ZKTeco Standalone SDK Development Manual

Version: 2.1, Rev. A.2

Date: 2018.04.25

Copyright

Release history

Date	Version	Change
2018-01-01	2.0, Rev. A.1	New release version
2018-04-25	2.1, Rev. A.2	Distribution group of International General Service Office

Contacts

No. 8, Room 2001, Chengyi North
Street, The Third Period Of Software
Park, Xiamen, Fujian, China
Tel:+86 592-7791134
For additional offices around the world, see www.zkteco.com corporate offices.

Contents

1 SDK Description

The offline communication SDK is an interface for data communication with offline fingerprint devices, access control devices, and RFID card devices. It can be used to conveniently manage user information and fingerprints, download attendance records, operation records, user information, and fingerprint templates, set devices, and configure access control. The SDK is used to:

1. Download attendance records.
2. Upload and download user information, card information, fingerprints, and face information.
3. Set access control rules of access control devices.
4. Set device time, match thresholds, etc.
5. Trigger various events of devices in real time, for example, fingerprint verification.
6. Directly enroll users online.
7. Set SMS and work code (available only on devices that support this function) of users.
8. Set personalized prompt tones, function keys, etc.

2 Quick Start

2.1 Terms

1. Real-time event

   After the SDK and the device communicate with each other successfully, some operations on the device (for example, connecting to the device, verifying a user, and enrolling a user) trigger corresponding events in real time, and data is transmitted to the PC (host computer). The triggered events are called real-time events. Users can monitor device states and user operations in real time through real-time events.

2. FP

   Shortened form of "fingerprint".

3. Fingerprint algorithm

   A fingerprint algorithm refers to the algorithm used to generate and verify fingerprint templates. At present, ZKFinger 9.0 is the latest fingerprint algorithm used by ZKSoftware black & white devices. It is a high-speed algorithm with higher performance. For details, see FAQs.

4. High-speed buffer

   A high-speed buffer refers to the memory requested by the SDK on a PC during usage. In the data upload or download process, data is first saved in the buffer before being processed.

5. Time slot, group, open door combination

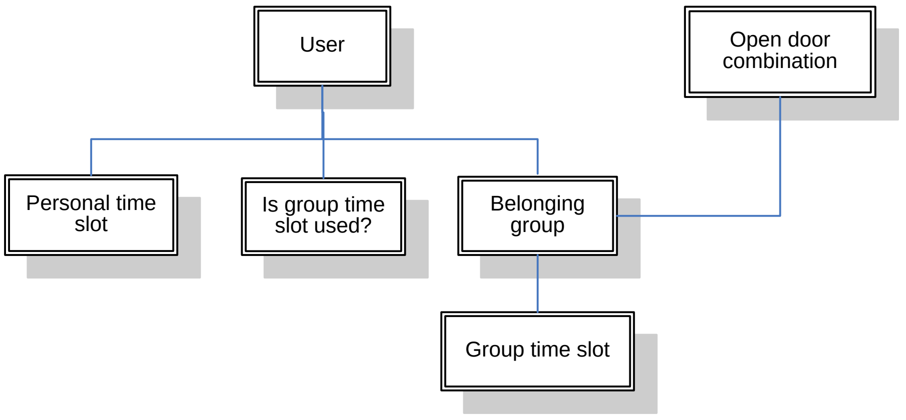
   These three terms are the most important concepts of access control.

   A time slot is a time range. A time slot includes the time information of one week, and a time range is specified for each day of this week. For example, the following expression indicates a time range from 00:00 to 22:11 in each day of one week: 00002211000022110000221100002211000022110000221100002211. Generally, 50 time slots can be set in the device.

   A group is a collection. When many users have the same access control privileges, these users can be added to the same group and use the group time slot. Then, time slots can be set for the group.

   An open door combination refers to the groups that are required for unlock. If the open door combination contains only one group, it indicates that the door is opened when any of the users in this group passes verification. If the open door combination contains two or more groups, the door is opened only after all groups pass verification. For example, an open door combination contains groups A and B, the door is opened only after a member of group A and a member of group B pass verification.

   The following figure shows the relationship of the three concepts:



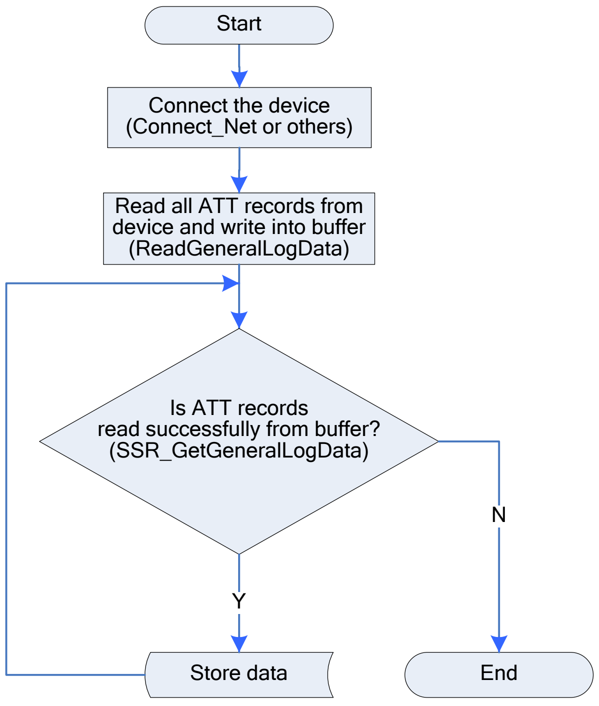
6. Operation record

   An operation record, also called management record, is a record generated when users or administrators operate on the device, for example, powering on/off the device and enrolling a user.

2.2 Common Processes

For details, see the descriptions of the demo program.

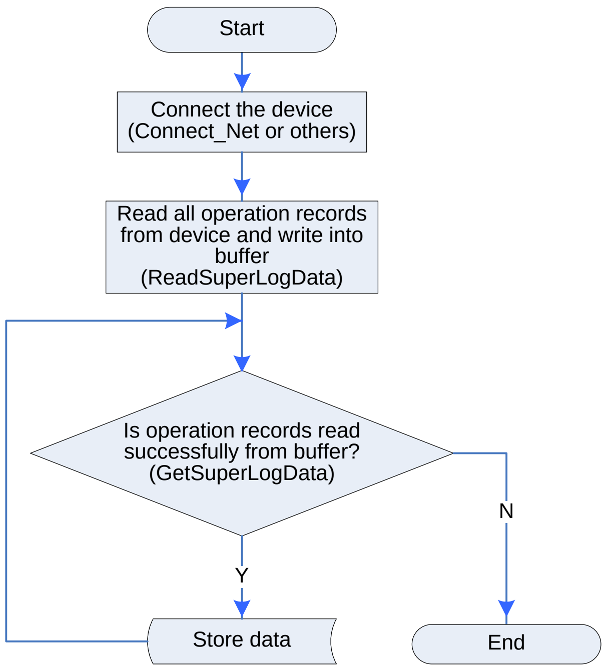
2.2.1 Downloading Attendance Records


Start
Connect the device (Connect_Net or others)
Read all ATT records from device and write into buffer (ReadGeneralLogData)
Is ATT records read successfully from buffer? (SSR_GetGeneralLogData)
N
Y
Store data
End

Note

BW device use GetGeneralLogData instead of SSR_GetGeneralLogData.

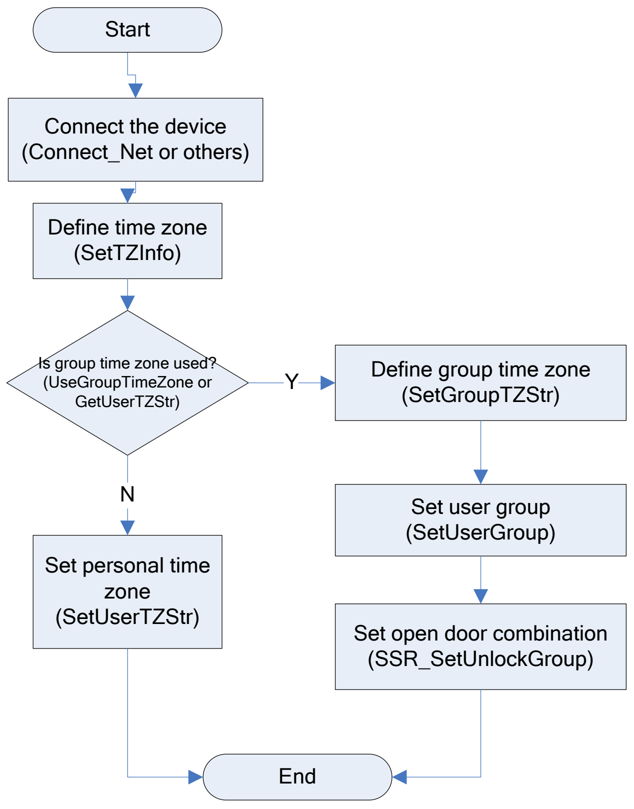
2.2.2 Downloading Operation Records


Start
Connect the device (Connect_Net or others)
Read all operation records from device and write into buffer (ReadSuperLogData)
Is operation records read successfully from buffer? (GetSuperLogData)
Y
N
Store data
End

Note

Applicable to BW, TFT and IFACE devices.

2.2.3 Setting Access Control

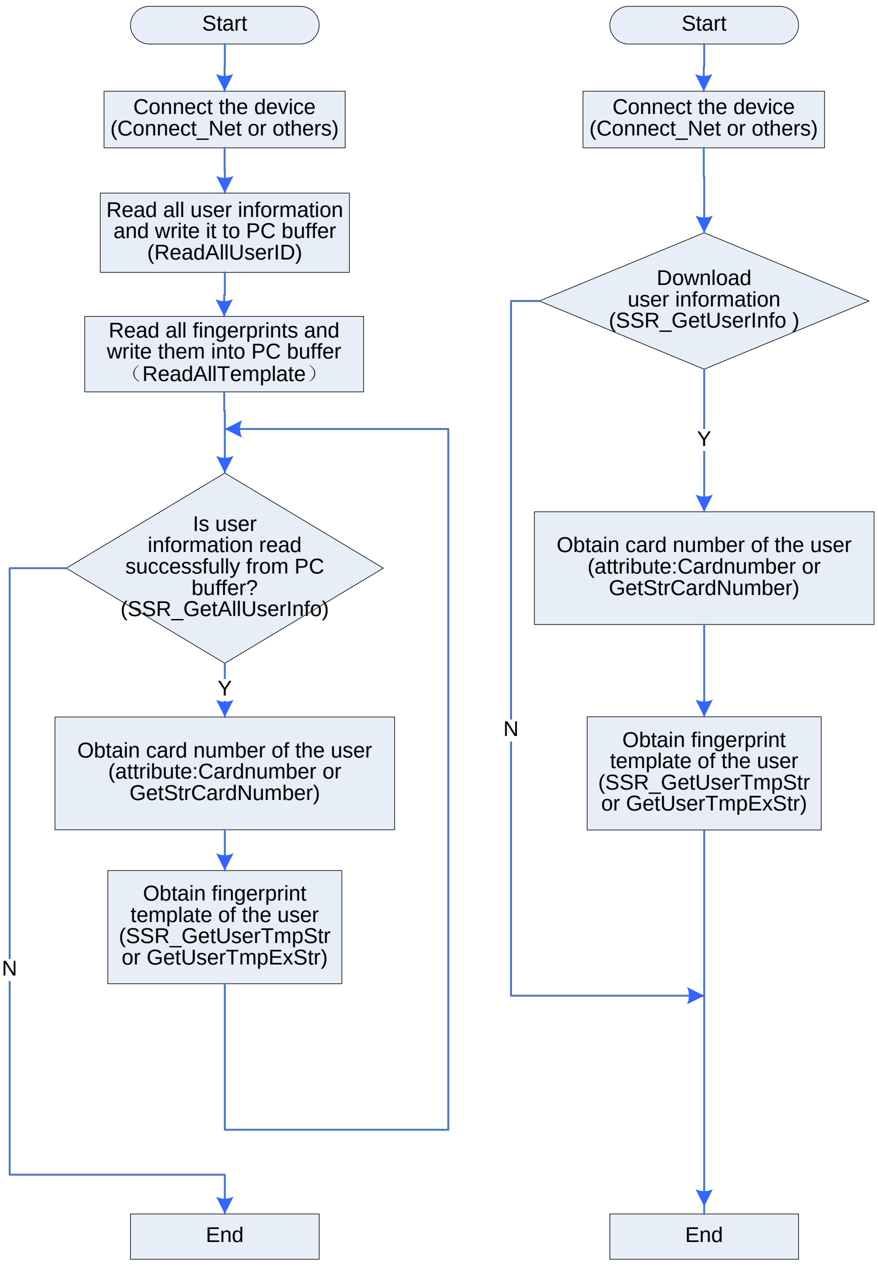

Start
Connect the device (Connect_Net or others)
Define time zone (SetTZInfo)
Is group time zone used? (UseGroupTimeZone or GetUserTZStr)
Y
Define group time zone (SetGroupTZStr)
Set user group (SetUserGroup)
Set open door combination (SSR_SetUnlockGroup)
N
Set personal time zone (SetUserTZStr)
End

Note

BW device use SetUnlockGroups instead of SSR_SetUnlockGroup.

