1	Params2	Params3	Params4
0	开机				
1	关机				
3	报警	报警类型，58 为错按报警，54 为门磁报警，53 为出门开关开门，55 为拆机报警，65535 时为关闭报警			
4	进入菜单				
5	更改设置	所设置的选项的编号			
6	登记指纹	被操作用户的 ID	操作结果，0 为成功，其他为失败	登记的指纹索引	指纹模板长度（为 2 时表示为胁迫指纹）
7	登记密码				
14	创建 MF 卡	被操作用户的 ID		写入 MF 卡的指纹数量	写入 MF 卡内的指纹数据大小
20	把卡中数据复制到机器内			读入 MF 卡的指纹数量	
22	恢复出厂设置				
30	登记新用户	被操作用户的 ID	操作结果		
32	胁迫报警	是否为验证报警，0 为按键报警，1 为验证报警	注：当为验证报警，此时 dwSEnrollNumber 返回被胁迫的用户号		
34	反潜	是否反潜			

Params3

指向 long 型变量的指针，其值含义视 dwManipulation 不同而不同
dwYear, dwMonth, dwDay, dwHour, dwMinute

都为指向 long 型变量的指针，其值分别接收操作记录的日期和时间

【返回值】

成功返回 True,否则返回 False

【相关函数】

GetSuperLogData2、GetAllSLogData

5.2.2.4 GetAllSLogData

【函数定义】

GetAllSLogData([in] long dwMachineNumber, [out] long* dwTMachineNumber, [out] long* dwSEnrollNumber, [out] long* dwSMachineNumber, [out] long* dwGEnrollNumber, [out] long* dwGMachineNumber, [out] long* dwManipulation, [out] long* dwBackupNumber, [out] long* dwYear, [out] long* dwMonth, [out] long* dwDay, [out] long* dwHour, [out] long* dwMinute)

【功能】

从内部缓冲区中逐一读取操作记录,使用该函数前,可使用 ReadAllSLogData 或 ReadSuperLogData 将操作记录从机器读到 PC 内部缓冲区。该函数每执行一次,指向操作记录的指针指向下一条记录,同 GetSuperLogData 完全一样,只是为了兼容,接口名称不同而已

【参数】

同 GetSuperLogData

【返回值】

成功返回 True,否则返回 False

【相关函数】

GetSuperLogData

5.2.2.5 ClearSLog

【函数定义】

VARIANT_BOOL ClearSLog([in] long dwMachineNumber)

【功能】

清除机器内所有操作记录

【参数】

dwMachineNumber
机器号

【返回值】

成功返回 True,否则返回 False

【相关函数】

ClearGLog、ClearKeeperData

5.2.2.6 GetSuperLogData2

【函数定义】

VARIANT_BOOL GetSuperLogData2([in] LONG dwMachineNumber, [out] LONG* dwTMachineNumber, [out] LONG* dwSEnrollNumber, [out] LONG* Params4, [out] LONG* Params1, [out] LONG* Params2, [out] LONG* dwManipulation, [out] LONG* Params3, [out] LONG* dwYear, [out] LONG* dwMonth, [out] LONG* dwDay, [out] LONG* dwHour,[out] LONG* dwMinute, [out] LONG* dwSecs)

【功能】

从内部缓冲区中逐一读取操作记录,使用该函数前,可使用 ReadAllSLogData 或 ReadSuperLogData 将操作记录从机器读到 PC 内部缓冲区。该函数每执行一次,指向操作记录的指针指向下一条记录,同

GetSuperLogData 不同的仅是 GetSuperLogData2 能获得精确到秒的操作记录时间

【参数】

dwYear, dwMonth, dwDay, dwHour, dwMinute, dwSecs

都为指向 long 型变量的指针，其值分别接收操作记录的日期和时间

其他参数参考 GetSuperLogData 相应参数说明

【返回值】

成功返回 True，否则返回 False

【相关函数】

ReadAllSLogData、GetSuperLogData

5.2.3 用户信息相关

5.2.3.1 ReadAllUserID

【函数定义】

VARIANT_BOOL ReadAllUserID([in] long dwMachineNumber)

【功能】

读取所有的用户信息到 PC 内存中，包括用户编号，密码，姓名，卡号等，指纹模板除外。在该函数执行完成后，可调用函数 GetAllUserID 取出用户信息

【参数】

dwMachineNumber

机器号

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_GetAllUserInfo

5.2.3.2 SSR_EnableUser

【函数定义】

VARIANT_BOOL SSR_EnableUser([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] VARIANT_BOOL bFlag)

【功能】

设置用户是否可用

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

bFlag

用户启用标志，True 为启用，False 为禁用

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.2.3.3 SetUserInfoEx

【函数定义】

VARIANT_BOOL SetUserInfoEx([in] LONG dwMachineNumber, [in] LONG dwEnrollNumber, [in] LONG VerifyStyle, [in] BYTE* Reserved)

【功能】

上传用户验证方式或组验证方式。只有有多种验证方式的机器可支持该函数

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

VerifyStyle

验证方式

彩屏门禁指纹机的值为:0 表示使用组验证方式, 128(FP/PW/RF), 129(FP), 130(PIN), 131(PW), 132(RF), 133(FP&RF), 134(FP/PW), 135(FP/RF), 136(PW/RF), 137(PIN&FP), 138(FP&PW), 139(PW&RF), 140(FP&PW&RF), 141(PIN&FP&PW), 142(FP&RF/PIN).

Reserved

保留参数, 暂无意义

【返回值】

成功返回 True, 否则返回 False

【相关函数】

GetUserInfoEx、DeleteUserInfoEx

5.2.3.4 GetUserInfoEx

【函数定义】

VARIANT_BOOL GetUserInfoEx([in] LONG dwMachineNumber, [in] LONG dwEnrollNumber, [out] LONG* VerifyStyle, [out] BYTE* Reserved)

【功能】

获取用户验证方式, 只有具有多种验证方式的机器可支持该函数

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

VerifyStyle

该参数返回由 dwEnrollNumber 描述的用户验证方式, 具体含义如下:

彩屏门禁指纹机的值为:0 表示使用组验证方式, 128(FP/PW/RF), 129(FP), 130(PIN), 131(PW), 132(RF), 133(FP&RF), 134(FP/PW), 135(FP/RF), 136(PW/RF), 137(PIN&FP), 138(FP&PW), 139(PW&RF), 140(FP&PW&RF), 141(PIN&FP&PW), 142(FP&RF/PIN).

Reserved

保留参数, 无意义

【返回值】

成功返回 True, 否则返回 False

【相关函数】

SetUserInfoEx、DeleteUserInfoEx

5.2.3.5 DeleteUserInfoEx

【函数定义】

VARIANT_BOOL DeleteUserInfoEx([in] LONG dwMachineNumber, [in] LONG dwEnrollNumber)

【功能】

删除指定用户设置的多种验证方式，只有有多种验证方式的机器可支持该函数

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetUserInfoEx、GetUserInfoEx

5.2.3.6 SSR_GetAllUserInfo

【函数定义】

VARIANT_BOOL SSR_GetAllUserInfo([in] LONG dwMachineNumber, [out] BSTR* dwEnrollNumber, [out] BSTR* Name, [out] BSTR* Password, [out] LONG* Privilege, [out] VARIANT_BOOL* Enabled)

【功能】

取得所有用户信息。在该函数执行之前，可用 ReadAllUserID 读取到所有用户信息到内存，SSR_GetAllUserInfo 每执行一次，指向用户信息指针移到下一记录，当读完所有用户信息后，函数返回 False。

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

Name

用户姓名

Password

用户密码

Privilege

用户权限，3 为管理员，0 为普通用户

Enabled

用户启用标志，1 为启用，0 为禁用

【返回值】

成功返回 True，否则返回 False

【相关函数】

ReadAllUserID、GetAllUserInfo

5.2.3.7 SSR_GetUserInfo

【函数定义】

VARIANT_BOOL SSR_GetUserInfo([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [out] BSTR* Name, [out] BSTR* Password, [out] LONG* Privilege, [out] VARIANT_BOOL* Enabled)

【功能】

获取指定用户的用户信息

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

Name

该参数返回由 dwEnrollNumber 描述的用户的姓名

Password

该参数返回由 dwEnrollNumber 描述的用户的密码

Privilege

该参数返回由 dwEnrollNumber 描述的用户的权限，3 为管理员，0 为普通用户

Enabled

用户启用标志，1 为启用，0 为禁用

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_SetUserInfo

5.2.3.8 SSR_SetUserInfo

【函数定义】

VARIANT_BOOL SSR_SetUserInfo([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] BSTR Name, [in] BSTR Password, [in] LONG Privilege, [in] VARIANT_BOOL Enabled)

【功能】

设置指定用户的用户信息，若机器内没用该用户，则会创建该用户

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

Name

用户姓名

Password

用户密码

Privilege

用户权限，3 为管理员，0 为普通用户

Enabled

用户启用标志，1 为启用，0 为禁用

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_GetUserInfo

5.2.4 登记数据（同时包括用户信息和指纹）

5.2.4.1 SSR_DeleteEnrollData

【函数定义】

VARIANT_BOOL SSR_DeleteEnrollData([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] LONG dwBackupNumber)

【功能】

删除登记数据

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

dwBackupNumber

指纹索引

一般范围为 0-9，同时会查询该用户是否还有其他指纹和密码，如都没有，则删除该用户

当为 10 是代表删除的是密码，同时会查询该用户是否有指纹数据，如没有，则删除该用户

11 是代表删除该用户所有指纹数据，12 代表删除该用户（包括所有指纹和卡号、密码数据）

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_SetUserInfo、SSR_SetUserTmp、SSR_DeleteEnrollDataExt

5.2.4.2 SSR_DeleteEnrollDataExt

【函数定义】

VARIANT_BOOL SSR_DeleteEnrollDataExt([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] LONG dwBackupNumber)

【功能】

删除登记数据，和 SSR_DeleteEnrollData 不同的是删除所有指纹数据可用参数 13 实现，该函数具有更高效率

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

dwBackupNumber

指纹索引

一般范围为 0-9，同时会查询该用户是否还有其他指纹和密码，如都没有，则删除该用户

当为 10 是代表删除的是密码，同时会查询该用户是否有指纹数据，如没有，则删除该用户

11 和 13 是代表删除该用户所有指纹数据，12 代表删除该用户（包括所有指纹和卡号、密码数据）

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_SetUserInfo、SSR_SetUserTmp、SSR_DeleteEnrollData

5.2.5 指纹模板相关

5.2.5.1 ReadAllTemplate

【函数定义】

VARIANT_BOOL ReadAllTemplate([in] LONG dwMachineNumber)