2.2.4 Downloading User Information, or Fingerprint Templates

The left diagram shows the process of downloading the information of all users. The right diagram shows the process of downloading the information of a specified user.

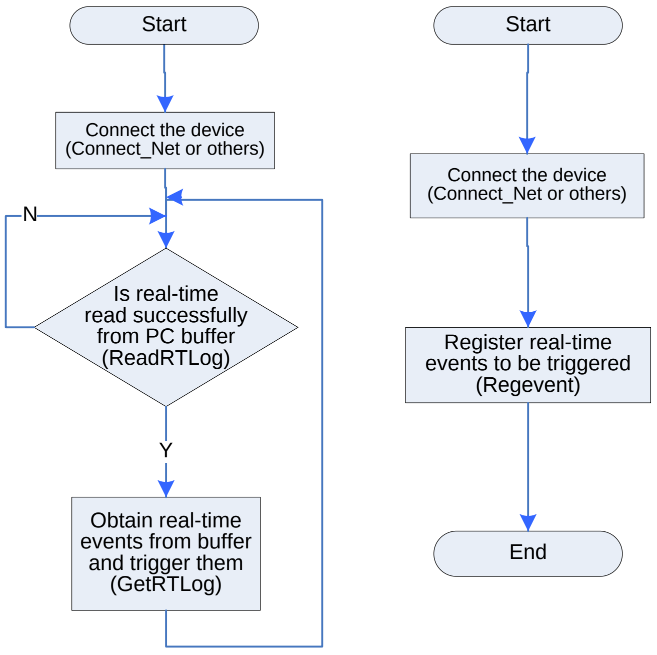

Start
Connect the device (Connect_Net or others)
Read all user information and write it to PC buffer (ReadAllUserID)
Read all fingerprints and write them into PC buffer （ReadAllTemplate）
Is user information read successfully from PC buffer? (SSR_GetAllUserInfo)
Y
N
Obtain card number of the user (attribute:Cardnumber or GetStrCardNumber)
Obtain fingerprint template of the user (SSR_GetUserTmpStr or GetUserTmpExStr)
End
Start
Connect the device (Connect_Net or others)
Download user information (SSR_GetUserInfo)
Y
N
Obtain card number of the user (attribute:Cardnumber or GetStrCardNumber)
Obtain fingerprint template of the user (SSR_GetUserTmpStr or GetUserTmpExStr)
End

Note

BW device use GetAllUserInfo instead of SSR_GetAllUserInfo, GetUserTmpStr instead of SSR_GetUserTmpStr, GetUserInfo instead of SSR_GetUserInfo.

2.2.5 Receiving Real-time Events

Real-time events can be received in two modes. The second mode is recommended.



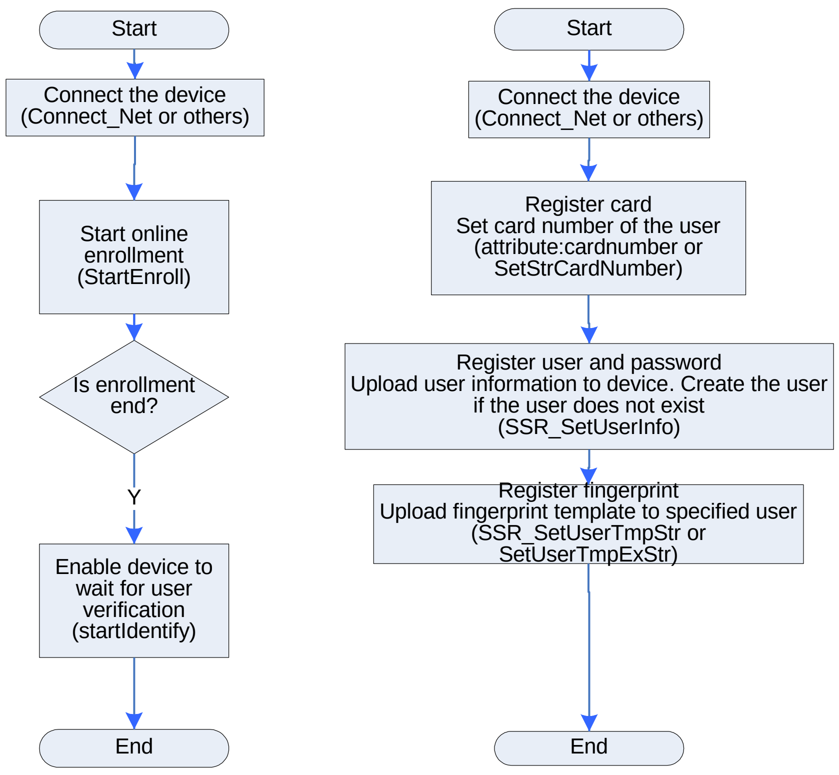
Start
Connect the device (Connect_Net or others)
N
Is real-time read successfully from PC buffer (ReadRTLog)
Y
Obtain real-time events from buffer and trigger them (GetRTLog)
Start
Connect the device (Connect_Net or others)
Register real-time events to be triggered (Regevent)
End

Note

Applicable to BW, TFT and IFACE devices

2.2.6 Enrolling Users Online (Uploading Information,and Fingerprint Templates of Users)

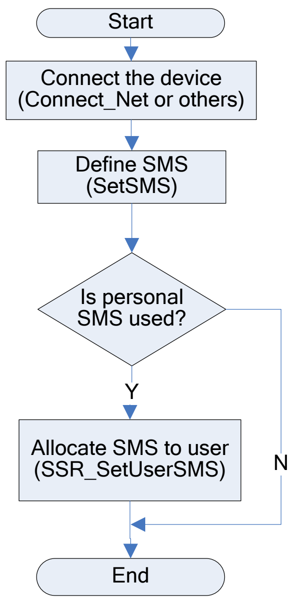
There are two online user enrollment modes. The left diagram shows the process in which the device accesses the enrollment interface to enroll a user after being connected. The right diagram shows the process of creating a user on the device and uploading the card number, password, and fingerprint information for the user (that is, enrolling a card user, a password user, and a fingerprint user).


Start
Connect the device (Connect_Net or others)
Start online enrollment (StartEnroll)
Is enrollment end?
Y
Enable device to wait for user verification (startIdentify)
End
Start
Connect the device (Connect_Net or others)
Register card
Set card number of the user (attribute:cardnumber or SetStrCardNumber)
Register user and password
Upload user information to device. Create the user if the user does not exist (SSR_SetUserInfo)
Register fingerprint
Upload fingerprint template to specified user (SSR_SetUserTmpStr or SetUserTmpExStr)
End

Note

BW device use SetUserInfo instead of SSR_SetUserInfo, SetUserTmpStr instead of SSR_SetUserTmpStr.

2.2.7 Uploading Short Messages


Start
Connect the device (Connect_Net or others)
Define SMS (SetSMS)
Is personal SMS used?
Y
N
Allocate SMS to user (SSR_SetUserSMS)
End

Note

BW device use SetUserSMS instead of SSR_SetUserSMS.

3 Related Attributes

3.1 AccGroup

To set or query the group to which a specified user belongs.

Type: LONG

See also

Attention

If this attribute is set before uploading the user, set the group to which this user belongs when invoking SetUserInfo. Otherwise, the default group 1 takes effect. This attribute is configurable.

Note

Applicable to BW, TFT and IFACE devices

3.2 AccTimeZones

To set the usage period of a user.

Type: LONG

See also

Attention

1. If this attribute is set before uploading the user, set the usage period of the user when invoking SetUserInfo.

2.This attribute is of the LONG* type. It is a LONG-type array with the subscript 3. This attribute is configurable.

Note

Applicable to BW, TFT and IFACE devices

3.3 BASE64

To set the code type.

Type: LONG

See also

Attention

If this attribute is set to True, the SDK will output a Base64 code when outputting the character string template. Otherwise, it will output a hexadecimal code.

Note

Applicable to BW, TFT and IFACE devices

3.4 CardNumber

To set or read the card number of a user.

Type: LONG

See also

Attention

If this attribute cannot be used, invoke GetStrCardnumber and SetStrCardnumber. This attribute is configurable.

Note

Applicable to BW, TFT and IFACE devices

3.5 CommPort

To set a serial port number, or a port number used for RS485 connection.

Type: LONG

See also

Attention

The attribute is of the LONG type and is configurable.

Note

Applicable to BW, TFT and IFACE devices

3.6 ConvertBIG5

To convert simplified Chinese characters into traditional Chinese characters.

Type: LONG

See also

Attention

1. If the value of this attribute is set to True, the SDK will automatically convert simplified Chinese characters into traditional Chinese characters for further development. This function is invalid on a multi-language machine. Therefore, do not set this attribute.

2.This attribute is invalid on a multi-language machine and later versions. Therefore, do not set this attribute. You do not need to modify this attribute in versions later than ZEM100 5.22 and ZEM200 5.30.

Note

Applicable to BW, TFT and IFACE devices

3.7 PINWidth

To label the maximum length of the user code, which is an Arabic numeral.

Type: LONG

See also

Attention

The attribute is of the LONG type and is read-only.

Note

Applicable to BW, TFT and IFACE devices

3.8 GetStrCardNumber

VARIANT_BOOL GetStrCardNumber(BSTR* ACardNumber)

To query the value of the SDK attribute cardnumber. You can invoke this function to query the card number of a user after obtaining the information about this user.

Parameters

Parameter description:

name	type	param direction	description of param
ACardNumber	BSTR*	[out]	Card number

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW

3.9 SetStrCardNumber

VARIANT_BOOL SetStrCardNumber(BSTR ACardNumber)

To set the value of the SDK attribute cardnumber. Before setting user information, you are advised to invoke this function to set the card number of the user.

Parameters

Parameter description:

name	type	param direction	description of param
ACardNumber	BSTR	[in]	Card number

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW

3.10 IsNewFirmwareMachine

VARIANT_BOOL IsNewFirmwareMachine(LONG dwMachineNumber)

To identify whether the current device firmware are new architecture firmware.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Apply to BW, TFT, IFACE devices

3.11 GetDeviceFirmwareVersion

VARIANT_BOOL GetDeviceFirmwareVersion(LONG dwMachineNumber, BSTR* strVersion)

To query the firmware version.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
strVersion	BSTR*	[out]	Firmware version of the machine

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

This interface is applicable to the new architecture firmware.

4 Real-time Event Functions

4.1 Obtaining Real-Time Events

4. 1. 1 RegEvent

VARIANT_BOOL RegEvent(LONG dwMachineNumber, LONG EventMask)

To register a real-time event to be triggered.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
EventMask	LONG	[in]	Event ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The event IDs are described as follows:

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

To register multiple real-time events, perform the OR operation for the binary event IDs. To register all real-time events, set EventMask to 65535.

Note

Applicable to BW, TFT and IFACE devices

4.1.2 ReadRTLog

VARIANT_BOOL ReadRTLog(LONG dwMachineNumber)

To read real-time events to the buffer of the PC.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

Attention

This function can be used together with GetRTLog to actively obtain real-time events from the machine after the machine is connected successfully.

Note

Applicable to BW, TFT and IFACE devices

4.1.3 GetRTLog

VARIANT_BOOL GetRTLog(LONG dwMachineNumber)

To get a real-time event from the buffer of the PC and trigger this event.

Parameters

Parameter description:

name	type	param direction	description of param

dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

This function can be used together with ReadRTLog to actively obtain real-time events from the machine after the machine is connected successfully.

Note

Applicable to BW, TFT and IFACE devices

4.2 Real-Time Events

4.2.1 OnConnected

OnConnected(LONG MachineNumber)

To trigger this event when the machine is connected successfully.

Parameters

Parameter description:

name	type	param direction	description of param
MachineNumber	LONG	[in]	Machine ID

Returns

None

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

4. 2. 2 OnDisConnected

OnDisConnected(LONG MachineNumber)

To trigger this event when the machine is disconnected.

Parameters

Parameter description:

name	type	param direction	description of param
MachineNumber	LONG	[in]	Machine ID

Returns

None

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

4. 2. 3 OnAlarm

OnAlarm (LONG AlarmType, LONG EnrollNumber, LONG Verified)

To trigger this event when the machine raises an alarm.