【功能】

读取机器内所有指纹模板到 PC 内存，该函数一次性将所有指纹数据读到内存，相比于一个一个从机器获取，该函数具有更高效率

【参数】

dwMachineNumber

机器号

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.2.5.2 SSR_GetUserTmp

【函数定义】

VARIANT_BOOL SSR_GetUserTmp([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber,[in] LONG dwFingerIndex, [in] BYTE* TmpData, [in] LONG* TmpLength)

【功能】

以二进制方式获取用户指纹模板,和 SSR_GetUserTmpStr 不同的仅是指纹模板格式不同而已

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

dwFingerIndex

指纹索引号，一般范围为 0-9

TmpData

该参数返回指纹模板数据

TmpLength

该参数返回指纹模板数据长度

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_SetUserTmp、SSR_GetUserTmpStr

5.2.5.3 SSR_GetUserTmpStr

【函数定义】

VARIANT_BOOL SSR_GetUserTmpStr([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] LONG dwFingerIndex, [out] BSTR* TmpData, [out] LONG* TmpLength)

【功能】

以字符串方式获取用户指纹模板，和 SSR_GetUserTmp 不同的仅是指纹模板格式不同而已

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

dwFingerIndex

指纹索引号，一般范围为 0-9

TmpData

该参数返回指纹模板数据

TmpLength

该参数返回指纹模板数据长度

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_GetUserTmp、SSR_SetUserTmpStr

5.2.5.4 SSR_SetUserTmp

【函数定义】

VARIANT_BOOL SSR_SetUserTmp([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] LONG dwFingerIndex, [in] BYTE* TmpData)

【功能】

以二进制方式上传用户指纹模板，和 SSR_SetUserTmpStr 不同的仅是指纹模板格式不同而已

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

dwFingerIndex

指纹索引号，一般范围为 0-9

TmpData

指纹模板

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_GetUserTmp、SSR_SetUserTmpStr

5.2.5.5 SSR_SetUserTmpStr

【函数定义】

VARIANT_BOOL SSR_SetUserTmpStr ([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] LONG dwFingerIndex, [in] BSTR TmpData)

【功能】

以字符串方式上传用户指纹模板，和 SSR_SetUserTmp 不同的仅是指纹模板格式不同而已

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

dwFingerIndex

指纹索引号，一般范围为 0-9

TmpData

指纹模板

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_GetUserTmpStr、SSR_SetUserTmp

5.2.5.6 SSR_DelUserTmp

【函数定义】

VARIANT_BOOL SSR_DelUserTmp([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] LONG dwFingerIndex)

【功能】

删除用户指纹模板

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

dwFingerIndex

指纹索引号，一般范围为 0-9

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_SetUserTmp、SSR_GetUserTmp

5.2.5.7 SSR_SetUserTmpExt

【函数定义】

VARIANT_BOOL SSR_SetUserTmpExt([in] LONG dwMachineNumber, [in] LONG IsDeleted,[in] BSTR dwEnrollNumber,[in] LONG dwFingerIndex, [in] BYTE* TmpData)

【功能】

上传用户指纹模板，为 SSR_SetUserTmp 的加强版

【参数】

dwMachineNumber

机器号

IsDeleted

删除标志，即上传时若机器内已存在该用户的指定索引号的指纹是否覆盖原指纹，1 为覆盖，0 为不覆盖

dwEnrollNumber

用户号

dwFingerIndex

指纹索引，一般范围为 0—9

TmpData

指纹模板

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_SetUserTmp

5.2.5.8 SSR_DelUserTmpExt

【函数定义】

VARIANT_BOOL SSR_DelUserTmpExt([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] LONG dwFingerIndex)

【功能】

删除指定用户的指纹模板，和 DelUserTmp 的区别在与前者可以支持 24 位的用户号

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

dwFingerIndex

指纹索引，一般范围为 0—9，当该值为 13 时删除该用户的所有指纹

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_DelUserTmp

5.2.5.9 SetUserTmpEx

【函数定义】

VARIANT_BOOL SetUserTmpEx([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] LONG dwFingerIndex, [in] LONG Flag, [in] BYTE* TmpData)

【功能】

以二进制方式上传用户普通指纹模板或者胁迫指纹模板,和 SetUserTmpExStr 不同的仅是指纹模板格式不同而已,注意: 机器上必须已存在该用户或者将用户信息同时上传, 相同用户的相同索引号模板如果已经登记, 则覆盖。注: 要求机器固件支持胁迫指纹功能 (固件内部版本号 Ver6.60 及以上)。

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

dwFingerIndex

指纹索引号

Flag

标示指纹模板是否有效或者是否为胁迫指纹， 0 表示指纹模板无效，1 表示指纹模板有效，3 表示为胁迫指纹

TmpData

指纹模板数据

TmpLength

指纹模板长度

【返回值】

成功返回 True，否则返回 False

【相关函数】

GetUserTmpEx, SSR_SetUserInfo, BeginBatchUpdate, BatchUpdate

5.2.5.10 SetUserTmpExStr

【函数定义】

VARIANT_BOOL SetUserTmpExStr([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] LONG dwFingerIndex,[in] LONG Flag, [in] BSTR TmpData)

【功能】

以字符串方式上传用户普通指纹模板或者胁迫指纹模板,和 SetUserTmpEx 不同的仅是指纹模板格式不同而已,注意: 机器上必须已存在该用户或者将用户信息同时上传, 相同用户的相同索引号模板如果已经登记, 则覆盖。注: 要求机器固件支持胁迫指纹功能(固件内部版本号 Ver6.60 及以上)。

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

dwFingerIndex

指纹索引号

Flag

标示指纹模板是否有效或者是否为胁迫指纹， 0 表示指纹模板无效，1 表示指纹模板有效，3 表示为胁迫指纹

TmpData

指纹模板数据

TmpLength

指纹模板长度

【返回值】

成功返回 True，否则返回 False

【相关函数】

GetUserTmpExStr, SSR_SetUserInfo, BeginBatchUpdate, BatchUpdate

5.2.5.11 GetUserTmpEx

【函数定义】

VARIANT_BOOL GetUserTmpEx([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] LONG dwFingerIndex,[out] LONG * Flag, [out] BYTE* TmpData, [out] LONG* TmpLength)

【功能】

以二进制方式下载用户普通指纹模板或者胁迫指纹模板,和 GetUserTmpExStr 不同的仅是指纹模板格式不同而已。注：要求机器固件支持胁迫指纹功能（固件内部版本号 Ver6.60 及以上）。

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

dwFingerIndex

指纹索引号

Flag

标示指纹模板是否有效或者是否为胁迫指纹， 0 表示指纹模板无效，1 表示指纹模板有效，3 表示为胁迫指纹

TmpData

指纹模板数据

TmpLength

指纹模板长度

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetUserTmpEx, SSR_GetUserInfo

5.2.5.12 GetUserTmpExStr

【函数定义】

VARIANT_BOOL GetUserTmpExStr([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] LONG dwFingerIndex,[out] LONG * Flag, [out] BSTR* TmpData, [out] LONG* TmpLength)

【功能】

以字符串方式下载用户普通指纹模板或者胁迫指纹模板,和 GetUserTmpEx 不同的仅是指纹模板格式不同而已。注：要求机器固件支持胁迫指纹功能（固件内部版本号 Ver6.60 及以上）。

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

dwFingerIndex

指纹索引号

Flag

标示指纹模板是否有效或者是否为胁迫指纹， 0 表示指纹模板无效，1 表示指纹模板有效，3 表示为胁迫指纹

TmpData

指纹模板数据

TmpLength

指纹模板长度

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetUserTmpExStr, SSR_GetUserInfo

5.2.6 其他

5.2.6.1 短消息相关

5.2.6.1.1 SetSMS

【函数定义】

VARIANT_BOOL SetSMS([in] LONG dwMachineNumber, [in] LONG ID, [in] LONG Tag, [in] LONG ValidMinutes, [in] BSTR StartTime, [in] BSTR Content)

【功能】

添加短消息到机器内，如果要设置用户短消息，需要使用 SetUserSMS 将该消息分配给某用户

【参数】

dwMachineNumber

机器号

ID

短消息编号

Tag

短消息类型，253 为公共短消息，254 为个人短消息，255 为预留短消息

ValidMinutes

短消息的有效时长，范围为 0——65535，即短消息会在 StartTime 开始生效，持续 ValidMinutes

StartTime

短消息开始生效的时间，输入字符串格式需为 yyyy-mm-dd hh:mm:ss

Content

短消息内容

【返回值】

成功返回 True，否则返回 False

【相关函数】

GetSMS, DeleteSMS, ClearSMS, SetUserSMS

5.2.6.1.2 SSR_SetUserSMS

【函数定义】

设置用户短消息，即将机器内某个编号的短消息分配给某个用户

【功能】

VARIANT_BOOL SSR_SetUserSMS([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] LONG SMSID)

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户 ID 号

SMSID

短消息编号

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetUserSMS, SetSMS, SetUserSMS, GetSMS, DeleteSMS, ClearSMS

5.2.6.1.3 GetSMS

【函数定义】

VARIANT_BOOL GetSMS([in] LONG dwMachineNumber, [in] LONG ID, [out] LONG* Tag, [out] LONG* ValidMinutes, [out] BSTR* StartTime, [out] BSTR *Content)

【功能】

根据短消息编号获取机器内该消息的详细信息，包括内容，开始时间，消息类型，有效时长

【参数】

dwMachineNumber

机器号

ID

短消息编号

Tag

短消息类型，253 为公共短消息，254 为个人短消息，255 为预留短消息

ValidMinutes

短消息的有效时长，范围为 0——65535，即短消息会在 StartTime 开始生效，持续 ValidMinutes

StartTime

短消息开始生效的时间

Content

短消息内容

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetSMS、GetSMS、DeleteSMS、ClearSMS

5.2.6.1.4 DeleteSMS

【函数定义】

VARIANT_BOOL DeleteSMS([in] LONG dwMachineNumber, [in] LONG ID)

【功能】

删除机器内指定编号的短消息

【参数】

dwMachineNumber

机器号

ID

短消息编号

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetSMS、ClearSMS

5.2.6.1.5 SSR_DeleteUserSMS

【函数定义】

VARIANT_BOOL SSR_DeleteUserSMS([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] LONG SMSID)

【功能】

删除指定用户的指定编号的短消息，只删除该对应关系（短消息分配给用户），不会删除短消息本身

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户 ID 号

SMSID

短消息编号

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_SetUserSMS

5.2.6.1.6 ClearUserSMS

【函数定义】

VARIANT_BOOL ClearUserSMS([in] LONG dwMachineNumber)

【功能】

清除所有的对应关系（短消息分配给用户），不会删除短消息本身

【参数】

dwMachineNumber

机器号

【返回值】

成功返回 True，否则返回 False

【相关函数】

ClearSMS、SetSMS

5.2.6.1.7 ClearSMS

【函数定义】

VARIANT_BOOL ClearSMS([in] LONG dwMachineNumber)

【功能】

清除机器内所有短消息，即清除的是消息本身

【参数】

dwMachineNumber

机器号

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetSMS

5.2.6.2 WorkCode 相关

5.2.6.2.1 SSR_GetWorkCode

【函数定义】

VARIANT_BOOL SSR_GetWorkCode([in] LONG AWorkCode, [out] BSTR *Name)

【功能】

获取指定 WorkCode 编号的名字

【参数】

AworkCode

WorkCode 编号

Name

该参数返回获取的由 AworkCode 描述的 WorkCode 编号的名字

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_SetWorkCode、SSR_DeleteWorkCode、SSR_ClearWorkCode

5.2.6.2.2 SSR_SetWorkCode

【函数定义】

VARIANT_BOOL SSR_SetWorkCode([in] LONG AWorkCode, [in] BSTR Name)

【功能】

设置指定编号的 WorkCode

【参数】

AworkCode

WorkCode 编号

Name

欲设置的由 AworkCode 描述的 WorkCode 编号的值

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_GetWorkCode、SSR_DeleteWorkCode、SSR_ClearWorkCode

5.2.6.2.3 SSR_DeleteWorkCode

【函数定义】

VARIANT_BOOL SSR_DeleteWorkCode([in] LONG AWorkCode)

【功能】

删除指定编号的 WorkCode

【参数】

AworkCode

WorkCode 编号

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_GetWorkCode、SSR_SetWorkCode、SSR_ClearWorkCode

5.2.6.2.4 SSR_ClearWorkCode

【函数定义】

VARIANT_BOOL SSR_ClearWorkCode(void)

【功能】

删除所有自定义的 WorkCode

【参数】

无

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_GetWorkCode、SSR_SetWorkCode、SSR_DeleteWorkCode

5.2.6.3 节假日相关

5.2.6.3.1 SSR_GetHoliday

【函数定义】

VARIANT_BOOL SSR_GetHoliday([in] LONG dwMachineNumber, [in] LONG HolidayID, [out] LONG* BeginMonth, [out] LONG* BeginDay, [out] LONG* EndMonth, [out] LONG* EndDay, [out] LONG* TimeZoneID)

【功能】

根据节假日编号获取机器上的节假日设置

【参数】

dwMachineNumber

机器号

HolidayID

节假日编号

BeginMonth、BeginDay

该参数接收节假日的开始日期

EndMonth、EndDay

该参数接收节假日的结束日期

TimeZoneID

该参数接收节假日的时间段编号

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_SetHoliday

5.2.6.3.2 SSR_SetHoliday

【函数定义】

VARIANT_BOOL SSR_SetHoliday([in] LONG dwMachineNumber, [in] LONG HolidayID, [in] LONG BeginMonth, [in] LONG BeginDay, [in] LONG EndMonth, [in] LONG EndDay, [in] LONG TimeZoneID)

【功能】

设置节假日

【参数】

dwMachineNumber

机器号

HolidayID

假日编号

BeginMonth、BeginDay

节假日开始时间

EndMonth、EndDay

节假日结束时间

TimeZoneID

节假日使用的时间段编号

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_GetHoliday

5.2.6.4 夏令时相关

5.2.6.4.1 SetDaylight

【函数定义】

VARIANT_BOOL SetDaylight([in] LONG dwMachineNumber, [in] LONG Support, [in] BSTR BeginTime, [in] BSTR EndTime)

【功能】

设置是否使用夏令时功能，以及夏令时开始时间和结束时间

【参数】

dwMachineNumber

机器号

Support

是否启用夏令时功能，1 为启用，0 为禁用

BeginTime

夏令时开始时间,日期格式为 mmdd hh:mm

EndTime

夏令时结束时间,日期格式为 mmdd hh:mm

【返回值】

成功返回 True，否则返回 False

【相关函数】

GetDaylight

5.2.6.4.2 GetDaylight

【函数定义】

VARIANT_BOOL GetDaylight ([in] LONG dwMachineNumber, [out] LONG* Support, [out] BSTR* BeginTime, [out] BSTR* EndTime)

【功能】

获取当前机器上夏令时功能的设置

【参数】

dwMachineNumber

机器号

Support

该参数返回值表示是否启用夏令时功能，1 为启用，0 为禁用

BeginTime

该参数返回值表示夏令时的开始时间,日期格式为 mmdd hh:mm

EndTime

该参数返回值表示夏令时的结束时间,日期格式为 mmdd hh:mm

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetDaylight

5.2.6.5 快捷键定义

5.2.6.5.1 SSR_SetShortcutkey

【函数定义】

VARIANT_BOOL SSR_SetShortcutkey([in] LONG ShortKeyID, [in] LONG ShortKeyFun, [in] LONG StateCode, [in] BSTR StateName, [in] LONG StateAutoChange, [in] BSTR StateAutoChangeTime)

【功能】

对某一功能键进行设置，相当于彩屏机内的功能键定义功能

【参数】

ShortKeyID

指定键的 ID，对应关系为：F1 — 1, F2 — 2, F3 — 3

ShortKeyFun

设置指定键的功能。

对应关系为：0：未定义，1：状态键，2：workcode，3：查看短消息。

请注意，该参数的取值将影响后面四个参数的设定，请参见下面四个参数的说明。

StateCode

状态键的对应状态值。

如果不是设置状态键，该参数的值将被忽略（即：如果 ShortKeyFun 参数的值不为 1，则该参数的值将被忽略）。

如果是将其设为状态键（即：ShortKeyFun 参数的值为 1），则该参数指定状态键的状态值，有效范围为[0 - 255]。状态键的状态值不能有重复，如果有重复，对函数的调用将失败。如：F2 键是状态键，它的状态值为 2，你现在调用该函数去设置 F3 为状态键，也指定其状态值为 2，那么此时对函数的调用将失败。

StateName

状态键的名称。

如果不是设置状态键，该参数的值将被忽略（即：如果 ShortKeyFun 参数的值不为 1，则该参数的值将被忽略）。

如果是将其设为状态键（即：ShortKeyFun 参数的值为 1），则该参数指定状态键的名称，最大支持 18 个字符。

StateAutoChange

如果不是设置状态键，该参数的值将被忽略（即：如果 ShortKeyFun 参数的值不为 1，则该参数的值将被忽略）。

如果是将其设为状态键（即：ShortKeyFun 参数的值为 1），则该参数指定状态键的自动切换是否打

开。0：关闭，1：打开。

StateAutoChangeTime

如果不是设置状态键，该参数的值将被忽略（即：如果 ShortKeyFun 参数的值不为 1，则该参数的值将被忽略）。

如果是将其设为状态键，则该参数设置状态键的自动切换时间。要求如下：

- 1、“08:30;09:00;08:00;12:00;11:12;00:00;00:00;”
- 2、小时与分钟用“:”分隔，天与天之间用“;”分隔，中间不能有空格。
- 3、必须指定每天的自动切换时间（以一个星期为周期），即机器到该时间后考勤状态变为该状态（名称为 StateName 描述，值为 StateCode 定义）。如果某天不自动切换，则指定其小时和分钟均为 0。

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_GetShortkey

5.2.6.5.2 SSR_GetShortkey

【函数定义】

VARIANT_BOOL SSR_GetShortkey([in] LONG ShortKeyID, [in] LONG *ShortKeyFun, [in] LONG *StateCode, [in] BSTR *StateName, [in] LONG *AutoChange, [in] BSTR *AutoChangeTime)

【功能】

获取某一功能键的设置

【参数】

ShortKeyID

要获取的指定键的 ID，对应关系为：F1 ? 1, F2 ? 2, F3 ? 3

ShortKeyFun

返回指定键的功能。

对应关系为：0：未定义，1：状态键，2：workcode，3：查看短消息。

StateCode

如果指定键是状态键（即：ShortKeyFun=1），则该参数返回该状态键的状态值。否则返回 0。

StateName

如果指定键是状态键（即：ShortKeyFun=1），则该参数返回该状态键的名称。否则返回一个空字符串。

AutoChange

如果指定键是状态键（即：ShortKeyFun=1），则该参数返回该状态键是否是自动切换。否则返回 0。

AutoChangeTime

如果指定键是状态键（即：ShortKeyFun=1），则该参数返回该状态键的自动切换时间（以字符串形式表示）。否则返回一个空字符串。

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_SetShortkey

5.2.6.6 指纹模板转换相关

5.2.6.6.1 GetFPTempLength

【函数定义】

GetFPTempLength([in] BYTE* dwEnrollData, [out] long * Len)

【功能】

计算指定指纹模板长度

【参数】

dwEnrollData

指向指纹模板的指针

Len

该参数返回由 dwEnrollData 描述的指纹模板的长度

【返回值】

成功返回 True，否则返回 False

【相关函数】

GetFPTempLengthStr

5.2.6.6.2 GetFPTempLengthStr

【函数定义】

GetFPTempLengthStr([in] BSTR dwEnrollData, [out] long * Len)

【功能】

计算指定指纹模板长度

【参数】

dwEnrollData

字符串形式表示的指纹模板

Len

该参数返回由 dwEnrollData 描述的指纹模板的长度

【返回值】

成功返回 True，否则返回 False

【相关函数】

GetFPTempLength

5.2.6.6.3 FPTempConvert

【函数定义】

VARIANT_BOOL FPTempConvert([in] BYTE* TmpData1, [out] BYTE* TmpData2, [out] long *Size)

【功能】

脱机指纹模板转换为 BIOKEY 指纹模板，和 FPTempConvertStr 的区别在于数据格式不同而已

【参数】

TmpData1

欲转换的脱机指纹模板

TmpData2

该参数返回转换后的 BIOKEY 指纹模板

Size

该参数返回转换后的 BIOKEY 指纹模板的大小

【返回值】

成功返回 True，否则返回 False

【相关函数】

FPTempConvertStr、FPTempConvertNew、FPTempConvertNewStr

5.2.6.6.4 FPTempConvertStr

【函数定义】

VARIANT_BOOL FPTempConvertStr([in] BSTR TmpData1,[out]BSTR* TmpData2, [out] long *Size)

【功能】

以字符串形式将脱机指纹模板转换为 BIOKEY 指纹模板，和 FPTempConvert 的区别在于数据格式不同而已

【参数】

TmpData1

欲转换的脱机指纹模板

TmpData2

该参数返回转换后的 BIOKEY 指纹模板

Size

该参数返回转换后的 BIOKEY 指纹模板的大小

【返回值】

成功返回 True，否则返回 False

【相关函数】

FPTempConvert、FPTempConvertNew、FPTempConvertNewStr

5.2.6.6.5 FPTempConvertNew

【函数定义】

VARIANT_BOOL FPTempConvertNew([in] BYTE* TmpData1, [out] BYTE* TmpData2, [out] long *Size)