Parameters

Parameter description:

name	type	param direction	description of param
AlarmType	LONG	[in]	Alarm type
EnrollNumber	LONG	[in]	User ID
Verified	LONG	[in]	Whether to perform verification

Returns

None

See also

Attention

1. Alarm Type: alarm type. The value 55 indicates a tamper alarm, 58 misoperation alarm, 32 duress alarm, and 34 passback alarm.

2.EnrollNumber: user ID. The value is 0 if the alarm is a tamper alarm, misoperation alarm, or duress key alarm. The value is the user ID if the alarm is another type of duress alarm or a passback alarm.

3.Verified: whether to perform verification. The value is 0 when the alarm is a tamper or misoperation alarm, and is 1 if the alarm is of other types.

Note

Applicable to BW, TFT and IFACE devices

4.2.4 OnDoor

OnDoor (LONG EventType)

To trigger this event when the machine opens the door.

Parameters

Parameter description:

name	type	param direction	description of param
EventType	LONG	[in]	Door opening type

Returns

None

See also

Attention

EventType: specifies the door opening type. The value 4 indicates that the door is not closed properly or has been opened, 53 indicates an exit button, 5 indicates that the door has been closed, and 1 indicates that the door is opened unexpectedly.

Note

Applicable to BW, TFT and IFACE devices

4.2.5 OnAttTransaction

OnAttTransaction(LONG EnrollNumber, LONG IsInValid, LONG AttState, LONG VerifyMethod, LONG Year, LONG Month, LONG Day, LONG Hour, LONG Minute, LONG Second, LONG WorkCode)

To trigger this event when the verification is passed.

Parameters

Parameter description:

name	type	param direction	description of param

EnrollNumber	LONG	[in]	Alarm type
IsInValid	LONG	[in]	Whether the record is valid
AttState	LONG	[in]	Attendance status
VerifyMode	LONG	[in]	Verification mode of the record
Year	LONG	[in]	Year
Month	LONG	[in]	Month
Day	LONG	[in]	Date
Hour	LONG	[in]	Hour
Minute	LONG	[in]	Minute

Returns

None

See also

Attention

1. The VerifyMode parameter specifies the verification mode. The values are described as follows:
Under normal conditions:
0 indicates password verification, 1 fingerprint verification, and 2 card verification.
Under multiple verification modes:
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

2.The AttState parameter specifies the attendance status. The values are described as follows:
0-Check-In Default
1-Check-Out

2-Break-Out
3-Break-In
4-OT-In
5-OT-Out

Note

Applicable to BW

4. 2. 6 OnAttTransactionEx

OnAttTransactionEx(BSTR EnrollNumber, LONG IsInValid, LONG AttState, LONG VerifyMethod, LONG Year, LONG Month, LONG Day, LONG Hour, LONG Minute, LONG Second, LONG WorkCode)

To trigger this event when the verification is passed.

Parameters

Parameter description:

name	type	param direction	description of param
EnrollNumber	BSTR	[in]	Alarm type
IsInValid	LONG	[in]	Whether the record is valid
AttState	LONG	[in]	Attendance status
VerifyMode	LONG	[in]	Verification mode of the record
Year	LONG	[in]	Year
Month	LONG	[in]	Month
Day	LONG	[in]	Date
Hour	LONG	[in]	Hour
Minute	LONG	[in]	Minute
Second	LONG	[in]	Second
WorkCode	LONG	[in]	Work code

Returns

None

See also

Attention

1. The VerifyMode parameter specifies the verification mode. The values are described as follows:

Under normal conditions:

0 indicates password verification, 1 fingerprint verification, and 2 card verification.

Under multiple verification modes:

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

2.The AttState parameter specifies the attendance status. The values are described as follows:

0-Check-In Default

1-Check-Out

2-Break-Out

3-Break-In

4-OT-In

5-OT-Out

The WorkCode parameter specifies the work code. If the machine does not support the work code, 0 is returned.

Note

Applicable to BW, TFT and IFACE devices

4. 2. 7 OnEnrollFinger

OnEnrollFinger(LONG EnrollNumber, LONG FingerIndex, LONG ActionResult, LONG TemplateLength)

To trigger this event when registering a fingerprint.

Parameters

Parameter description:

name	type	param direction	description of param

EnrollNumber	LONG	[in]	ID of the user whose fingerprint is to be registered
FingerIndex	LONG	[in]	Index of the fingerprint to be registered
ActionResult	LONG	[in]	Operation result. The value is 0 if the operation succeeds and is larger than 0 if the operation fails.
TemplateLength	LONG	[in]	Length of the fingerprint template

Returns

None

See also

Attention

Use under certain circumstances of 9-digit job width and 5-digit job width

Note

Applicable to BW,IFACE devices

4.2.8 OnEnrollFingerEx

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

To trigger this event when registering a fingerprint.

Parameters

Parameter description:

name	type	param direction	description of param
EnrollNumber	BSTR	[in]	ID of the user whose fingerprint is to be registered
FingerIndex	LONG	[in]	Index of the fingerprint to be registered
ActionResult	LONG	[in]	Operation result. The value is 0 if the operation succeeds and is larger than 0 if the operation fails.
TemplateLength	LONG	[in]	Length of the fingerprint template

Returns

None

See also

Attention

Note

Applicable to TFT and IFACE devices

4.2.9 OnDeleteTemplate

OnDeleteTemplate(LONG EnrollNumber, LONG FingerIndex)

To trigger this event when deleting a fingerprint template on the machine.

Parameters

Parameter description:

name	type	param direction	description of param
EnrollNumber	LONG	[in]	ID of the user to which the fingerprint template to be deleted belongs
FingerIndex	LONG	[in]	Index of the fingerprint template to be deleted

Returns

None

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

4.2.10 OnFinger

OnFinger()

To trigger this message when a fingerprint is scanned by the machine.

Parameters

None

Returns

None

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

4. 2. 11 OnFingerFeature

OnFingerFeature(LONG Score)

To trigger this message if a finger is pressed onto the fingerprint reader when registering user fingerprints.

Parameters

Parameter description:

name	type	param direction	description of param
Score	LONG	[in]	Quality score of the fingerprint

Returns

None

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

4. 2. 12 OnHIDNum

OnHIDNum (LONG CardNumber)

To trigger this message when punching a card.

Parameters

Parameter description:

name	type	param direction	description of param
CardNumber	LONG	[in]	Card number

Returns

None

See also

Attention

The card can be an ID card or HID card. For an MIFARE card, this event will be triggered only when it is used as an ID card.

Note

Applicable to BW, TFT and IFACE devices

4.2.13 OnKeyPress

OnKeyPress(LONG Key)

To trigger this message when a key is available.

Parameters

Parameter description:

name	type	param direction	description of param
Key	LONG	[in]	Key value

Returns

None

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

4.2.14 OnNewUser

OnNewUser (LONG EnrollNumber)

To trigger this message when a new user is registered successfully.

Parameters

Parameter description:

name	type	param direction	description of param
EnrollNumber	LONG	[in]	ID of the newly registered user

Returns

None

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

4.2.15 OnVerify

OnVerify (LONG UserID)

To trigger this message during user verification.

Parameters

Parameter description:

name	type	param direction	description of param
UserID	LONG	[in]	ID of the user to be verified

Returns

None

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

4. 2. 16 OnWriteCard

OnWriteCard (LONG EnrollNumber, LONG ActionResult, LONG Length)

To trigger this event when the machine writes data to a card.

Parameters

Parameter description:

name	type	param direction	description of param
EnrollNumber	LONG	[in]	User ID of the card to which data is to be written
ActionResult	LONG	[in]	Result of the write operation. The value 0 indicates operation success and other values indicate operation failure.
Length	LONG	[in]	Total size of the data written to the card

Returns

None

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

4. 2. 17 OnEmptyCard

OnEmptyCard (LONG ActionResult)

To trigger this event when the machine writes data to a card.

Parameters

Parameter description:

name	type	param direction	description of param
ActionResult	LONG	[in]	Result of the clear operation. The value 0 indicates operation success and other values indicate operation failure.

Returns

None

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

4. 2. 18 OnEMData

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

To trigger this event when the machine sends an unknown event to the SDK.

Parameters

Parameter description:

name	type	param direction	description of param
DataType	LONG	[in]	Event type
DataLen	LONG	[in]	Total data length
DataBuffer	CHAR*	[in]	Data

Returns

None

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5 Common Functions

5.1 Device Connection Functions

5.1.1 Connect_Net

VARIANT_BOOL Connect_Net(BSTR IPAdd, LONG Portl)

To connect to the machine to set up a network connection with the machine by using an IP address.

Parameters

Parameter description:

name	type	param direction	description of param
IPAdd	BSTR	[in]	IP address of the machine
Portl	LONG	[in]	Port number of the machine

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The default port number for connecting to the machine is 4370.

Note

Applicable to BW, TFT and IFACE devices

5.1.2 Connect_Com

VARIANT_BOOL Connect_Com(LONG ComPort, LONG MachineNumber, LONG BaudRate)

To connect to the machine through a serial port, that is, the RS232 RS485 port.

Parameters

Parameter description:

name	type	param direction	description of param
ComPort	LONG	[in]	Serial port of the PC to be connected to the machine
MachineNumber	LONG	[in]	Machine ID
BaudRate	LONG	[in]	Baud rate

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

This function is also used when a machine communicates with a PC by using a USB client. The precondition is that the USB client drive is installed beforehand to simulate a virtual serial port, which can be viewed in the device manager on the PC. The program can also search for this serial port. For details, see the description of USBClient in DEMO.

Note

Applicable to BW, TFT and IFACE devices

5.1.3 Connect_USB

VARIANT_BOOL Connect_USB(LONG MachineNumber)

To connect to the machine through a USB port.

Parameters

Parameter description:

name	type	param direction	description of param
MachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Connect_Com

Attention

This function applies only to H series machines and cannot be used in the communication conducted through a USB client. For details about the communication conducted through a USB client, see the description of Connect_Com.

Note

Applicable to BW

5. 1. 4 Connect_P4P

VARIANT_BOOL Connect_P4P(BSTR uid)

To connect P2P devices.

Parameters

Parameter description:

name	type	param direction	description of param
uid	BSTR	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

uid is the factory number, used as the identify ID of the P2P devices

Note

Applicable to some P2P devices, such as the K Pro series attendace machine

5.1.5 Disconnect

Disconnect()

To disconnect from the machine to release relevant resources.

Parameters

None

Returns

None

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.2 Data Management Functions

5.2.1 Attendance Record Data

5.2.1.1 ReadGeneralLogData

VARIANT_BOOL ReadGeneralLogData(LONG dwMachineNumber)

To read attendance records to the internal buffer of the PC. The function is the same as ReadAllGLogData.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success

False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.2.1.2 ReadAllGLogData

VARIANT_BOOL ReadAllGLogData(LONG dwMachineNumber)

To read attendance records to the internal buffer of the PC. The function is the same as ReadGeneralLogData.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.2.1.3 ReadTimeGLogData

VARIANT_BOOL ReadTimeGLogData(LONG dwMachineNumber, BSTR sTime, BSTR eTime)

To download attendance records based on the specified start time and end time, accurate to seconds.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
sTime	BSTR	[in]	Start time in the format of YYYY-MM-DD hh:mm:ss
eTime	BSTR	[in]	End time in the format of YYYY-MM-DD hh:mm:ss

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

This interface is applicable to the new architecture firmware.

5.2.1.4 ReadNewGLogData

VARIANT_BOOL ReadNewGLogData(LONG dwMachineNumber)

To download the new generated attendance records.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value

True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

This interface is applicable to the new architecture firmware.

5.2.1.5 GetGeneralLogData

VARIANT_BOOL GetGeneralLogData(LONG dwMachineNumber, LONG* dwTMachineNumber, LONG* dwEnrollNumber, LONG* dwEMachineNumber, LONG* dwVerifyMode, LONG* dwInOutMode, LONG* dwYear, LONG* dwMonth, LONG* dwDay, LONG* dwHour, LONG* dwMinute)

To read attendance records from the internal buffer one by one. Before using this function, execute ReadAllGLogData or ReadGeneralLogData to read the attendance records from the machine to the internal buffer of the PC. Each time this function is executed, the pointer moves to the next attendance record. This function is the same as GetAllGLogData. They differ only in the interface name for compatibility.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwTMachineNumber	LONG*	[out]	Pointer that points to the LONG variable. Its value is the machine ID of an attendance record.
dwEnrollNumber	LONG*	[out]	Pointer that points to the LONG variable. Its value is the user ID of an attendance record.
dwEMachineNumber	LONG*	[out]	Pointer that points

			to the LONG variable. Its value is the machine ID of an attendance record.
dwVerifyMode	LONG*	[out]	Pointer that points to the LONG variable. Its value is the verification mode of an attendance record.
dwInOutMode	LONG*	[out]	Pointer that points to the LONG variable. Its value is the attendance status of an attendance record.
dwYear	LONG*	[out]	Pointer that points to the LONG variable. Its value is the year of an attendance record.
dwMonth	LONG*	[out]	Pointer that points to the LONG variable. Its value is the month of an attendance record.
dwDay	LONG*	[out]	Pointer that points to the LONG variable. Its value is the day of an attendance record.
dwHour	LONG*	[out]	Pointer that points to the LONG variable. Its value is the hour of an attendance record.
dwMinute	LONG*	[out]	Pointer that points

			to the LONG variable. Its value is the minute of an attendance record.