【功能】

将 BIOKEY 指纹模板转换为脱机指纹模板，和 FPTempConvertNewStr 的区别在于数据格式不同而已

【参数】

TmpData1

欲转换的脱机指纹模板

TmpData2

该参数返回转换后的脱机指纹指纹模板

Size

该参数返回转换后的脱机指纹模板的大小

【返回值】

成功返回 True，否则返回 False

【相关函数】

FPTempConvertNewStr、FPTempConvert、FPTempConvertStr

5.2.6.6.6 FPTempConvertNewStr

【函数定义】

VARIANT_BOOL FPTempConvertNewStr([in] BSTR TmpData1, [out] BSTR* TmpData2, [out] long *Size)

【功能】

以字符串形式将 BIOKEY 指纹模板转换为脱机指纹模板，和 FPTempConvertNew 的区别在于数据格式不同而已

【参数】

TmpData1

欲转换的脱机指纹模板

TmpData2

该参数返回转换后的脱机指纹指纹模板

Size

该参数返回转换后的脱机指纹模板的大小

【返回值】

成功返回 True，否则返回 False

【相关函数】

FPTempConvertNew、FPTempConvert、FPTempConvertStr

5.2.7 系统数据管理相关

5.2.7.1 ClearKeeperData

【函数定义】

VARIANT_BOOL ClearKeeperData([in] long dwMachineNumber)

【功能】

清楚机器内所有的数据

【参数】

dwMachineNumber

机器号

【返回值】

成功返回 True,否则返回 False

【相关函数】

ClearGLog, ClearSLog

5.2.7.2 ClearData

【函数定义】

VARIANT_BOOL ClearData([in] LONG dwMachineNumber, [in] LONG DataFlag)

【功能】

清除机器内由 DataFlag 指定的记录

【参数】

dwMachineNumber

机器号

DataFlag

该参数指定需清除的记录类型，范围为 1—5，具体含义如下

- | | |
|---|--------|
| 1 | 考勤记录 |
| 2 | 指纹模板数据 |
| 3 | 无 |
| 4 | 操作记录 |
| 5 | 用户信息 |

当该参数为 5 时，即删除机器内所有用户，注：此操作同时会删除机器内所有指纹模板

【返回值】

成功返回 True，否则返回 False

【相关函数】

ClearKeeperData

5.2.7.3 GetDataFile

【函数定义】

VARIANT_BOOL GetDataFile([in] LONG dwMachineNumber, [in] LONG DataFlag, [in] BSTR FileName)

【功能】

从机器获取指定数据文件

【参数】

dwMachineNumber

机器号

DataFlag

需要获取的数据文件类型

- | | |
|---|-----------------|
| 1 | 考勤记录数据文件 |
| 2 | 指纹模板数据文件 |
| 3 | 无 |
| 4 | 操作记录数据文件 |
| 5 | 用户信息数据文件 |
| 6 | 短消息数据文件 |
| 7 | 短消息与用户关系的数据文件 |
| 8 | 扩展用户信息数据文件 |
| 9 | Workcode 信息数据文件 |

FileName

接收到获取的数据文件存储文件名

【返回值】

成功返回 True，否则返回 False

【相关函数】

SaveTheDataToFile

5.2.7.4 SendFile

【函数定义】

VARIANT_BOOL SendFile([in] LONG dwMachineNumber,[in] BSTR FileName)

【功能】

发送文件到机器，一般发送到/mnt/mtdblock/下，彩屏机如传的是用户照片或宣传图片，需命名为以下格式：图片会自动被转移到相应目录下。

宣传图片命名方式：“ad_”为前缀,后加数字，范围为 1—20，后缀为.jpg，如 ad_4.jpg

用户照片命名方式：“用户 ID”+“.jpg”，如 1.jpg

【参数】

dwMachineNumber

机器号

FileName

要发送的文件名

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.2.7.5 RefreshData

【函数定义】

VARIANT_BOOL RefreshData([in] long dwMachineNumber)

【功能】

刷新机器内数据，一般在上传用户信息或指纹后调用，这样能使所作的修改立即起作用，起到同步作用。

【参数】

dwMachineNumber

机器号

【返回值】

成功返回 True, 否则返回 False

【相关函数】

无

5.3 门禁相关（时间段，组，开锁组合）

5.3.1 GetUserGroup

【函数定义】

VARIANT_BOOL GetUserGroup([in] long dwMachineNumber, [in] long dwEnrollNumber, [out] long *UserGrp)

【功能】

获取指定用户所属组号

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

UserGrp

该参数返回由 dwEnrollNumber 描述的用户的组号，一般范围为 1-5

【返回值】

成功返回 True， 否则返回 False

【相关函数】

SetUserGroup

5.3.2 SetUserGroup

【函数定义】

VARIANT_BOOL SetUserGroup([in] long dwMachineNumber, [in] long dwEnrollNumber, [in] long UserGrp)

【功能】

设置指定用户所属组

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

UserGrpl

组号，一般范围为 1-5

【返回值】

成功返回 True，否则返回 False

【相关函数】

GetUserGroup

5.3.3 GetTZInfo

【函数定义】

VARIANT_BOOL GetTZInfo([in] long dwMachineNumber, [in] long TZIndex, [out] BSTR *TZ)

【功能】

获取指定时间段编号的时间段信息

【参数】

dwMachineNumber

机器号

TZIndex

时间段序号

TZ

该参数返回由 TZIndex 描述的时间段序号的时间段信息，，每 8 位表示一个时间段，格式为 hhmmhhmm，如:101112230000235900002359000023590000235900002359 代表周日为 10:11 到 12: 23，其他周一到周六为全天的时间段

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetTZInfo

5.3.4 SetTZInfo

【函数定义】

VARIANT_BOOL SetTZInfo([in] long dwMachineNumber, [in] long TZIndex, [in] BSTR TZ)

【功能】

设置指定时间段编号的时间段信息

【参数】

dwMachineNumber

机器号

TZIndex

时间段序号

TZ

需要设置的时间段信息，每 8 位表示一个时间段，格式为 hhmmhhmm，如 101112230000235900002359000023590000235900002359 代表周日为 10:11 到 12:23，其他周一到周六为全天的时间段

【返回值】

成功返回 True，否则返回 False

【相关函数】

GetTZInfo

5.3.5 SSR_SetUnLockGroup

【函数定义】

VARIANT_BOOL SSR_SetUnLockGroup([in] LONG dwMachineNumber, [in] LONG CombNo, [in] LONG Group1,[in] LONG Group2, [in] LONG Group3, [in] LONG Group4, [in] LONG Group5)

【功能】

设置开锁组合

【参数】

dwMachineNumber

机器号

CombNo

组合编号，范围为 1—10，即机器最多支持 10 个开锁组合

Group1、Group2、Group3、Group4、Group5

开锁组合的组编号，每个开锁组合包含 5 个组编号，每个组编号范围为 1-99，如 SSR_SetUnLockGroup (1, 1, 2, 23, 14, 0, 56) 代表同时需要组 2、组 23、组 14、组 56 的人员一起验证才能开门

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_GetUnLockGroup

5.3.6 SSR_GetUnLockGroup

【函数定义】

VARIANT_BOOL SSR_GetUnLockGroup([in] LONG dwMachineNumber, [in] LONG CombNo, [out] LONG* Group1, [out] LONG* Group2, [out] LONG* Group3, [out] LONG* Group4, [out] LONG* Group5)

【功能】

根据开锁组合编号获取组合信息

【参数】

dwMachineNumber

机器号

CombNo

组合编号，范围为 1—10

Group1、Group2、Group3、Group4、Group5

这些参数返回指定开锁组合的具体组合状况，即该组合包含哪些组，返回值为组编号，每个组合最多包含 5 个组。其中每个组编号的范围为 1-99.

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_SetUnLockGroup

5.3.7 SSR_SetGroupTZ

【函数定义】

VARIANT_BOOL SSR_SetGroupTZ([in] LONG dwMachineNumber, [in] LONG GroupNo, [in] LONG Tz1, [in] LONG Tz2, [in] LONG Tz3, [in] LONG VaildHoliday, [in] LONG VerifyStyle)

【功能】

设置组时间段

【参数】

dwMachineNumber

机器号

GroupNo

组编号，范围为 1-99

Tz1、Tz2、Tz3

时间段编号，每个组可包含 3 个时间段

VaildHoliday

节假日是否有效

VerifyStyle

组验证方式，具体含义为 0(FP/PW/RF), 1(FP), 2(PIN), 3(PW), 4(RF), 5(FP&RF), 6(FP/PW), 7(FP/RF), 8(PW/RF), 9(PIN&FP), 10(FP&PW), 11(PW&RF), 12(FP&PW&RF), 13(PIN&FP&PW), 14(FP&RF/PIN)

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_GetGroupTZ

5.3.8 SSR_GetGroupTZ

【函数定义】

VARIANT_BOOL SSR_GetGroupTZ([in] LONG dwMachineNumber, [in] LONG GroupNo, [out] LONG* Tz1, [out] LONG* Tz2, [out] LONG* Tz3, LONG* VaildHoliday, LONG* VerifyStyle)

【功能】

获取组时间段

【参数】

dwMachineNumber

指纹机机号

GroupNo

组编号，该值的范围是 1-99。

Tz1 、Tz2、Tz3

这三个参数返回指定组合的三个时间段编号，每个组编号的值的范围是 1-50。

VaildHoliday

节假日是否有效。1：有效，0 无效。

VerifyStyle

指纹机验证方式。其值是：0(FP/PW/RF), 1(FP), 2(PIN), 3(PW), 4(RF), 5(FP&RF), 6(FP/PW),

7(FP/RF), 8(PW/RF), 9(PIN&FP), 10(FP&PW), 11(PW&RF),
12(FP&PW&RF), 13(PIN&FP&PW), 14(FP&RF/PIN)

【返回值】

成功返回 True，否则返回 False

【相关函数】

SSR_SetGroupTZ

5.3.9 GetUserTZs

【函数定义】

VARIANT_BOOL GetUserTZs([in] long dwMachineNumber, [in] long UserID, [out] long *TZs)

【功能】

取得用户时间段，每个用户有三个时间段，和 GetUserTZStr 不同的是返回的时间段编号格式不同而已

【参数】

dwMachineNumber

机器号

UserID

用户号

TZs

用户开门时间段。TZs 指针中共有三个值，分别存储三个时间段编号，可用 TZs[0]、TZs[1]、TZs[2] 取出

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetUserTZs、SetUserTZStr

5.3.10 SetUserTZs

【函数定义】

VARIANT_BOOL SetUserTZs([in] long dwMachineNumber, [in] long UserID, [in] long *TZs)

【功能】

设置用户时间段，每个用户最多可设置三个时间段，和 SetUserTZStr 不同的是传入的时间段编号格式不同而已

【参数】

dwMachineNumber

机器号

UserID

用户 ID

TZs

时间段编号，三个时间段都为空的时候，使用组的设置，不为空时，则指定一个用户自己的设置，该参数为 long 型指针，三个时间段编号可通过 TZs[0]、TZs[1]、TZs[2] 传入

【返回值】

成功返回 True，否则返回 False

【相关函数】

GetUserTZs、GetUserTZStr

5.3.11 GetUserTZStr

【函数定义】

VARIANT_BOOL GetUserTZStr([in] long dwMachineNumber, [in] long UserID, [out] BSTR *TZs)

【功能】

取得用户时间段，和 GetUserTZs 不同的是返回的时间段编号格式不同而已

【参数】

dwMachineNumber

机器号

UserID

用户号

TZs

用户开锁时间段。其格式为：

彩屏门禁：X1:X2:X3:X4，X1 表示是否使用组时间段，为空表示使用组时间段，为 1 时表示不使用组时间段，即使用个人时间段。X2、X3、X4 表示使用的时间段编号。例如用户 A 使用自定义的时间段 1、2，那么指纹机返回的值为是："1:1:2:"

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetUserTZStr、GetUserTZs

5.3.12 SetUserTZStr

【函数定义】

VARIANT_BOOL SetUserTZStr([in] long dwMachineNumber, [in] long UserID, [in] BSTR TZs)

【功能】

设置用户时间段，每个时间段以“:”隔开，和 SetUserTZs 不同的是传入的时间段编号格式不同而已

【参数】

dwMachineNumber

机器号

UserID

用户号

TZs

三个时间段编号，为空的时候，使用组的设置，不为空时，则指定一个用户自己的设置。具体含义可参考 GetUserTZStr

【返回值】

成功返回 True，否则返回 False

【相关函数】

GetUserTZStr、SetUserTZs

5.3.13 ACUnlock

【函数定义】

VARIANT_BOOL ACUnlock([in] long dwMachineNumber, [in] long Delay)

【功能】

开门，使开门控制器输出开门电平，并延迟（Delay/10）秒后关门

【参数】

dwMachineNumber

机器号

Delay

开门延迟时间

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.3.14 GetACFun

【函数定义】

VARIANT_BOOL GetACFun([out] long* ACFun)

【功能】

获取机器是否具有门禁功能

【参数】

ACFun

该参数接收机器的门禁功能标志，为 0 是代表没有门禁功能，15 为有门禁功能

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.3.15 GetDoorState

【函数定义】

VARIANT_BOOL GetDoorState([in] LONG MachineNumber, [out] LONG* State)

【功能】

获取当前门的状态，1：门开，0：门关

【参数】

MachineNumber

机器号

State

门的状态

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.3.16 UseGroupTimeZone

【函数定义】

VARIANT_BOOL UseGroupTimeZone(void)

【功能】

判断用户是不是使用组时间段。此功能必须和 GetUserTZs 函数或者 GetUserTZStr 一起使用其返回值才正确。即通过 GetUserTZs 函数或者 GetUserTZStr 先获取指定用户的相关时间段信息,然后使用 UseGroupTimeZone 判断该用户是否使用组时间段。

【参数】

无

【返回值】

使用组时间段返回 True, 否则返回 False

【相关函数】

GetUserTZs、GetUserTZStr

5.4 机器管理

5.4.1 IsTFTMachine

【函数定义】

VARIANT_BOOL IsTFTMachine([in] LONG dwMachineNumber)

【功能】

判断当前机器是否为彩屏机

【参数】

dwMachineNumber

机器号

【返回值】

是彩屏机返回 True,否则返回 False

【相关函数】

无

5.4.2 GetDeviceStatus

【函数定义】

VARIANT_BOOL GetDeviceStatus([in] long dwMachineNumber, [in] long dwStatus, [out] long* dwValue)

【功能】

获取机器内数据存储状态, 例如管理员数, 当前用户数等

【参数】

dwMachineNumber

机器号

dwStatus

需获取的数据, 范围为 1-22, 含义如下

- 1 管理员数量
- 2 注册用户数量
- 3 机器内指纹模板数量
- 4 密码数量
- 5 操作记录数
- 6 考勤记录数
- 7 指纹模板容量
- 8 用户容量
- 9 考勤记录容量
- 10 剩余指纹模板容量
- 11 剩余用户数容量
- 12 剩余考勤记录容量
- 21 人脸总数
- 22 人脸容量
- 其他状况返回 0

dwValue

返回指定 dwStatus 的值

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.4.3 GetDeviceInfo

【函数定义】

VARIANT_BOOL GetDeviceInfo([in] long dwMachineNumber, [in] long dwInfo, [out] long* dwValue)

【功能】

获取机器相关信息，例如语言，波特率等

【参数】

dwMachineNumber

机器号

dwInfo

需获取的信息类型，范围为 1-68（注：不能等于 65），具体含义如下

- 1 最大管理员数，总是返回 500
- 2 机器号
- 3 语言
 - 在该情况下 dwValue 接收的返回值含义为：
 - 0 语言后缀为'E'，一般代表英文
 - 1 其他状况
 - 2 语言后缀为'T'，一般代表繁体中文
 - 3 语言后缀为'L'，一般代表泰语
- 4 空闲时长(分钟)，即空闲该段时间后机器进入待机或关机
- 5 锁控时长，即锁驱动时长
- 6 考勤记录报警数，即当考勤记录达到该数量时，机器会报警以提示用户

- 7 操作记录报警数，即当操作记录达到该数量时，机器会报警以提示用户
- 8 重复记录时间，即同一用户打同一考勤状态的最小时间间隔
- 9 232/485 通讯的波特率
在该情况下，dwValue 接受的返回值含义为
- 0 1200bps
 - 1 2400
 - 2 4800
 - 3 9600
 - 4 19200
 - 5 38400
 - 6 57600
 - 其他 115200
- 10 奇偶校验，总是返回 0
- 11 停止位，总是返回 0
- 12 日期分隔符，总是返回 1
- 13 网络功能是否启用，返回值：1 为启用，0 为禁用
- 14 RS232 是否启用
- 15 RS485 是否启用
- 16 是否有声音功能
- 17 是否进行高速比对
- 18 空闲方式，即空闲时间后进入的状态，返回值：87 为关机，88 休眠
- 19 自动关机时间点，默认值为 255，即不自动关机
- 20 自动开机时间点，默认值为 255，即不自动开机
- 21 自动休眠时间点，默认值为 255，即不自动休眠
- 22 自动响铃时间点 1，不支持
- 23 1:N 比对匹配阈值
- 24 登记时匹配阈值
- 25 1:1 匹配阈值
- 26 验证时是否显示匹配分数
- 27 同时开锁人数
- 28 只验证号码卡
- 29 网络速度
在该情况下，dwValue 接受的返回值含义为
- 1 100M-H
 - 4 10M-F
 - 5 100M-F
 - 8 AUTO
 - 其他 10M-H
- 30 卡是否必需注册
- 31 机器在临时状态无操作自动返回的等待时间
- 32 机器在输入 PIN 号时无操作时自动返回的等待时间
- 33 机器在进入菜单后无操作自动返回的等待时间
- 34 时间格式
- 35 是否必需使用 1:1 比对

36—40 不支持

41—56 自动状态转换时间点 1—16，默认值都为-1，即不会自动状态转换

57 wiegand 失败 ID

58 wiegand 胁迫 ID

59 wiegand 区位码

60 wiegand 输出的脉冲宽度

61 wiegand 输出的的脉冲间隔

62 MIFARE 卡存储指纹开始扇区数

63 MIFARE 卡存储指纹扇区总数

64 MIFARE 卡存储指纹数

66 是否显示考勤状态

67-68 暂无意义

8999 此时 dwValue 同时作为输入和输出参数，输入代表需获取的其他 option 名称,输出作为该 option 的值，此时类似 GetSysOption

[注]上述时间点的返回值均为一数值，换算为时间点的方法为：将该值转换为二进制，低 8 位表示分钟数，高 8 位表示小时数，如返回值为 2860，则转换成二进制为 101100101100，低 8 位为 00101100，该值为 44，高 8 位为 00001011，该值为 11，则实际时间点为 11:44

dwValue

接收由 dwInfo 描述的值

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetDeviceInfo

5.4.4 SetDeviceInfo

【函数定义】

VARIANT_BOOL SetDeviceInfo([in] long dwMachineNumber, [in] long dwInfo, [in] long dwValue)

【功能】

设置机器相关信息，例如语言、重复记录时间等

【参数】

dwMachineNumber

机器号

dwInfo

欲设置的信息类型，范围为 1—20，其含义参考 GetDeviceInfo 函数该参数含义

dwValue

欲设置的由 dwInfo 描述的信息的值

【返回值】

成功返回 True，否则返回 False

【相关函数】

GetDeviceInfo

5.4.5 SetDeviceTime

【函数定义】

VARIANT_BOOL SetDeviceTime([in] long dwMachineNumber)

【功能】

将本地电脑的时间设置为机器时间，如需要设置指定时间，可参考 SetDeviceTime2

【参数】

dwMachineNumber

机器号

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetDeviceTime2, GetDeviceTime

5.4.6 SetDeviceTime2

【函数定义】

VARIANT_BOOL SetDeviceTime2([in] LONG dwMachineNumber, [in] LONG dwYear, [in] LONG dwMonth, [in] LONG dwDay, [in] LONG dwHour, [in] LONG dwMinute, [in] LONG dwSecond)

【功能】

设置机器时间（可指定时间）

【参数】

dwMachineNumber

机器号

dwYear, dwMonth, dwDay, dwHour, dwMinute, dwSecond

欲设置的日期时间

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetDeviceTime, GetDeviceTime

5.4.7 GetDeviceTime

【函数定义】

VARIANT_BOOL GetDeviceTime([in] long dwMachineNumber, [out] long* dwYear, [out] long* dwMonth, [out] long* dwDay, [out] long* dwHour, [out] long* dwMinute, [out] long* dwSecond)

【功能】

获取设备时间

【参数】

dwMachineNumber

机器号

dwYear, dwMonth, dwDay, dwHour, dwMinute, dwSecond

都为 long 型指针，其变量分别接受机器的日期时间

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetDeviceTime, SetDeviceTime2

5.4.8 GetSerialNumber

【函数定义】

VARIANT_BOOL GetSerialNumber([in] long dwMachineNumber, [out] BSTR* dwSerialNumber)

【功能】

获取机器序列号

【参数】

dwMachineNumber

机器号

dwSerialNumber

序列号

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.4.9 GetProductCode

【函数定义】

VARIANT_BOOL GetProductCode([in] long dwMachineNumber, [out] BSTR* lpszProductCode)