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The dwVerifyMode parameter specifies the verification mode. The values are described as follows:
Generally:
0 indicates password verification, 1 indicates fingerprint verification and 2 indicates card verification.
Under multiple verification modes:
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

2. The dwInOutMode parameter specifies the attendance status. The values are described as follows:
0-Check-In Default
1-Check-Out
2-Break-Out
3-Break-In

4-OT-In
5-OT-Out

Note

Applicable to BW

5.2.1.6 GetAllGLogData

VARIANT_BOOL GetAllGLogData(LONGdwMachineNumber , LONG* dwTMachineNumber, LONG* dwEnrollNumber,LONG* dwEMachineNumber, LONG* dwVerifyMode, LONG* dwInOutMode, LONG*dwYear, LONG* dwMonth,LONG* dwDay, LONG* dwHour,LONG* dwMinute)

To read attendance records from the internal buffer one by one. Before using this function, execute ReadAllGLogData or ReadGeneralLogData to read the attendance records from the machine to the internal buffer of the PC. Each time this function is executed, the pointer moves to the next attendance record. This function is the same as GetGeneralLogData. They differ only in the interface name for compatibility.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwTMachineNumber	LONG*	[out]	Pointer that points to the LONG variable. Its value is the machine ID of an attendance record.
dwEnrollNumber	LONG*	[out]	Pointer that points to the LONG variable. Its value is the user ID of an attendance record.
dwEMachineNumber	LONG*	[out]	Pointer that points to the LONG variable. Its value is the machine ID of an attendance record.
dwVerifyMode	LONG*	[out]	Pointer that points to the LONG variable. Its value is the

			verification mode of an attendance record.
dwInOutMode	LONG*	[out]	Pointer that points to the LONG variable. Its value is the attendance status of an attendance record.
dwYear	LONG*	[out]	Pointer that points to the LONG variable. Its value is the year of an attendance record.
dwMonth	LONG*	[out]	Pointer that points to the LONG variable. Its value is the month of an attendance record.
dwDay	LONG*	[out]	Pointer that points to the LONG variable. Its value is the day of an attendance record.
dwHour	LONG*	[out]	Pointer that points to the LONG variable. Its value is the hour of an attendance record.
dwMinute	LONG*	[out]	Pointer that points to the LONG variable. Its value is the minute of an attendance record.

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

GetGeneralLogData

Attention

1. The dwVerifyMode parameter specifies the verification mode. The meanings of the values are the same as those of the GetGeneralLogData parameter.

2. The dwInOutMode parameter specifies the attendance status. The meanings of the values are the same as those of the GetGeneralLogData parameter.

Note

Applicable to BW

5.2.1.7 GetGeneralLogDataStr

VARIANT_BOOL GetGeneralLogDataStr(LONG dwMachineNumber, LONG* dwEnrollNumber, LONG* dwVerifyMode, LONG* dwInOutMode, BSTR* TimeStr)

To read attendance records from the internal buffer one by one. Before using this function, execute ReadAllGLogData or ReadGeneralLogData to read the attendance records from the machine to the internal buffer of the PC. Each time this function is executed, the pointer moves to the next attendance record. This function is the same as GetGeneralLogData. They differ in the format of time in the returned values.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG*	[out]	Pointer that points to the LONG variable. Its value is the user ID of an attendance record.
dwVerifyMode	LONG*	[out]	Pointer that points to the LONG variable. Its value is the verification mode of an attendance record. The meanings of the values are the same as those of the GetGeneralLogData parameter.
dwInOutMode	LONG*	[out]	Pointer that points to the LONG variable. Its value is the attendance status of an attendance record. The meanings of the values are the same as those of the GetGeneralLogData

			parameter.
TimeStr	BSTR*	[out]	Pointer that points to the LONG variable. Its value is the attendance time of an attendance record.

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

GetGeneralLogData

Attention

1. The dwVerifyMode parameter specifies the verification mode. The meanings of the values are the same as those of the GetGeneralLogData parameter.

2. The dwInOutMode parameter specifies the attendance status. The meanings of the values are the same as those of the GetGeneralLogData parameter.

Note

Applicable to BW

5.2.1.8 GetGeneralExtLogData

VARIANT_BOOL GetGeneralExtLogData(LONG dwMachineNumber, LONG* dwEnrollNumber, LONG* dwVerifyMode, LONG* dwInOutMode, LONG* dwYear, LONG* dwMonth, LONG* dwDay, LONG* dwHour, LONG* dwMinute, LONG* dwSecond, LONG* dwWorkCode, LONG* dwReserved)

To read attendance records from the internal buffer one by one. Before using this function, execute ReadAllGLogData or ReadGeneralLogData to read the attendance records from the machine to the internal buffer of the PC. Each time this function is executed, the pointer moves to the next attendance record. This function is an enhanced version of GetGeneralLogData. They are compatible.

Parameters

Parameter description:

name	type	param direction	description of param

dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG*	[out]	Pointer that points to the LONG variable. Its value is the user ID of an attendance record.
dwVerifyMode	LONG*	[out]	Pointer that points to the LONG variable. Its value is the verification mode of an attendance record.
dwInOutMode	LONG*	[out]	Pointer that points to the LONG variable. Its value is the attendance status of an attendance record.
dwYear	LONG*	[out]	Pointer that points to the LONG variable. Its value is the year of an attendance record.
dwMonth	LONG*	[out]	Pointer that points to the LONG variable. Its value is the month of an attendance record.
dwDay	LONG*	[out]	Pointer that points to the LONG variable. Its value is the day of an attendance record.
dwHour	LONG*	[out]	Pointer that points to the LONG variable. Its value is the hour of an attendance record.
dwMinute	LONG*	[out]	Pointer that points to the LONG variable. Its value is the minute of an attendance record.
dwSecond	LONG*	[out]	Pointer that points to

			the LONG variable. Its value is the second of an attendance record.
dwWorkCode	LONG*	[out]	Pointer that points to the LONG variable. Its value is the work code of an attendance record.
dwReserved	LONG*	[out]	Reserved parameter. It is meaningless.

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

GetGeneralLogData

Attention

1. The dwVerifyMode parameter specifies the verification mode. The meanings of the values are the same as those of the GetGeneralLogData parameter.

2. The dwInOutMode parameter specifies the attendance status. The meanings of the values are the same as those of the GetGeneralLogData parameter.

Note

Applicable to BW

5.2.1.9 SSR_GetGeneralLogData

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

To read attendance records from the internal buffer one by one. Before using this function, execute ReadAllGLogData or ReadGeneralLogData to read the attendance records from the machine to the internal buffer of the PC. Each time this function is executed, the pointer moves to the next attendance record. This function is the same as GetGeneralLogData. The difference is that this function applies to color-screen machines.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR*	[out]	Pointer that points to the BSTR variable. Its value is the user ID of an attendance record. A user ID contains a maximum of 24 digits.
dwVerifyMode	LONG*	[out]	Pointer that points to the LONG variable. Its value is the verification mode of an attendance record.
dwInOutMode	LONG*	[out]	Pointer that points to the LONG variable. Its value is the attendance status of an attendance record.
dwYear	LONG*	[out]	Pointer that points to the LONG variable. Its value is the year of an attendance record.
dwMonth	LONG*	[out]	Pointer that points to the LONG variable. Its value is the month of an attendance record.
dwDay	LONG*	[out]	Pointer that points to the LONG variable. Its value is the day of an attendance record.
dwHour	LONG*	[out]	Pointer that points to the LONG variable. Its value is the hour of an attendance record.
dwMinute	LONG*	[out]	Pointer that points to the LONG variable. Its value is the minute of an attendance record.
dwSecond	LONG*	[out]	Pointer that points to the LONG variable. Its value is the second of an attendance record.
dwWorkcode	LONG*	[out]	Pointer that points to the LONG variable. Its value is the work code of an attendance record.

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The dwVerifyMode parameter specifies the verification mode. The values are described as follows:

Generally:

0 indicates password verification, 1 indicates fingerprint verification and 2 indicates card verification.

Under multiple verification modes:

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

2. The dwInOutMode parameter specifies the attendance status. The values are described as follows:

0-Check-In Default

1-Check-Out

2-Break-Out

3-Break-In

4-OT-In

5-OT-Out

Note

Applicable to TFT and IFACE devices

5.2.1.10 ClearGLog

VARIANT_BOOL ClearGLog(long dwMachineNumber)

To clear all attendance records on the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.2.1.11 DeleteAttlogBetweenTheDate

VARIANT_BOOL DeleteAttlogBetweenTheDate(LONG dwMachineNumber, BSTR sTime, BSTR eTime)

To delete attendance records based on the specified start time and end time, accurate to seconds.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
sTime	BSTR	[in]	Start time in the format of YYYY-MM-DD hh:mm:ss
eTime	BSTR	[in]	End time in the format of YYYY-MM-DD hh:mm:ss

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

This interface is applicable to the new architecture firmware.

5.2.1.12 DeleteAttlogByTime

VARIANT_BOOL DeleteAttlogByTime(LONG dwMachineNumber, BSTR sTime)

To delete all attendance records generated before the specified time point, accurate to seconds.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
sTime	BSTR	[in]	Start time point in the format of YYYY-MM-DD hh:mm:ss

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

This interface is applicable to the new architecture firmware.

5. 2. 2 Operation Record Data

5.2.2.1 ReadSuperLogData

VARIANT_BOOL ReadSuperLogData(long dwMachineNumber)

To read operation records to the internal buffer of the PC. The function is the same as ReadAllSLogData.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.2.2.2 ReadAllSLogData

VARIANT_BOOL ReadAllSLogData(long dwMachineNumber)

To read operation records to the internal buffer of the PC. The function is the same as ReadSuperLogData.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.2.2.3 GetSuperLogData

VARIANT_BOOL GetSuperLogData(LONG dwMachineNumber, LONG* dwTMachineNumber, LONG* dwSEnrollNumber, LONG* Params4, LONG* Params1, LONG* Params2, LONG* dwManipulation, LONG* Params3, LONG* dwYear, LONG* dwMonth, LONG* dwDay, LONG* dwHour, LONG* dwMinute)

To read operation records from the internal buffer one by one. Before using this function, execute ReadAllSLogData or ReadSuperLogData to read the operation records from the machine to the internal buffer of the PC. Each time this function is executed, the pointer moves to the next operation record. This function differs from GetSuperLogData2 in that the GetSuperLogData2 function can obtain the operation record time accurate to seconds.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwTMachineNumber	LONG*	[out]	Pointer that points to the LONG variable. Its value is the machine ID of an operation record.

dwSEnrollNumber	LONG*	[out]	Pointer that points to the LONG variable. Its value is the administrator ID of an operation record.
Params4	LONG*	[out]	Pointer that points to the LONG variable. The value varies according to that of dwManipulation.
Params1	LONG*	[out]	Pointer that points to the LONG variable. The value varies according to that of dwManipulation.
Params2	LONG*	[out]	Pointer that points to the LONG variable. The value varies according to that of dwManipulation.
dwManipulation	LONG*	[out]	Pointer that points to the LONG variable. Its value is the operation type.
Params3	LONG*	[out]	Pointer that points to the LONG variable. The value varies according to that of dwManipulation.
dwYear	LONG*	[out]	Pointer that points to the LONG variable. Its value is the year of an operation record.
dwMonth	LONG*	[out]	Pointer that points to the LONG variable. Its value is the month of an operation record.
dwDay	LONG*	[out]	Pointer that points to the LONG variable. Its value is the day of an operation record.
dwHour	LONG*	[out]	Pointer that points to the LONG variable. Its value is the hour of an operation record.

dwMinute	LONG*	[out]	Pointer that points to the LONG variable. Its value is the minute of an operation record.

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The meaning of different combinations of the dwManipulation, Params1, Params2, Params3 and Params4 parameters is as follows:

1. dwManipulation=0: dwManipulation indicates starting the machine.

2. dwManipulation=1: dwManipulation indicates shutting down the machine.

3. dwManipulation=3: dwManipulation indicates that an alarm is raised. Params1 specifies the alarm type. The value 58 indicates a misoperation alarm, 54 door sensor alarm, 53 door opening alarm, 55 tamper alarm, and 65535 shutdown alarm.

4. dwManipulation=4: dwManipulation indicates accessing the menu.

5. dwManipulation=5: dwManipulation indicates changing settings. Params1 specifies the number of the option that is set.

6. dwManipulation=6: dwManipulation indicates registering fingerprints. Params1 specifies the ID of the operated user. Params2 specifies the operation result, the value 0 indicates operation success, and other values indicate operation failure. Params3 specifies the registered fingerprint index. Params4 specifies the length of the fingerprint template, and the value 2 indicates duress fingerprint.

7. dwManipulation=7: dwManipulation indicates registering the password. Params1 specifies the ID of the operated user. Params2 specifies the operation result, the value 0 indicates operation success, and other values indicate operation failure.

8. dwManipulation=14: dwManipulation indicates creating an MF card. Params1 specifies the ID of the operated user. Params2 specifies the operation result, the value 0 indicates operation success, and other values indicate operation failure. Params3 specifies the number of fingerprints written to the MF card. Params4 specifies the size of fingerprint data written to the MF card.

9. dwManipulation=20: dwManipulation indicates copying data from the MF card to the machine. Params1 specifies the ID of the operated user. Params2 specifies the operation result, the value 0 indicates operation success, and other values indicate operation failure. Params3 specifies the number of fingerprints read to the MF card.

10. dwManipulation=22: dwManipulation indicates restoring factory settings.

11. dwManipulation=30: dwManipulation indicates registering a new user. Params1 specifies the ID of the operated user. Params2 specifies the operation result, the value 0 indicates operation success, and other values indicate operation failure.

12. dwManipulation=32: dwManipulation indicates duress alarm. Params1 specifies whether the alarm is a verification alarm, the value 0 indicates a key alarm, and 1 indicates a verification alarm. Note: If the alarm is a verification alarm, dwSEnrollNumber will return the duress user ID.

13. dwManipulation=34: dwManipulation indicates blockade. Params1 specifies whether to block.

Note

Applicable to BW, TFT and IFACE devices

5.2.2.4 GetAllSLogData

GetAllSLogData(LONG dwMachineNumber, LONG* dwTMachineNumber, LONG* dwSEnrollNumber, LONG* dwSMachineNumber, LONG* dwGEnrollNumber, LONG* dwGMachineNumber, LONG* dwManipulation, LONG* dwBackupNumber, LONG* dwYear, LONG* dwMonth, LONG* dwDay, LONG* dwHour, LONG* dwMinute)

To read operation records from the internal buffer one by one. Before using this function, execute ReadAllSLogData or ReadSuperLogData to read the operation records from the machine to the internal buffer of the PC. Each time this function is executed, the pointer moves to the next operation record. This function is the same as GetSuperLogData. They differ only in the interface name for compatibility.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwTMachineNumber	LONG*	[out]	Pointer that points to the LONG variable. Its value is the machine ID of an operation record.
dwSEnrollNumber	LONG*	[out]	Pointer that points to the LONG variable. Its value is the administrator ID of an operation record.
dwSMachineNumber	LONG*	[out]	Pointer that points to the LONG variable. The value varies

			according to that of dwManipulation.
dwGEnrollNumber	LONG*	[out]	Pointer that points to the LONG variable. The value varies according to that of dwManipulation.
dwGMachineNumber	LONG*	[out]	Pointer that points to the LONG variable. The value varies according to that of dwManipulation.
dwManipulation	LONG*	[out]	Pointer that points to the LONG variable. Its value is the operation type. The meanings of the values are as follows:
dwBackupNumber	LONG*	[out]	Pointer that points to the LONG variable. The value varies according to that of dwManipulation.
dwYear	LONG*	[out]	Pointer that points to the LONG variable. Its value is the year of an operation record.
dwMonth	LONG*	[out]	Pointer that points to the LONG variable. Its value is the month of an operation record.
dwDay	LONG*	[out]	Pointer that points to the LONG variable. Its value is the day of an operation record.
dwHour	LONG*	[out]	Pointer that points to the LONG variable. Its value is the hour of an operation record.
dwMinute	LONG*	[out]	Pointer that points to the LONG variable. Its value is the minute of an operation record.*

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

GetSuperLogData

Attention

The meanings of the parameters are the same as those of GetSuperLogData.

Note

Applicable to BW, TFT and IFACE devices

5.2.2.5 ClearSLog

VARIANT_BOOL ClearSLog(LONG dwMachineNumber)

To clear all operation records on the machine.

Parameters.

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.2.2.6 GetSuperLogData2

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

To read operation records from the internal buffer one by one. Before using this function, execute ReadAllSLogData or ReadSuperLogData to read the operation records from the machine to the internal buffer of the PC. Each time this function is executed, the pointer moves to the next operation record. GetSuperLogData and GetSuperLogData2 differ in that the GetSuperLogData2 function can obtain the operation record time accurate to seconds.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwTMachineNumber	LONG*	[out]	Pointer that points to the LONG variable. Its value is the machine ID of an operation record.
dwSEnrollNumber	LONG*	[out]	Pointer that points to the LONG variable. Its value is the administrator ID of an operation record.
Params4	LONG*	[out]	Pointer that points to the LONG variable. The value varies according to that of dwManipulation.
Params1	LONG*	[out]	Pointer that points to the LONG variable. The value varies according to that of dwManipulation.
Params2	LONG*	[out]	Pointer that points to the LONG variable. The value varies according to that of dwManipulation.
dwManipulation	LONG*	[out]	Pointer that points to the LONG

			variable. Its value is the operation type. The meanings of the values are as follows:
Params3	LONG*	[out]	Pointer that points to the LONG variable. The value varies according to that of dwManipulation.
dwYear	LONG*	[out]	Pointer that points to the LONG variable. Its value is the year of an operation record.
dwMonth	LONG*	[out]	Pointer that points to the LONG variable. Its value is the month of an operation record.
dwDay	LONG*	[out]	Pointer that points to the LONG variable. Its value is the day of an operation record.
dwHour	LONG*	[out]	Pointer that points to the LONG variable. Its value is the hour of an operation record.
dwMinute	LONG*	[out]	Pointer that points to the LONG variable. Its value is the minute of an operation record.
dwSecs	LONG*	[out]	Pointer that points to the LONG variable. Its value is the second of an operation record.

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

GetSuperLogData

Attention

dwYear, dwMonth, dwDay, dwHour, dwMinute and dwSecs are all pointers that point to the LONG variable. Their values indicate the date and time of an operation record, accurate to seconds.

Note

Applicable to BW, TFT and IFACE devices

5. 2. 3 User Information Functions

5.2.3.1 ReadAllUserID

VARIANT_BOOL ReadAllUserID(LONG dwMachineNumber)

To read all user information to the memory of the PC, including the user ID, password, name, and card number. Fingerprint templates are not read. After this function is executed, invoke the function GetAllUserID to get the user information.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.2.3.2 EnableUser

VARIANT_BOOL EnableUser(LONG dwMachineNumber, LONG dwEnrollNumber, LONG dwEMachineNumber, LONG dwBackupNumber, VARIANT_BOOL bFlag)

To set whether a user account is available.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	User ID
dwEMachineNumber	LONG	[in]	Invalid parameter. It is meaningless.
dwBackupNumber	LONG	[in]	Invalid parameter. It is meaningless.
bFlag	BOOL	[in]	Flag that indicates whether a user account is enabled

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The bFlag parameter is a flag that indicates whether a user account is enabled. The value True indicates that the user account is enabled and False indicates that the user account is disabled.

Note

Applicable to BW

5.2.3.3 SSR_EnableUser

VARIANT_BOOL SSR_EnableUser(LONG dwMachineNumber, BSTR dwEnrollNumber, VARIANT_BOOL bFlag)

To set whether a user account is available.

Parameters

Parameter description:

name	type	param direction	description of param

dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID
bFlag	BOOL	[in]	Flag that indicates whether a user account is enabled

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The bFlag parameter is a flag that indicates whether a user account is enabled. The value True indicates that the user account is enabled and False indicates that the user account is disabled.

Note

Applicable to TFT and IFACE devices

5.2.3.4 SetUserInfoEx

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

To upload the user verification mode or group verification mode.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	User ID
VerifyStyle	LONG	[in]	Verification mode
Reserved	BYTE*	[in]	Reserved

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. This function is valid only on machines with multiple verification modes.

2. On a monochrome machine, the VerifyStyle parameter specifies the verification mode, and 15 verification modes are available. For details about the meanings of the values, see the description of the GetGeneralLogData parameter. If a group verification mode is used, the value of VerifyStyle ranges from 129 to 124, indicating group 1 to group 5 respectively.

3. On a color-screen machine, the VerifyStyle parameter specifies the verification mode. On the color-screen access control fingerprint machine, the value 0 indicates group verification. 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).

Note

Applicable to BW, TFT and IFACE devices

5.2.3.5 GetAllUserID

VARIANT_BOOL GetAllUserID(LONG dwMachineNumber, LONG* dwEnrollNumber, LONG* dwEMachineNumber, LONG* dwBackupNumber, LONG* dwMachinePrivilege, LONG* dwEnable)

To get all user information. Before executing this function, invoke the function ReadAllUserID to read all user information to the memory. Each time GetAllUserID is executed, the pointer moves to the next user information record. After all user information is read, the function returns False. This function differs from GetAllUserInfo in that the GetAllUserInfo function can obtain user names and passwords.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

dwEnrollNumber	LONG*	[out]	User ID
dwEMachineNumber	LONG*	[out]	Invalid parameter
dwBackupNumber	LONG*	[out]	Comment
dwMachinePrivilege	LONG*	[out]	User privilege
dwEnable	LONG*	[out]	Whether a user account is enabled

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The value of dwEMachineNumber is always 0.

2. The dwMachinePrivilege parameter specifies the user privilege. The value 0 indicates common user, 1 registrar, 2 administrator, and 3 super administrator.

3. The dwEnable parameter specifies whether a user account is enabled. The value 1 indicates that the user account is enabled and 0 indicates that the user account is disabled.

Note

Applicable to BW

5.2.3.6 GetAllUserInfo

VARIANT_BOOL GetAllUserInfo(LONG dwMachineNumber, LONG* dwEnrollNumber, BSTR* Name, BSTR* Password, LONG* Privilege, VARIANT_BOOL* Enabled)

To get all user information. Before executing this function, invoke the function ReadAllUserID to read all user information to the memory. Each time GetAllUserInfo is executed, the pointer moves to the next user information record. After all user information is read, the function returns False. The GetAllUserInfo function differs from GetAllUserID in that it can obtain more information.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[out]	Machine ID
dwEnrollNumber	LONG*	[out]	User ID
Name	BSTR*	[out]	User name
Name	BSTR*	[out]	User password
Privilege	LONG*	[out]	User privilege
Enabled	BOOL*	[out]	Whether a user account is enabled

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The dwMachinePrivilege parameter specifies the user privilege. The value 0 indicates common user, 1 registrar, 2 administrator, and 3 super administrator.

2. The dwEnable parameter specifies whether a user account is enabled. The value 1 indicates that the user account is enabled and 0 indicates that the user account is disabled.

Note

Applicable to BW

5.2.3.7 GetUserInfoEx

VARIANT_BOOL GetUserInfoEx(LONG dwMachineNumber, LONG dwEnrollNumber, LONG* VerifyStyle, BYTE* Reserved)

To obtain the user verification mode.

Parameters

Parameter description:

name	type	param direction	description of param

dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	User ID
VerifyStyle	LONG*	[out]	The value is the user verification mode described by dwEnrollNumber.
Reserved	BYTE*	[out]	Reserved

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. This function is valid only on machines with multiple verification modes.

2. On a monochrome machine, the VerifyStyle parameter specifies the verification mode, and 15 verification modes are available. For details about the meanings of the values, see the description of the GetGeneralLogData parameter. If a group verification mode is used, the value of VerifyStyle ranges from 129 to 124, indicating group 1 to group 5 respectively.

3. On a color-screen machine, the VerifyStyle parameter specifies the verification mode. On the color-screen access control fingerprint machine, the value 0 indicates group verification. 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).

Note

Applicable to BW, TFT and IFACE devices

5.2.3.8 DeleteUserInfoEx

VARIANT_BOOL DeleteUserInfoEx(LONG dwMachineNumber, LONG dwEnrollNumber)

To delete the multiple verification modes set by a specified user.

Parameters

Parameter description:

name	type	param direction	description of param

dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	User ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

This function is valid only on machines with multiple verification modes.

Note

Applicable to BW, TFT and IFACE devices

5.2.3.9 SSR_GetAllUserInfo

VARIANT_BOOL SSR_GetAllUserInfo(LONG dwMachineNumber, BSTR* dwEnrollNumber, BSTR* Name, BSTR* Password, LONG* Privilege, VARIANT_BOOL* Enabled)

To get all user information. Before executing this function, invoke the function ReadAllUserID to read all user information to the memory. Each time SSR_GetAllUserInfo is executed, the pointer moves to the next user information record. After all user information is read, the function returns False.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR*	[out]	User ID
Name	BSTR*	[out]	User name
Name	BSTR*	[out]	User password
Privilege	LONG*	[out]	User privilege

Enabled	BOOL*	[out]	Flag that indicates whether a user account is enabled

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The Privilege parameter specifies the user privilege. The value 0 indicates common user, 1 registrar, 2 administrator, and 3 super administrator.