【功能】

获取机器名称

【参数】

dwMachineNumber

机器号

lpszProductCode

机器名称

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.4.10 GetFirmwareVersion

【函数定义】

VARIANT_BOOL GetFirmwareVersion([in] long dwMachineNumber, [out] BSTR* strVersion)

【功能】

获取机器固件版本

【参数】

dwMachineNumber

机器号

strVersion

固件版本

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.4.11 GetSDKVersion

【函数定义】

VARIANT_BOOL GetSDKVersion([out] BSTR* strVersion)

【功能】

获取 SDK 版本号

【参数】

strVersion

SDK 版本号

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.4.12 GetDeviceIP

【函数定义】

VARIANT_BOOL GetDeviceIP([in] long dwMachineNumber, [out] BSTR *IPAddr)

【功能】

获取机器 IP 地址

【参数】

dwMachineNumber

机器号

IPAddr

IP 地址

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetDeviceIP

5.4.13 SetDeviceIP

【函数定义】

VARIANT_BOOL SetDeviceIP([in] long dwMachineNumber, [in] BSTR IPAddr)

【功能】

设置机器 IP 地址

【参数】

dwMachineNumber

机器号

IPAddr

IP 地址

【返回值】

成功返回 True，否则返回 False

【相关函数】

GetDeviceIP

5.4.14 GetDeviceMAC

【函数定义】

VARIANT_BOOL GetDeviceMAC([in] LONG dwMachineNumber, [out] BSTR *sMAC)

【功能】

获取机器的 MAC 地址

【参数】

dwMachineNumber

机器号

sMAC

MAC 地址

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetDeviceMAC

5.4.15 SetDeviceMAC

【函数定义】

VARIANT_BOOL SetDeviceMAC([in] LONG dwMachineNumber, [in] BSTR sMAC)

【功能】

设置机器的 MAC 地址

【参数】

dwMachineNumber

机器号

sMAC

MAC 地址

【返回值】

成功返回 True，否则返回 False

【相关函数】

GetDeviceMAC

5.4.16 GetWiegandFmt

【函数定义】

VARIANT_BOOL GetWiegandFmt([in] LONG dwMachineNumber, [out] BSTR *sWiegandFmt)

【功能】

获取机器 wiegand 格式

【参数】

dwMachineNumber

机器号

sWiegandFmt

Wiegand 格式

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetWiegandFmt

5.4.17 SetWiegandFmt

【函数定义】

VARIANT_BOOL SetWiegandFmt([in] LONG dwMachineNumber, [in] BSTR sWiegandFmt)

【功能】

设置机器 wiegand 格式

【参数】

dwMachineNumber

机器号

sWiegandFmt

Wiegand 格式

【返回值】

成功返回 True，否则返回 False

【相关函数】

GetWiegandFmt

5.4.18 GetCardFun

【函数定义】

VARIANT_BOOL GetCardFun([in] LONG dwMachineNumber, [in] LONG* CardFun)

【功能】

获取机器是否支持射频卡功能

【参数】

dwMachineNumber

机器号

CardFun

当返回值为 1 时，机器仅支持射频卡，为 2 是即支持射频卡也支持指纹，为 0 不支持射频卡

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.4.19 SetDeviceCommPwd

【函数定义】

VARIANT_BOOL SetDeviceCommPwd([in] LONG dwMachineNumber, [in] LONG CommKey)

【功能】

设置机器通讯密码，该函数设置机器端通讯密码，该通讯密码会保存在机器内

【参数】

dwMachineNumber

机器号

CommKey

通讯密码

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetCommPassword

5.4.20 SetCommPassword

【函数定义】

VARIANT_BOOL SetCommPassword([in] long CommKey)

【功能】

设置 PC 端通讯密码，只有当 PC 端通讯密码和机器通讯密码相同是才可建立连接

【参数】

CommKey

通讯密码

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetDeviceCommPwd

5.4.21 QueryState

【函数定义】

VARIANT_BOOL QueryState([out] LONG *State)

【功能】

查询当前机器状态

【参数】

State

该参数返回机器当前状态，具体含义如下

- 0 等待状态
- 1 登记指纹状态
- 2 识别指纹状态
- 3 进入菜单状态
- 4 忙状态（正在处理其他工作）
- 5 等待写卡状态

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.4.22 GetVendor

【函数定义】

VARIANT_BOOL GetVendor([in] BSTR* strVendor)

【功能】

获取机器制造商名称

【参数】

strVendor

该参数返回设备制造商名称

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.4.23 GetDeviceStrInfo

【函数定义】

VARIANT_BOOL GetDeviceStrInfo([in] LONG dwMachineNumber, [in] LONG dwInfo, [out] BSTR* Value)

【功能】

获取机器的出厂时间

【参数】

dwMachineNumber

机器号

dwInfo

该参数只能被设置为 1

Value

该参数返回机器的出厂时间

【返回值】

成功返回 True，否则返回 False

【相关函数】

GetDeviceInfo

5.4.24 GetPlatform

【函数定义】

VARIANT_BOOL GetPlatform([in] LONG dwMachineNumber, [out] BSTR* Platform)

【功能】

获取设备的平台名称

【参数】

dwMachineNumber

机器号

Platform

该参数返回平台名称

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.4.25 ReadAOptions

【函数定义】

VARIANT_BOOL ReadAOptions([in] BSTR AOption, [out] BSTR* AValue)

【功能】

读取机器内指定配置参数的值，不能读带“~”开头的参数

【参数】

Aoption

参数名称

Avalue

返回的由 Aoption 描述的参数值

【返回值】

成功返回 True，否则返回 False

【相关函数】

GetSysOption

5.4.26 GetSysOption

【函数定义】

VARIANT_BOOL GetSysOption([in] LONG dwMachineNumber, [in] BSTR Option, [out] BSTR* Value)

【功能】

获取机器内的参数配置情况，注：可通过该函数获取机器使用的算法版本

【参数】

dwMachineNumber

机器号

Option

参数名称，当该参数为字符串“~ZKFPVersion”时，由 Value 描述的返回值为 10，代表当前机器使用的 10.0 指纹算法，如为空或为 9 时，代表当前机器使用的是 9.0 算法

Value

该参数返回 Option 描述的参数的值

【返回值】

成功返回 True，否则返回 False

【相关函数】

SetSysOption

5.4.27 SetSysOption

【函数定义】

VARIANT_BOOL SetSysOption([in] LONG dwMachineNumber, [in] BSTR Option, [in] BSTR Value)

【功能】

配置机器内的参数

【参数】

dwMachineNumber

机器号

Option

欲设置参数名称

Value

欲设置 Option 所描述的参数的值

【返回值】

成功返回 True，否则返回 False

【相关函数】

GetSysOption

5.5 其他

5.5.1 机器控制

5.5.1.1 ClearAdministrators

【函数定义】

VARIANT_BOOL ClearAdministrators([in] long dwMachineNumber)

【功能】

清除机器内所有管理员权限

【参数】

dwMachineNumber

机器号

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.5.1.2 EnableDevice

【函数定义】

VARIANT_BOOL EnableDevice([in] long dwMachineNumber, [in] VARIANT_BOOL bFlag)

【功能】

启用或者禁用机器，禁用即意味着关闭指纹头，键盘，卡模块等

【参数】

dwMachineNumber

机器号

bFlag

启用标志，1 为启用，0 为禁用

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.5.1.3 EnableClock

【函数定义】

VARIANT_BOOL EnableClock([in] LONG Enabled)

【功能】

启用或禁用机器时钟的“:”显示，启用时机器显示并刷新到主界面，禁用时不显示“:”

【参数】

Enabled

显示控制，1 为启用，0 为禁用

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.5.1.4 DisableDeviceWithTimeOut

【函数定义】

VARIANT_BOOL DisableDeviceWithTimeOut([in] LONG dwMachineNumber, [in] LONG TimeOutSec)

【功能】

禁用机器一段时间

【参数】

dwMachineNumber

机器号

TimeOutSec

禁用时间，该参数描述需要禁用机器的时间

【返回值】

成功返回 True，否则返回 False

【相关函数】

EnableDevice

5.5.1.5 PowerOffDevice

【函数定义】

VARIANT_BOOL PowerOffDevice([in] long dwMachineNumber)

【功能】

关机

【参数】

dwMachineNumber

机器号

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.5.1.6 RestartDevice

【函数定义】

VARIANT_BOOL RestartDevice([in] LONG dwMachineNumber)

【功能】

重启机器

【参数】

dwMachineNumber

机器号

【返回值】

成功返回 True，否则返回 False

【相关函数】

PowerOffDevice

5.5.1.7 SleepDevice

【函数定义】

VARIANT_BOOL SleepDevice([in] long dwMachineNumber)

【功能】

使机器进入休眠状态

【参数】

dwMachineNumber

机器号

【返回值】

成功返回 True，否则返回 False

【相关函数】

RestartDevice、PowerOffDevice、ResumeDevice、SuspendDevice

5.5.2 连机登记相关操作

5.5.2.1 StartEnrollEx

【函数定义】

VARIANT_BOOL StartEnrollEx([in] BSTR UserID, [in] LONG FingerID, [in] LONG Flag)

【功能】

登记用户，让机器进入登记用户状态，等待用户按指纹。注：使用该函数后，用户按三次指纹完成登记后，可能会出现按指纹不反应的情况，此时可用 **StartIdentify** 使机器进入等待状态

【参数】

UserID

需要登记的用户 ID

FingerID

需要登记的用户的指纹索引号,范围为 0—9

Flag

标示指纹模板是否有效或者是否为胁迫指纹， 0 表示指纹模板无效，1 表示指纹模板有效，3 表示为胁迫指纹

【返回值】

成功返回 True，否则返回 False

【相关函数】

CancelOperation, StartVerify, StartIdentify

5.5.2.2 StartVerify

【函数定义】

VARIANT_BOOL StartVerify([in] LONG UserID, [in] LONG FingerID)

【功能】

开始 1: 1 比对

【参数】

UserID

需要验证的用户 ID

FingerID

需要验证的用户的指纹索引号,范围为 0—9

【返回值】

成功返回 True，否则返回 False

【相关函数】

StartIdentify, CancelOperation

5.5.2.3 StartIdentify

【函数定义】

VARIANT_BOOL StartIdentify(void)

【功能】

开始 1: N 比对，使机器进入 1: N 验证状态

【参数】

无

【返回值】

成功返回 True，否则返回 False

【相关函数】

CancelOperation, StartVerify

5.5.2.4 CancelOperation

【函数定义】

VARIANT_BOOL CancelOperation(void)