2. The Enable parameter specifies whether a user account is enabled. The value 1 indicates that the user account is enabled and 0 indicates that the user account is disabled.

Note

Applicable to TFT and IFACE devices

5.2.3.10 GetUserInfo

VARIANT_BOOL GetUserInfo(LONG dwMachineNumber, LONG dwEnrollNumber, BSTR* Name, BSTR* Password, LONG* Privilege, VARIANT_BOOL * Enabled)

To get information about a specified user.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	User ID
Name	BSTR*	[out]	Returned user name
Password	BSTR*	[out]	User password

Privilege	LONG*	[out]	User privilege
Enabled	BOOL*	[out]	Flag that indicates whether a user account is enabled

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The Password indicates the returned user password. If this parameter is left blank, the user does not use a password on the machine.

2. The Privilege parameter specifies the user privilege. The value 0 indicates common user, 1 registrar, 2 administrator, and 3 super administrator.

3. The Enable parameter specifies whether a user account is enabled. The value 1 indicates that the user account is enabled and 0 indicates that the user account is disabled.

Note

Applicable to BW

5.2.3.11 GetUserInfoByPIN2

VARIANT_BOOL GetUserInfoByPIN2(LONG dwMachineNumber, BSTR* Name, BSTR* Password, LONG* Privilege, VARIANT_BOOL* Enabled)

To obtain user information based on the current attribute value pin2.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
Name	BSTR*	[out]	User name
Password	BSTR*	[out]	User password

Privilege	LONG*	[out]	User privilege
Enabled	BOOL*	[out]	Flag that indicates whether a user account is enabled

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The Privilege parameter specifies the user privilege. The value 0 indicates common user, 1 registrar, 2 administrator, and 3 super administrator.

2. The Enable parameter specifies whether a user account is enabled. The value 1 indicates that the user account is enabled and 0 indicates that the user account is disabled.

Note

Applicable to BW

5.2.3.12 GetUserInfoByCard

VARIANT_BOOL GetUserInfoByCard(LONG dwMachineNumber, BSTR* Name, BSTR* Password, LONG* Privilege, VARIANT_BOOL* Enabled)

To obtain user information based on the current attribute value CardNumber.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
Name	BSTR*	[out]	User name
Password	BSTR*	[out]	User password
Privilege	LONG*	[out]	User privilege

Enabled	BOOL*	[out]	Flag that indicates whether a user account is enabled

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The Privilege parameter specifies the user privilege. The value 0 indicates common user, 1 registrar, 2 administrator, and 3 super administrator.

2. The Enable parameter specifies whether a user account is enabled. The value 1 indicates that the user account is enabled and 0 indicates that the user account is disabled.

Note

Applicable to BW

5.2.3.13 GetUserIDByPIN2

VARIANT_BOOL GetUserIDByPIN2(LONG PIN2, LONG* UserID)

To obtain the user ID based on pin2.

Parameters

Parameter description:

name	type	param direction	description of param
PIN2	LONG	[in]	Pin2 value
UserID	LONG*	[out]	User ID

Returns

Value description:

name	type	description of value

True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW

5.2.3.14 GetPIN2

VARIANT_BOOL GetPIN2(LONG UserID, LONG* PIN2)

To obtain the pin2 value based on the user ID.

Parameters

Parameter description:

name	type	param direction	description of param
UserID	LONG	[in]	User ID
PIN2	LONG*	[out]	Pin2 value

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW

5.2.3.15 SSR_GetUserInfo

VARIANT_BOOL SSR_GetUserInfo(LONG dwMachineNumber, BSTR dwEnrollNumber, BSTR* Name, BSTR* Password, LONG* Privilege, VARIANT_BOOL* Enabled)

To obtain information about a specified user.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID
Name	BSTR*	[out]	The value is the user name described by dwEnrollNumber.
Password	BSTR*	[out]	The value is the user password described by dwEnrollNumber.
Privilege	LONG*	[out]	The value is the user privilege described by dwEnrollNumber.
Enabled	BOOL*	[out]	Flag that indicates whether a user account is enabled

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The Privilege parameter specifies the user privilege. The value 0 indicates common user, 1 registrar, 2 administrator, and 3 super administrator.

2. The Enable parameter specifies whether a user account is enabled. The value 1 indicates that the user account is enabled and 0 indicates that the user account is disabled.

Note

Applicable to TFT and IFACE devices

5.2.3.16 SetUserInfo

VARIANT_BOOL SetUserInfo(LONG dwMachineNumber, LONG dwEnrollNumber, BSTR Name, BSTR Password, LONG Privilege, VARIANT_BOOL Enabled)

To set information about a user. If the user does not exist on the machine, the user will be created.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	User ID
Name	BSTR	[in]	User name to be set
Password	BSTR	[in]	User password to be set. If this parameter is left blank, the password of the user will be cleared on the machine.
Privilege	LONG	[in]	User privilege. The value 0 indicates common user, 1 registrar, 2 administrator, and 3 super administrator.
Enabled	BOOL	[in]	Flag that indicates whether a user account is enabled. The value 1 indicates that the user account is enabled and 0 indicates that the user account is disabled.

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The Password parameter specifies the password to be set. If this parameter is left blank, the password of the user will be cleared on the machine.

2. The Privilege parameter specifies the user privilege. The value 0 indicates common user, 1 registrar, 2 administrator, and 3 super administrator.

3. The Enable parameter specifies whether a user account is enabled. The value 1 indicates that the user account is enabled and 0 indicates that the user account is disabled.

Note

Applicable to BW

5.2.3.17 SSR_SetUserInfo

VARIANT_BOOL SetUserInfo(LONG dwMachineNumber, LONG dwEnrollNumber, BSTR Name, BSTR Password, LONG Privilege, VARIANT_BOOL Enabled)

To set information about a user. If the user does not exist on the machine, the user will be created.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID
Name	BSTR	[in]	User name
Password	BSTR	[in]	User password
Privilege	LONG	[in]	User privilege
Enabled	BOOL	[in]	Flag that indicates whether a user account is enabled

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The Privilege parameter specifies the user privilege. The value 0 indicates common user, 1 registrar, 2 administrator, and 3 super administrator.

2. The Enable parameter specifies whether a user account is enabled. The value 1 indicates that the user account is enabled and 0 indicates that the user account is disabled.

Note

Applicable to TFT and IFACE devices

5.2.3.18 ModifyPrivilege

VARIANT_BOOL ModifyPrivilege(LONG dwMachineNumber, LONG dwEnrollNumber, LONG dwEMachineNumber, LONG dwBackupNumber, LONG dwMachinePrivilege)

To modify user privilege.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	User ID
dwEMachineNumber	LONG	[in]	Invalid parameter. It is meaningless.
dwBackupNumber	LONG	[in]	Invalid parameter. It is meaningless.
dwMachinePrivilege	LONG	[in]	User privilege to be set

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The dwMachinePrivilege parameter specifies the user privilege. The value 0 indicates common user, 1 registrar, 2 administrator, and 3 super administrator.

Note

Applicable to BW

5. 2. 4 Registration Data Functions (Including Both User Information and Fingerprint)

5.2.4.1 GetEnrollData

VARIANT_BOOL GetEnrollData(LONG dwMachineNumber, LONG dwEnrollNumber, LONG dwEMachineNumber, LONG dwBackupNumber, LONG* dwMachinePrivilege, LONG* dwEnrollData, LONG* dwPassWord)

To obtain registration data (fingerprint template and part of the user information) based on the user ID and corresponding index.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	Machine ID
dwEMachineNumber	LONG	[in]	User ID
dwBackupNumber	LONG	[in]	Fingerprint index
dwMachinePrivilege	LONG*	[out]	User privilege
dwEnrollData	LONG*	[out]	Fingerprint template
dwPassWord	LONG*	[out]	Password

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The values of dwMachineNumber and dwEMachineNumber must be the same.

2. The fingerprint index ranges from 0 to 9. If the fingerprint template is obtained successfully, the password also is obtained. The index 10 indicates obtaining only the password.

3. The value 0 indicates common user, 1 registrar, 2 administrator, and 3 super administrator.

Note

Applicable to BW

5.2.4.2 SetEnrollData

VARIANT_BOOL SetEnrollData**(LONG dwMachineNumber, LONG dwEnrollNumber, LONG dwEMachineNumber, LONG dwBackupNumber, LONG dwMachinePrivilege, LONG* dwEnrollData, LONG dwPassWord)**

To set registration data (fingerprint template and part of the user information)

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	Machine ID
dwEMachineNumber	LONG	[in]	User ID
dwBackupNumber	LONG	[in]	Fingerprint index
dwMachinePrivilege	LONG	[in]	User privilege
dwEnrollData	LONG*	[in]	Fingerprint template to be uploaded

dwPassWord	LONG	[in]	User password

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The values of dwMachineNumber and dwEMachineNumber must be the same.
2. The fingerprint index ranges from 0 to 9. The index 10 indicates setting the user password.
3. The value 0 indicates common user, 1 registrar, 2 administrator, and 3 super administrator.

Note

Applicable to BW

5.2.4.3 DeleteEnrollData

VARIANT_BOOL DeleteEnrollData(LONG dwMachineNumber, LONG dwEnrollNumber, LONG dwEMachineNumber, LONG dwBackupNumber)

To delete registration data.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	User ID
dwEMachineNumber	LONG	[in]	Machine ID
dwBackupNumber	LONG	[in]	Fingerprint index

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The values of dwMachineNumber and dwEMachineNumber must be the same.

2. The dwBackupNumber parameter specifies the fingerprint index. The meanings are described as follows:
The index range is 0-9. The machine will also check whether a user has other fingerprints and passwords. If no, the machine will delete the user. The index 10 indicates deleting the password. The machine will also check whether the user has fingerprint data. If no, the machine will delete the user. The index 11 indicates deleting all fingerprint data of the user, and 12 indicates deleting the user (including the fingerprints, card number and password).

Note

Applicable to BW

5.2.4.4 SSR_DeleteEnrollData

VARIANT_BOOL SSR_DeleteEnrollData(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG dwBackupNumber)

To delete registration data.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID
dwBackupNumber	LONG	[in]	Fingerprint index

Returns

Value description:

name	type	description of value

True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The dwBackupNumber parameter specifies the fingerprint index. The meanings are described as follows:
The index range is 0-9. The machine will also check whether a user has other fingerprints and passwords. If no, the machine will delete the user. The index 10 indicates deleting the password. The machine will also check whether the user has fingerprint data. If no, the machine will delete the user. The index 11 indicates deleting all fingerprint data of the user, and 12 indicates deleting the user (including the fingerprints, card number and password).

Note

Applicable to TFT and IFACE devices

5.2.4.5 SSR_DeleteEnrollDataExt

VARIANT_BOOL SSR_DeleteEnrollDataExt(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG dwBackupNumber)

To delete registration data.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID
dwBackupNumber	LONG	[in]	Fingerprint index

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The dwBackupNumber parameter specifies the fingerprint index. The meanings are described as follows:
The index range is 0-9. The machine will also check whether a user has other fingerprints and passwords. If no, the machine will delete the user. The index 10 indicates deleting the password. The machine will also check whether the user has fingerprint data. If no, the machine will delete the user. The indexes 11 and 13 indicate deleting all fingerprint data of the user. The index 12 indicates deleting the user, including the fingerprints, card number and password.

2. This function differs from SSR_DeleteEnrollData in that it can delete all fingerprint data by using parameter 13, and therefore this function has higher efficiency.

Note

Applicable to TFT and IFACE devices

5.2.4.6 GetEnrollDataStr

VARIANT_BOOL GetEnrollDataStr(LONG dwMachineNumber, LONG dwEnrollNumber, LONG dwEMachineNumber, LONG dwBackupNumber, LONG* dwMachinePrivilege, BSTR* dwEnrollData, LONG* dwPassWord)

To obtain registration data (fingerprint template and part of the user information) based on the user ID and corresponding index. This function differs from GetEnrollData only in the fingerprint template format.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	Machine ID
dwEMachineNumber	LONG	[in]	User ID
dwBackupNumber	LONG	[in]	Fingerprint index
dwMachinePrivilege	LONG*	[out]	User privilege
dwEnrollData	BSTR*	[out]	Fingerprint template to be uploaded
dwPassWord	LONG*	[out]	User password

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The values of dwMachineNumber and dwEMachineNumber must be the same.