【功能】

取消机器当前的指纹登记状态

【参数】

无

【返回值】

成功返回 True，否则返回 False

【相关函数】

StartEnroll, StartEnrollEx

5.5.3 卡操作相关

5.5.3.1 WriteCard

【函数定义】

VARIANT_BOOL WriteCard([in] LONG dwMachineNumber, [in] LONG dwEnrollNumber, [in] LONG dwFingerIndex1, [in] BYTE* TmpData1, [in] LONG dwFingerIndex2, [in] BYTE* TmpData2, [in] LONG dwFingerIndex3, [in] BYTE* TmpData3, [in] LONG dwFingerIndex4, [in] BYTE* TmpData4,)

【功能】

将指定用户及其指纹模板写入 MF 卡中，运行此函数后，需要在机器上出示卡

【参数】

dwMachineNumber

机器号

dwEnrollNumber

用户号

dwFingerIndex1、dwFingerIndex2、dwFingerIndex3、dwFingerIndex4

指纹索引号 (0-3)

TmpData1、TmpData2、TmpData3、TmpData4

与对应指纹索引的指纹模板，其中 TmpData1 不能为空

【返回值】

成功返回 True，否则返回 False

【相关函数】

EmptyCard

5.5.3.2 EmptyCard

【函数定义】

VARIANT_BOOL EmptyCard([in] LONG dwMachineNumber)

【功能】

清空 MF 卡

【参数】

dwMachineNumber

机器号

【返回值】

成功返回 True，否则返回 False

【相关函数】

WriteCard

5.5.4 其他

5.5.4.1 GetLastError

【函数定义】

GetLastError([out] long* dwErrorCode)

【功能】

获取最后一次错误信息

【参数】

dwErrorCode

该参数返回错误码，其含义如下：

-100	不支持或数据不存在
-10	传输的数据长度不对
-5	数据已经存在
-4	空间不足
-3	错误的大小
-2	文件读写错误
-1	SDK 未初始化，需要重新连接
0	找不到数据或重复数据
1	操作正确
4	参数错误
101	分配缓冲区错误

【返回值】

函数本身无返回值

【相关函数】

成功返回 True，否则返回 False

5.5.4.2 GetHIDEventCardNumAsStr

【函数定义】

VARIANT_BOOL GetHIDEventCardNumAsStr([out] BSTR* strHIDEventCardNum)

【功能】

获取最近一次发生刷卡事件的卡号

【参数】

strHIDEventCardNum

刷卡的卡号

【返回值】

成功返回 true, 否则返回 false

【相关函数】

OnHIDNum

5.5.4.3 CaptureImage

【函数定义】

VARIANT_BOOL CaptureImage([in] VARIANT_BOOL FullImage, [in] LONG *Width, [in] LONG *Height, [in] BYTE *Image, [in] BSTR ImageFile)

【功能】

捕获当前指纹头的指纹图像

【参数】

FullImage

是否获取整个图像，True 为整个图像，False 为只是指纹部分

Width

指定捕获的图像的宽度

Height

指定捕获图像的高度

Image

该参数接收指定的二进制格式的指纹图像、

ImageFile

该参数指定捕获的指纹图像的保存名（需包含路径）

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.5.4.4 UpdateFirmware

【函数定义】

VARIANT_BOOL UpdateFirmware([in] BSTR FirmwareFile)

【功能】

升级固件，如需要使用此功能，需先从技术人员获得相应的正确的固件

【参数】

FirmwareFile

需要升级的固件文件名（需包含路径）

【返回值】

成功返回 True，否则返回 False

【相关函数】

无

5.5.4.5 BeginBatchUpdate

【函数定义】

VARIANT_BOOL BeginBatchUpdate([in] LONG dwMachineNumber, [in] LONG UpdateFlag)

【功能】

准备以批处理模式上传数据，如在上传用户模板、用户信息等数据前使用该函数，则在上传时 SDK 将临时地把这些数据都存储在缓冲区，然后在执行 BatchUpdate 将临时数据一起传进机器

【参数】

dwMachineNumber

机器号

UpdateFlag

存在指纹覆盖标志，即上传用户指纹模板时，如该用户该指纹索引已经存在指纹，是否覆盖之前的质问模板，当该参数为 1 时，为强制覆盖，为 0 时，不覆盖

【返回值】

成功返回 True，否则返回 False

【相关函数】

BatchUpdate、CancelBatchUpdate

5.5.4.6 BatchUpdate

【函数定义】

VARIANT_BOOL BatchUpdate([in] LONG dwMachineNumber)

【功能】

开始批量上传数据，一般在使用函数 BeginBatchUpdate 后再上传完相关数据才使用该函数

【参数】

dwMachineNumber

机器号

【返回值】

成功返回 True，否则返回 False

【相关函数】

BeginBatchUpdate、CancelBatchUpdate

5.5.4.7 CancelBatchUpdate

【函数定义】

VARIANT_BOOL CancelBatchUpdate([in] LONG dwMachineNumber)

【功能】

取消批处理模式上传数据，一般在使用 BeginBatchUpdate 后，使用 BatchUpdate 之前可使用该函数，该函数释放为批处理上传准备的缓冲区

【参数】

dwMachineNumber

机器号

【返回值】

成功返回 True，否则返回 False

【相关函数】

BeginBatchUpdate 、BatchUpdate

5.5.4.8 PlayVoice

【函数定义】

VARIANT_BOOL PlayVoice([in] LONG Position, [in] LONG Length)

【功能】

播放指定的连续序号语音，具体序号视机器而定，用户可在机器内声音测试内查看到序号，一般为 0-11

【参数】

Position

开始语音序号

Length

结束语音序号

【返回值】

成功返回 True，否则返回 False

【相关函数】

PlayVoiceByIndex

5.5.4.9 PlayVoiceByIndex

【函数定义】

VARIANT_BOOL PlayVoiceByIndex([in] LONG Index)

【功能】

播放指定序号语音，具体序号视机器而定，用户可在机器内声音测试内查看到序号，一般为 0-11

【参数】

Index

需要播放的语音序号

【返回值】

成功返回 True，否则返回 False

【相关函数】

PlayVoice

6. 常见问题解答

6.1 如何下载考勤记录

首先用 ReadGeneralLogData 函数可以把所有的考勤记录读到内存中，然后使用 SSR_GetGeneralLogData 函数循环取得考勤记录，当函数 SSR_GetGeneralLogData 返回 False 的时候，说明读取考勤记录完成，把读取到的记录写到数据库或以其他形式显示，即可完成考勤记录的下载；下载管理记录和该方式一样。

6.2 如何联机创建用户

首先使用函数 SSR_SetUserInfo 给机器写入用户相关记录，如登记号、密码、姓名等信息，然后用 SSR_SetUserTmpStr/SSR_SetUserTmp 函数给该用户写入指纹模板；这种方法适合已经采集到用户信息，并以诸如数据库等存储了用户信息，不需要用户脱机再登记，提高登记的效率。

如果您需要批处理方式上传用户信息和对应指纹模板，建议您搭配以下函数使用：BeginBatchUpdata, SSR_SetUserInfo 和 BarchUpdata,EnableDevice,RefreshData,详细用法请参考 Demo 程序

6.3 U 盘数据导入及下载

在现有的脱机产品中，很多产品可以提供 U 盘数据下载；很多客户关心 U 盘数据格式，由于下载到的数据格式较复杂，故我们有把 U 盘数据导入到数据库中的工具软件，该数据库开放，客户可以从该数据库中读到下载数据，同时在 Demo 中也提供了处理 U 盘采集的数据文件（dat 文件等）以及将数据写入到指定文件数据文件的例子。所有 struct 的字节对齐方式均为 1 字节对齐。

U 盘数据主要包含了用户信息、指纹模板信息、面部模板信息、考勤记录信息、短消息等。具体的数据结构在 Demo 中均有应用。在此简单描述如下：

用户数据结构：

```
typedef struct _User_{
```

```

    U16 PIN; //用户内部编号
    U8 Privilege;
    char Password[8];
    char Name[24];
    U8 Card[4];
    U8 Group;
    U16 TimeZones[4];
    char PIN2[24];    //用户 ID 号
}

```

9.0 指纹模板数据结构:

```

typedef struct _Template_{
    U16 Size; //指纹模板长度
    U16 PIN; //用户内部编号，可和用户表中的 PIN2 对照。
    BYTE FingerID; // 指纹备份号
    BYTE Valid;
    BYTE Template[MAXTEMPLATESIZE]; //maximize template length
}                //MAXTEMPLATESIZE    602 Bytes

```

10.0 算法指纹模板 template.fp10 数据结构

```

typedef struct _Template_{
    U16 Size;                //整个结构的数据大小
    U16 PIN;                //用户 ID
    BYTE FingerID;          //指纹编号
    BYTE Valid;             //标志
    BYTE *Template; //template
}

```

考勤记录:

```

struct _AttLog_{
    U16 PIN;
    U8 PIN2[24];
    BYTE verified;
    time_t time_second;
    BYTE status;
    U32 workcode;
    BYTE reserved[4];
}

```

短消息数据结构

```

typedef struct _SMS_{
    BYTE Tag;    //类别
    U16 ID;      //数据内容标识，0 表示记录无效
    U16 ValidMinutes;    //有效分钟数，0 表示永久
}

```

```

    U16 Reserved;
    U32 StartTime;          //开始时间
    BYTE Content[MAX_SMS_CONTENT_SIZE+1]; //短消息内容
}                          // MAX_SMS_CONTENT_SIZE 160 Bytes

```

短消息和用户 Pin 之间关系的数据结构//user->sms,udata.dat

```

typedef struct _UData_{
    U16 PIN;          //0 表示无效记录
    U16 SmsID;
}GCC_PACKED TUData, *PUData; //4Bytes

```

6.4 BIOKEY 采集指纹模板写入脱机

当用 BIOKEY 采集到指纹的时候，一般在登记的时候取得指纹模板，如 OnEnroll 事件中就可以取得当前登记指纹模板，在取得指纹模板后，转换成脱机指纹模板，即可写入机器内。

6.5 取得所有用户所有信息

使用 ReadAllUserID 把所有的用户 ID 号读取到内存中，然后用 SSR_GetAllUserInfo 循环得到用户 EnrollNumber，接着用 SSR_GetUserInfo 函数就可得到用户信息。如果还需要得到指纹模板数据，可用 SSR_GetUserTmpStr 函数取得字符串类型的指纹模板。

6.6 机器连接

在连接过程中可以把机器看成一个独立的 PC，进行连接。但是要注意的是，机器的 IP 地址要和所连接的 IP 地址一一对应，在有的机器中，如 F4 有串口连接和网络连接，在不同的连接过程中要对机器进行不同的设置，需要修改通讯方式，把所带控制器开关拨为 TCP/IP 或 232/485，否则不能连接。有时候也因为串口繁忙而不能连接机器，重新启动程序即可连接；有时候因为应用软件连接机器而没有人为断开，可用 DisableDeviceWithTimeOut 函数设置机器自动断开时间；在有的连接中，由于通过串口或网络下载、修改数据等，和数据库一样的道理，为了保持数据的一致性，避免未知错误的发生，可用 EnableDevice 使机器处于工作状态下，在通讯完成后再恢复。

6.7 在 SetUserInfo 后密码不能用问题

在执行该函数后，有可能把 Password 设置为空，所以使用密码的时候就不能验证。在 SSR_SetUserInfo 之前，需要用 SSR_GetUserInfo 得到用户密码，把密码值传输给 SSR_SetUserInfo 的 Password 参数，在写入用户信息的时候可保持密码不变。

6.8 联机模板转换为脱机模板

可以使用 `FPTempConvertNew` 将所采集到的模板转换为脱机指纹模板。如何得到 BIOKEY 采集的模板，请参阅 Biokey SDK 的说明，该函数对二进制的指纹模板转换。其参数 `temp1` 和 `temp2` 都为二进制类型。也可以使用函数 `FPTempConvertNewStr` 将字符串格式的 BIOKEY 指纹模板转换为脱机指纹模板。

6.9 Demo 程序不能接机器问题

在有的时候，用户安装了考勤管理程序，用考勤管理程序是可以连接机器的，但是使用 Demo 就不能连接机器，其原因是在安装考勤管理软件的时候，动态链接库拷贝到考勤管理程序目录下，注册于安装目录，当开发时，一般情况，开发工具于系统目录下引用控件，故如果系统目录开发包和考勤软件目录下开发包版本不一致，容易造成冲突（不同版本动态连接库函数地址不同，但是写程序时 OCX 函数接口相同，所以在运行时才能表现出）。

注意,开发包在系统中注册的一般步骤为

- 1，如果系统中已经存在开发包注册（注册过开发包），请执行 `regsvr32 /u zkemkeeper.dll`，反注册旧的开发包。
- 2，拷贝所有动态连接库到系统目录下，如 win2000 位于 `winnt\system32`。
- 3，运行下执行 `regsvr32` “注册路径\zkemkeeper.dll” 注册开发包。
- 4，在开发环境下正确引用控件（请自行了解开发工具使用，这里不做介绍）。
- 5，在开发环境或运行环境下，请尽量使用同一版本开发包。

6.10 脱机指纹机连接后处于工作状态问题

在和脱机指纹机进行通讯的时候，为了保持数据的一致性，避免意外情况的发生，在连接脱机指纹机后，常用函数 `EnableDevice` 使脱机指纹机处于工作状态（请查阅 `EnableDevice` 函数说明），处于工作状态后，脱机指纹机键盘、指纹读头停止正常工作，即暂时处于不可用状态，在通讯完成后，可断开连接机器或再次使用 `EnableDevice`，使脱机指纹机恢复到正常状态。

推荐使用 `DisableDeviceWithTimeOut` 函数。

6.11 上传非英文姓名或短消息到机器内出现乱码，或不能正常显示

首先需要确定机器是否支持该语言，如当前机器为英文，上传一个阿拉伯语姓名至机器，自然不能显示，如机器已经支持相应语言，当上传相应语言的姓名至机器后，仍然不能显示，则用户在上传前，可使用相关函数将用户姓名转成 UTF-8，如在 `dephi` 中使用 `AnsiToUTF8()` 实现，再使用 `SSR_SetUserInfo` 上传，则可解决该问题

6.12 卡管理问题

如何在机器登记、取得用户卡号？

我们 SDK 有一个属性 `cardnumber`，由于开发环境的关系，有些环境可能看不到该属性，可使用函数 `GetStrCardNumber`、`SetStrCardNumber` 对其操作

对于机器上用户来说，卡号是用户信息的一部分，在设置用户信息,如 SSR_SetUserInfo 时或自动将当前属性 cardnumber 值作为卡号，设置给 SSR_SetUserInfo 所描述的用户。

登记卡流程：

设置 cardnumber 属性并上传用户信息

取用户卡号流程：

获取指定用户信息并获取 cardnumber 属性

说明：卡号在机器内部定义为无符号 4 个字节，接口属性为 Long 型，如在 VB 中不能达到无符号 4 字节，故传入卡号后三个字节也可进行验证（如果在不重复的情况下）。

6.13 连接穿越防火墙或路由器问题

在很多时候，要连接的机器要穿越防火墙或路由器等，开发包通信 Socket 也使用 UDP 协议、4370 端口。所以在防火墙设置或路由器中要开放 UDP 协议及 4370 端口。如果通过端口映射来穿越网关,可通过访问路由器等的端口+IP 访问某台机器。在一般情况下，如果开放了 UDP 及 4370,能 PING 通，即可连通。当然，下载数据时要考虑网络情况。在有的机器中，可以支持 SOAP 接口，可以通过机器内置 WebServer +SOAP 访问机器。

6.14 ZKFinger10.0 算法和 ZKFinger9.0 算法的区别以及相应模板的比较

算法性能：ZKFinger10.0 算法与 ZKFinger9.0 算法相比较，其误识率（FAR）、拒识率（FRR）、拒登率（ERR）等性能都大大提高，对过干、太湿、伤疤、脱皮等低质量的指纹图像处理效果明显增强，算法比对速度提升了 10 倍以上。

模板大小：ZKFinger10.0 算法的指纹模板约 1.6KB，ZKFinger9.0 算法的指纹模板约 512B。在使用 ZKFinger10.0 算法时,用户进行 Mifare 卡存储等操作需选择 2KB 及更大容量的 Mifare 卡。

模板兼容性：ZKFinger10.0 算法的指纹模板和 ZKFinger9.0 算法的指纹模板互不兼容。原 ZKFinger9.0 算法中已登记指纹模板的用户，选用 ZKFinger10.0 算法时需要重新登记指纹模板，反之亦然。

6.15 大容量指纹上传

大容量指纹一般指超过 1500 枚指纹，在一些设备中，最大能够容纳 8000 枚指纹或更多，在上传时，必须使用缓冲模式上传指纹模板。既批处理模式上传，在该模式下，上传速度极大提高。如何使用批处理模式上传指纹模板，请参考批处理函数说明。

6.16 高速上传和普通上传的区别

普通上传每使用一次上传函数(SSR_SetUserinfo,SSR_SetUserTmpStr 等),SDK 都会与机器进行通讯,上传相关数据到机器。

高速上传使用 **BeginBatchUpdate** 创建临时缓冲区,在该函数之后的所有上传操作都会将需要被上传的数据存放在该缓冲区,当使用函数 **BatchUpdate** 即可将缓冲区的数据一次性上传到机器,该模式大大减少了 SDK 与机器间的频繁通讯,提高上传速度,特别是对于大容量的数据,效果相当明显

6.17 如何判断机器使用的是 10.0 算法还是 9.0 算法

可使用以下函数进行判断:

VARIANT_BOOL GetSysOption([in] LONG dwMachineNumber, [in] BSTR Option, [out] BSTR* Value)

参数 Option 固定传~ZKFPVersion 字符串, 如果 Value 返回值为 10 即该机为 10.0 算法的指纹机, 如果返回 9 或者为空(因为老的彩屏机器没该值, 所以会返回为空)即该机为 9.0 算法的指纹机器。

如:

`zkem.GetSysOption(EmManth.EmMan.Dev.MachineNumber,'~ZKFPVersion',verSionFp);`

如果 verSionFp='10' ; 该机为 10.0 算法的指纹机

如果 verSionFp='9' 或者 verSionFp='' ; 为 9.0 算法的指纹机器

6.18 如何支持 ZKFinger10.0 算法模板的上传、下载、删除

ZKFinger10.0 算法提供了更高速的比对速度,但是其模板的大小及存储方式和旧版本算法有区别,表现在:

1、在使用 ZKFinger10.0 算法时, 一个手指指纹的模板在 1.6k 左右, 而旧版本一般小于 608 字节;

2、如下四个用于 10.0 模板上传下载的函数, 仅支持固件内部版本号高于 Ver6.60 的彩屏机器 (包括 Ver6.60), 函数列举如下:

以字节数组方式下载上传:

VARIANT_BOOL SetUserTmpEx([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] LONG dwFingerIndex,[in] LONG Flag, [in] BYTE* TmpData)

VARIANT_BOOL GetUserTmpEx([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] LONG dwFingerIndex,[out] LONG * Flag, [out] BYTE* TmpData, [out] LONG* TmpLength)

以字符串方式下载上传:

VARIANT_BOOL SetUserTmpExStr([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] LONG dwFingerIndex,[in] LONG Flag, [in] BSTR TmpData)

VARIANT_BOOL GetUserTmpExStr([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] LONG dwFingerIndex,[out] LONG * Flag, [out] BSTR* TmpData, [out] LONG* TmpLength)

以上四个函数除了支持普通的指纹模板 (参数 Flag 为 1) 的上传下载之外, 还支持胁迫指纹的上传下载 (参数 Flag=3)。需要注意的是, 这四个函数既支持 ZKFinger10.0 算法的指纹模板, 也支持 ZKFinger9.0 算法指纹模板。

10.0 模板的删除则主要使用彩屏机器删除指纹模板时通用的两个函数, 如下所示:

VARIANT_BOOL SSR_DelUserTmp([in] LONG dwMachineNumber, [in] BSTR EnrollNumber, [in] LONG dwFingerIndex)

VARIANT_BOOL SSR_DelUserTmpExt([in] LONG dwMachineNumber, [in] BSTR dwEnrollNumber, [in] LONG

dwFingerIndex)

说明:

Ver6.60 为机器固件的内部版本号, 可通过 SDK 函数 GetFirmwareVersion 在连接机器后获取 (或者通过考勤软件获取), 请注意内部版本号和从机器中的设备信息中查看到的固件版本号不同。

6.19 ZKFinger9.0 算法模板的上传、下载、删除

在本版本的 SDK 接口函数中, 兼容以往用于上传、下载 ZKFinger9.0 算法普通模板的函数 (SSR_GetUserTmp、SSR_GetUserTmpStr、SSR_SetUserTmp、SSR_SetUserTmpStr)。

而在本版本的 SDK 中新增了四个用于上传、下载 ZKFinger9.0 算法模板的接口函数 (SetUserTmpEx、GetUserTmpEx、SetUserTmpExStr、GetUserTmpExStr), 这四个函数除了用于普通指纹模板 (Flag=1) 的上传之外, 还可用于胁迫指纹 (Flag=3) 的上传、下载。但是这四个函数只能用于机器固件内部版本号高于 Ver6.60 (包含) 的彩屏机器。需要注意的是, 这四个函数既支持 ZKFinger10.0 算法的指纹模板, 也支持 ZKFinger9.0 算法指纹模板。

ZKFinger9.0 算法模板的删除则可以选用函数 SSR_DelUserTmp 或者 SSR_DelUserTmpExt。