2. The fingerprint index ranges from 0 to 9. If the fingerprint template is obtained successfully, the password also is obtained. The index 10 indicates obtaining only the password.

3. The value 0 indicates common user, 1 registrar, 2 administrator, and 3 super administrator.

Note

Applicable to BW

5.2.4.7 SetEnrollDataStr

VARIANT_BOOL SetEnrollDataStr(LONG dwMachineNumber, LONG dwEnrollNumber, LONG dwEMachineNumber, LONG dwBackupNumber, LONG dwMachinePrivilege, BSTR dwEnrollData, LONG dwPassWord)

To set registration data (fingerprint template and part of the user information)

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID. The two values must be the same.
dwEnrollNumber	LONG	[in]	Machine ID. The two values must be the same.
dwEMachineNumber	LONG	[in]	User ID
dwBackupNumber	LONG	[in]	Fingerprint index
dwMachinePrivilege	LONG	[in]	User privilege
dwEnrollData	BSTR	[in]	Fingerprint template to be uploaded

dwPassWord	LONG	[in]	User password

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The values of dwMachineNumber and dwEMachineNumber must be the same.

2. The fingerprint index ranges from 0 to 9. If the fingerprint template is obtained successfully, the password also is obtained. The index 10 indicates obtaining only the password.

3. The value 0 indicates common user, 1 registrar, 2 administrator, and 3 super administrator.

4. The dwEnrollData parameter specifies the uploaded fingerprint template, which is a character string.

Note

Applicable to BW

5. 2. 5 Fingerprint Template Functions

5.2.5.1 ReadAllTemplate

VARIANT_BOOL ReadAllTemplate(LONG dwMachineNumber)

To read all fingerprint templates on the machine to the memory of the PC. This function reads all fingerprint data to the memory at a time. Compared with the function that reads data records from the machine one by one, this function has higher efficiency.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.2.5.2 SSR_GetUserTmp

VARIANT_BOOL GetUserTmp(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG dwFingerIndex, BYTE* TmpData, LONG* TmpLength)

To obtain a fingerprint template in binary format. This function differs from SSR_GetUserTmpStr only in the fingerprint template format.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID
dwFingerIndex	LONG	[in]	Fingerprint index, ranging from 0 to 9
TmpData	BYTE*	[in]	Fingerprint template data
TmpLength	LONG*	[in]	Length of the fingerprint template

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to TFT

5.2.5.3 SSR_GetUserTmpStr

VARIANT_BOOL GetUserTmpStr(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG dwFingerIndex, BSTR* TmpData, LONG* TmpLength)

To obtain a fingerprint template in character string format. This function differs from SSR_GetUserTmp only in the fingerprint template format.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID
dwFingerIndex	LONG	[in]	Fingerprint index, ranging from 0 to 9
TmpData	BSTR*	[out]	Fingerprint template data
TmpLength	LONG*	[out]	Length of the fingerprint template

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to TFT

5.2.5.4 SSR_SetUserTmp

VARIANT_BOOL SetUserTmp(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG

dwFingerIndex, BYTE* TmpData)

To upload a fingerprint template in binary format. This function differs from SSR_SetUserTmpStr only in the fingerprint template format.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID
dwFingerIndex	LONG	[in]	Fingerprint index, ranging from 0 to 9
TmpData	BYTE*	[in]	Fingerprint template

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to TFT

5.2.5.5 SSR_SetUserTmpStr

VARIANT_BOOL SetUserTmpStr(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG dwFingerIndex, BSTR TmpData)

To upload a fingerprint template in character string format. This function differs from SSR_SetUserTmp only in the fingerprint template format.

Parameters

Parameter description:

name	type	param direction	description of param

dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID
LONG dwFingerIndex	LONG	[in]	Fingerprint index, ranging from 0 to 9
TmpData	BSTR	[in]	Fingerprint template

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to TFT

5.2.5.6 DelUserTmp

VARIANT_BOOL DelUserTmp(LONG dwMachineNumber, LONG dwEnrollNumber, LONG dwFingerIndex)

To delete a specified fingerprint template.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	User ID
dwFingerIndex	LONG	[in]	Fingerprint index

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW

5.2.5.7 SSR_DelUserTmp

VARIANT_BOOL SSR_DelUserTmp(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG dwFingerIndex)

To delete a fingerprint template.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID
dwFingerIndex	LONG	[in]	Fingerprint index, ranging from 0 to 9

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to TFT and IFACE devices

5.2.5.8 SSR_SetUserTmpExt

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

To upload a fingerprint template. This function is an enhanced version of SSR_SetUserTmp.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
IsDeleted	LONG	[in]	Deletion flag
dwEnrollNumber	BSTR	[in]	User ID
dwFingerIndex	LONG	[in]	Fingerprint index, ranging from 0 to 9
TmpData	BYTE*	[in]	Fingerprint template

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

IsDeleted is a deletion flag. If a fingerprint with the specified index already exists on the machine when you upload a fingerprint template, this parameter specifies whether to overwrite the original fingerprint. The value 1 indicates overwriting the original fingerprint and 0 indicates not overwriting the original fingerprint.

Note

Applicable to TFT and IFACE devices

5.2.5.9 SSR_DelUserTmpExt

VARIANT_BOOL SSR_DelUserTmpExt(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG dwFingerIndex)

Deletes the specified fingerprint template for the specified user.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID
dwFingerIndex	LONG	[in]	Fingerprint template ID of the specified user

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to TFT devices

5.2.5.10 SetUserTmp

VARIANT_BOOL SetUserTmp(LONG dwMachineNumber, LONG dwEnrollNumber, LONG dwFingerIndex, BYTE* TmpData)

To upload a fingerprint template in binary format. This function differs from SetUserTmpStr only in the fingerprint template format.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	User ID

dwFingerIndex	LONG	[in]	Fingerprint index
TmpData	BYTE*	[in]	Fingerprint template data

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The user must already exist on the machine. If the index number of a user has been registered, the fingerprint template will be overwritten.

Note

Applicable to BW

5.2.5.11 SetUserTmpStr

VARIANT_BOOL SetUserTmpStr(LONG dwMachineNumber, LONG dwEnrollNumber, LONG dwFingerIndex, BSTR TmpData)

To obtain a fingerprint template in character string format. This function differs from SetUserTmp only in the fingerprint template format.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	User ID
dwFingerIndex	LONG	[in]	Fingerprint index
TmpData	BSTR	[in]	Fingerprint template

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The user must already exist on the machine. If the index number of a user has been registered, the fingerprint template will be overwritten.

Note

Applicable to BW

5.2.5.12 SetUserTmpEx

VARIANT_BOOL SetUserTmpEx(LONG dwMachineNumber,

LONG dwEnrollNumber, LONG dwFingerIndex, LONG Flag, BYTE* TmpData)

To upload fingerprint template ZKFinger 10.0 in binary format.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	User ID
dwFingerIndex	LONG	[in]	Fingerprint index
Flag	LONG	[in]	Flag that indicates whether the fingerprint template is valid or a duress fingerprint
TmpData	BYTE*	[in]	Fingerprint template data

Returns

Value description:

name	type	description of value

True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The machine firmware must support the duress fingerprint function. That is, the internal version of the firmware must be Ver6.60 or later.

2. The Flag parameter specifies whether the fingerprint template is valid or a duress fingerprint. The value 0 indicates that the fingerprint template is invalid, 1 indicates that the fingerprint template is valid, and 3 indicates that the fingerprint template is a duress fingerprint.

Note

Applicable to BW, TFT and IFACE devices

5.2.5.13 SetUserTmpExStr

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

To upload fingerprint template ZKFinger 10.0 in character string format.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	User ID
dwFingerIndex	LONG	[in]	Fingerprint index
Flag	LONG	[in]	Flag that indicates whether the fingerprint template is valid or a duress fingerprint
TmpData	BSTR	[in]	Fingerprint template

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success

False	BOOL	Function execution failure

See also

Attention

1. The machine firmware must support the duress fingerprint function. That is, the internal version of the firmware must be Ver6.60 or later.

2. The Flag parameter specifies whether the fingerprint template is valid or a duress fingerprint. The value 0 indicates that the fingerprint template is invalid, 1 indicates that the fingerprint template is valid, and 3 indicates that the fingerprint template is a duress fingerprint.

Note

Applicable to BW, TFT and IFACE devices

5.2.5.14 GetUserTmp

VARIANT_BOOL GetUserTmp(LONG dwMachineNumber, LONG dwEnrollNumber, LONG dwFingerIndex, BYTE* TmpData, LONG* TmpLength)

To obtain a fingerprint template in binary format. This function differs from GetUserTmpStr only in the fingerprint template format.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	User ID
dwFingerIndex	LONG	[in]	Fingerprint index
TmpData	BYTE*	[out]	Fingerprint template data
TmpLength	LONG*	[out]	Length of the fingerprint template

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW

5.2.5.15 GetUserTmpStr

VARIANT_BOOL GetUserTmpStr(LONG dwMachineNumber, LONG dwEnrollNumber, LONG dwFingerIndex, BSTR* TmpData, LONG* TmpLength)

To obtain a fingerprint template in character string format. This function differs from GetUserTmp only in the fingerprint template format.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	User ID
dwFingerIndex	LONG	[in]	Fingerprint index
TmpData	BSTR*	[out]	Fingerprint template data
TmpLength	LONG*	[out]	Length of the fingerprint template

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW

5.2.5.16 GetUserTmpEx

VARIANT_BOOL GetUserTmpEx(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG dwFingerIndex, LONG* Flag, BYTE* TmpData, LONG* TmpLength)

To obtain fingerprint template ZKFinger 10.0 in binary format.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID
dwFingerIndex	LONG	[in]	Fingerprint index
Flag	LONG*	[out]	Flag that indicates whether the fingerprint template is valid or a duress fingerprint
TmpData	BYTE*	[out]	Fingerprint template
TmpLength	LONG*	[out]	Length of the fingerprint template

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The machine firmware must support the duress fingerprint function. That is, the internal version of the firmware must be Ver6.60 or later.

The Flag parameter specifies whether the fingerprint template is valid or a duress fingerprint. The value 0 indicates that the fingerprint template is invalid, 1 indicates that the fingerprint template is valid, and 3 indicates that the fingerprint template is a duress fingerprint.

Note

Applicable to BW, TFT and IFACE devices

5.2.5.17 GetUserTmpExStr

VARIANT_BOOL GetUserTmpExStr(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG dwFingerIndex, LONG* Flag, BSTR* TmpData, LONG* TmpLength)

To obtain fingerprint template ZKFinger 10.0 in character string format.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID
dwFingerIndex	LONG	[in]	Fingerprint index
Flag	LONG*	[out]	Flag that indicates whether the fingerprint template is valid or a duress fingerprint
TmpData	BSTR*	[out]	Fingerprint template
TmpLength	LONG*	[out]	Length of the fingerprint template

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The machine firmware must support the duress fingerprint function. That is, the internal version of the firmware must be Ver6.60 or later.

2. The Flag parameter specifies whether the fingerprint template is valid or a duress fingerprint. The value 0 indicates that the fingerprint template is invalid, 1 indicates that the fingerprint template is valid, and 3 indicates that the fingerprint template is a duress fingerprint.

Note

Applicable to BW, TFT and IFACE devices

5.2.5.18 GetFPTempLength

VARIANT_BOOL GetFPTempLength(BYTE* dwEnrollData, LONG* Len)

To calculate length of a specified fingerprint template.

Parameters

Parameter description:

name	type	param direction	description of param
dwEnrollData	BYTE*	[in]	Pointer that points to the fingerprint template
Len	LONG*	[out]	The value is the fingerprint template length described by dwEnrollData.

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.2.5.19 GetFPTempLengthStr

VARIANT_BOOL GetFPTempLengthStr(BSTR dwEnrollData, LONG* Len)

To calculate length of a specified fingerprint template.

Parameters

Parameter description:

name	type	param direction	description of param

dwEnrollData	BSTR	[in]	Fingerprint template in character string format
Len	LONG*	[out]	The value is the fingerprint template length described by dwEnrollData.

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.2.5.20 FPTempConvert

VARIANT_BOOL FPTempConvert(BYTE* TmpData1, BYTE* TmpData2, LONG* Size)

To convert an offline fingerprint template into a BIOKEY fingerprint template. This function differs from FPTempConvertStr only in the data format.

Parameters

Parameter description:

name	type	param direction	description of param
TmpData1	BYTE*	[in]	Offline fingerprint template to be converted
TmpData2	BYTE*	[out]	The value is the BIOKEY fingerprint template after conversion.
Size	LONG*	[out]	The value is the size of the BIOKEY fingerprint template after conversion.

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.2.5.21 FPTempConvertStr

VARIANT_BOOL FPTempConvertStr(BSTR TmpData1, BSTR* TmpData2, LONG* Size)

To convert an offline fingerprint template into a BIOKEY fingerprint template in character string format. This function differs from FPTempConvert only in the data format.

Parameters

Parameter description:

name	type	param direction	description of param
TmpData1	BSTR	[in]	Offline fingerprint template to be converted
TmpData2	BSTR*	[out]	The value is the BIOKEY fingerprint template after conversion.
Size	LONG*	[out]	The value is the size of the BIOKEY fingerprint template after conversion.

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.2.5.22 FPTempConvertNew

VARIANT_BOOL FPTempConvertNew(BYTE* TmpData1, BYTE* TmpData2, LONG* Size)

To convert a BIOKEY fingerprint template into an offline fingerprint template. This function differs from FPTempConvertNewStr only in the data format.

Parameters

Parameter description:

name	type	param direction	description of param
TmpData1	BYTE*	[in]	Offline fingerprint template to be converted
TmpData2	BYTE*	[out]	The value is the offline fingerprint template after conversion.
Size	LONG*	[out]	The value is the size of the offline fingerprint template after conversion.

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.2.5.23 FPTempConvertNewStr

VARIANT_BOOL FPTempConvertNewStr(BSTR TmpData1, BSTR* TmpData2, LONG* Size)

To convert a BIOKEY fingerprint template into an offline fingerprint template in character string format. This function differs from FPTempConvertNew only in the data format.

Parameters

Parameter description:

name	type	param direction	description of param
TmpData1	BSTR	[in]	Offline fingerprint template to be converted
TmpData2	BSTR*	[out]	The value is the offline fingerprint template after conversion.
Size	LONG*	[out]	The value is the size of the offline fingerprint template after conversion.

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.2.6 Face Template Functions

5.2.6.1 SetUserFace

VARIANT_BOOL SetUserFace(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG dwFaceIndex, BYTE* TmpData, LONG TmpLength)

To upload a face template. This function differs from SetUserFaceStr only in the face template format.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

dwEnrollNumber	BSTR	[in]	User ID (not more than 24 digits)
dwFaceIndex	LONG	[in]	Face index
TmpData	BYTE*	[in]	Face template
TmpLength	LONG	[in]	Length of the face template

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The value of dwFaceIndex is always 50, which indicates uploading all face templates of a user.

Note

Applicable to IFACE

5.2.6.2 GetUserFace

VARIANT_BOOL GetUserFace(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG dwFaceIndex, BYTE* TmpData, LONG* TmpLength)

To download a face template. This function differs from GetUserFaceStr only in the face template format.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID (not more than 24 digits)
dwFaceIndex	LONG	[in]	Face index
TmpData	BYTE*	[out]	Face template

TmpLength	LONG*	[out]	Length of the face template

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The value of dwFaceIndex is always 50, which indicates downloading all face templates of a user.

Note

Applicable to IFACE

5.2.6.3 DelUserFace

VARIANT_BOOL DelUserFace(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG dwFaceIndex)

To delete a face template.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID (not more than 24 digits)
dwFaceIndex	LONG	[in]	Face index

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success

False	BOOL	Function execution failure

See also

Attention

The value of dwFaceIndex is always 50, which indicates downloading all face templates of a user.

Note

Applicable to IFACE

5.2.6.4 GetUserFaceStr

VARIANT_BOOL GetUserFaceStr(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG dwFaceIndex, BSTR* TmpData, LONG* TmpLength)

To download a face template. This function differs from GetUserFace in that it returns a face template in character string format.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID (not more than 24 digits)
dwFaceIndex	LONG	[in]	Face index
TmpData	BSTR*	[out]	Face template
TmpLength	LONG*	[out]	Length of the face template

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The value of dwFaceIndex is always 50, which indicates downloading all face templates of a user.

Note

Applicable to IFACE

5.2.6.5 SetUserFaceStr

VARIANT_BOOL SetUserFaceStr(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG dwFaceIndex, BSTR TmpData, LONG TmpLength)

To upload a face template. This function differs from SetUserFace only in the face template format.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID (not more than 24 digits)
dwFaceIndex	LONG	[in]	Face index
TmpData	BSTR	[in]	Face template
TmpLength	LONG	[in]	Length of the face template

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The value of dwFaceIndex is always 50, which indicates uploading all face templates of a user.

Note

Applicable to IFACE

5. 2. 7 User Verify Functions

5.2.7.1 SetUserVerifyStyle

VARIANT_BOOL SetUserVerifyStyle(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG VerifyStyle, BYTE* Reserved)

To set the user verification mode.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID
VerifyStyle	LONG	[in]	Verification mode
Reserved	BSTR*	[in]	Reserved

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The VerifyStyle parameter specifies the verification mode. The values are described as follows:

Group Verify = 0
FP/PW/RF = 128
FP = 129
PIN = 130
PW = 131
RF = 132
FP/PW = 133
FP/RF = 134
PW/RF = 135
PIN&FP = 136

FP&PW = 137
FP&RF = 138
PW&RF = 139
FP&PW&RF = 140
PIN&FP&PW = 141
FP&RF/PIN = 142

2. The Reserved parameter is reserved and not used at present.

Note

This interface is applicable to the new architecture firmware.

5.2.7.2 GetUserVerifyStyle

VARIANT_BOOL GetUserVerifyStyle(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG* VerifyStyle, BYTE* Reserved)

To obtain the user verification mode.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID
VerifyStyle	LONG*	[out]	Verification mode
Reserved	BSTR*	[out]	Reserved

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The VerifyStyle parameter specifies the verification mode. The values are described as follows:
Group Verify = 0
FP/PW/RF = 128

FP = 129
PIN = 130
PW = 131
RF = 132
FP/PW = 133
FP/RF = 134
PW/RF = 135
PIN&FP = 136
FP&PW = 137
FP&RF = 138
PW&RF = 139
FP&PW&RF = 140
PIN&FP&PW = 141
FP&RF/PIN = 142

2. The Reserved parameter is reserved and not used at present.

Note

This interface is applicable to the new architecture firmware.

5. 2. 8 Shortcut Keys Functions

5.2.8.1 SSR_SetShortkey

VARIANT_BOOL SSR_SetShortkey(LONG ShortKeyID, LONG ShortKeyFun, LONG StateCode, BSTR StateName, LONG StateAutoChange, BSTR StateAutoChangeTime)

To set a functional key. It is similar to the functional key definition function on the color-screen machine.

Parameters

Parameter description:

name	type	param direction	description of param
ShortKeyID	LONG	[in]	Key ID
ShortKeyFun	LONG	[in]	Function of the key
StateCode	LONG	[in]	State code of the status key
StateName	BSTR	[in]	Name of the status key
StateAutoChange	LONG	[in]	Auxiliary
StateAutoChangeTime	BSTR	[in]	Auxiliary

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. ShortKeyID: specifies the ID of the key. The mapping is as follows: F1 ? 1, F2 ? 2, F3 ? 3 ...

2. ShortKeyFun: function of the specified key. The value 0 indicates that the function of the key is not defined, 1 indicates that the specified key is a status key, 2 indicates the work code of the key, and 3 indicates viewing the short message. Note: the value of ShortKeyFun will influence settings of the following four parameters. Please refer to the below instructions.

3. StateCode: To set state code of specified status key. If the specified key is not a status key (namely, the value of ShortKeyFun is not 1), the value of StateCode will be ignored. If the specified key is a status key (ShortKeyFun=1), the state code of the specified status key is determined by the StateCode value, which ranges from 0 to 255.The state code of status key cannot be repeatedly, if the state code of different status key is set as the same, invoking of StateCode will fail. Such as F2 is a status key and the state code is 2; while you invoke StateCode Fun to set F3 as a status key and set its state code as 2, the invoking will fail.

4. StateName: To set name of status key. If the specified key is not a status key (namely, the value of ShortKeyFun is not 1), the value of StateName will be ignored. If the specified key is a status key (ShortKeyFun=1), the name of specified status key will be determined by the StateName value. At most, 18 characters are supported.

5. StateAutoChange: If the specified key is not a status key (namely, the value of ShortKeyFun is not 1), the value of StateAutoChange will be ignored. If the specified key is a status key (ShortKeyFun=1), the value of StateAutoChange indicates whether the status key automatically changes. 0: disable, 1: enable.

6. StateAutoChangeTime: If the specified key is not a status key (namely, the value of ShortKeyFun is not 1), the value of StateAutoChangeTime will be ignored. If the specified key is a status key (ShortKeyFun=1), the automatic change time of the status key is set by the return value of StateAutoChangeTime. Requirements as below:

1) "08:30;09:00;08:00;12:00;11:12;00:00;00:00;".

2) Hour and minute are separated with ":",Dates are separated with ";",space is not allowed between them.

3) Everyday's automatic change time shall be specified(one whole week is a cycle), reaches which the attendance state will change to specified state automatically (the name of automatic change state is decided by StateName, value is defined by the return value of StateCode). If

someday needs not to change attendance state automatically, set hour and minute of the StateAutoChangeTime as zero.

Note

Applicable to TFT

5.2.8.2 SSR_GetShortkey

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

To query the settings of a functional key.

Parameters

Parameter description:

name	type	param direction	description of param
ShortKeyID	LONG	[in]	Key ID
ShortKeyFun	LONG*	[out]	Function of the key
StateCode	LONG*	[out]	State code of the status key
StateName	BSTR*	[out]	Name of the status key
AutoChange	LONG*	[out]	Whether the status key automatically changes
AutoChangeTime	BSTR*	[out]	Automatic change time of the status key

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. ShortKeyID: specifies the ID of the key. The mapping is as follows: F1 ? 1, F2 ? 2, F3 ? 3 ...

2. ShortKeyFun: function of the specified key. The value 0 indicates that the function of the key is not defined, 1 indicates that the specified key is a status key, 2 indicates the work code of the

key, and 3 indicates viewing the short message.

3. StateCode: If the specified key is a status key (ShortKeyFun=1), the state code of the status key is returned. Otherwise, 0 is returned.

4. StateName: If the specified key is a status key (ShortKeyFun=1), the name of the status key is returned. Otherwise, a blank character string is returned.

5. AutoChange: If the specified key is a status key (ShortKeyFun=1), the value of this parameter indicates whether the status key automatically changes. Otherwise, 0 is returned.

6. AutoChangeTime: If the specified key is a status key (ShortKeyFun=1), the automatic change time of the status key is returned, in the format of a character string. Otherwise, a blank character string is returned.

Note

Applicable to TFT

5.2.8.3 EnableCustomizeAttState

VARIANT_BOOL EnableCustomizeAttState(LONG dwMachineNumber, LONG StateID, LONG Enable)

To specify whether to enable a customized attendance status value.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
StateID	LONG	[in]	Attendance status value to be defined
Enable	LONG	[in]	Whether to enable the attendance status definition function for the attendance status value specified by StateID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. This function is a customization function. To use this function, the extension function must be enabled for the machine and the machine must support the attendance status definition function.

2. Attendance status definition function: The mapping between the attendance status values and the states are as follows:
If this function is enabled, you can invoke SetCustomizeAttState to change the status value of a state: 0-Check-In 1-Check-Out 2-Break-Out 3-Break-In 4-OT-In 5-OT-Out
For example, EnableCustomizeAttState(1,0,1)// indicates enabling the customized state of 0 (check-in)
SetCustomizeAttState(1,0,8)// indicates setting the customized status value to 8 for the status value 0 (check-in)
If a user chooses check-in on the machine and passes verification, the saved attendance status value is 8.

Note

Applicable to BW

5.2.8.4 SetCustomizeAttState

VARIANT_BOOL SetCustomizeAttState(LONG dwMachineNumber, LONG StateID, LONG NewStatel)

To set a customized attendance status value based on the original attendance status value.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
StateID	LONG	[in]	Original status value
NewStatel	LONG	[in]	New status value

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

This function is a customization function. To use this function, the extension function must be enabled for the machine and the machine must support the attendance status definition function. For details, see the description of EnableCustomizeAttState.

Note

Applicable to BW

5.2.8.5 DelCustomizeAttState

VARIANT_BOOL DelCustomizeAttState(LONG dwMachineNumber, LONG StateID)

To delete the customized attendance status value of an original attendance status value.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
StateID	LONG	[in]	Original attendance status value of which the customized attendance value is to be deleted

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

This function is a customization function. To use this function, the extension function must be enabled for the machine and the machine must support the attendance status definition function. For details, see the description of EnableCustomizeAttState.

Note

Applicable to BW

5.2.8.6 GetAllSFIDName

VARIANT_BOOL GetAllSFIDName(LONG dwMachineNumber, BSTR* ShortkeyIDName, LONG BufferSize1, BSTR* FunctionIDName, LONG BufferSize2)

To query the shortcut key name and function lists.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
ShortkeyIDName	BSTR*	[out]	List of shortcut key IDs and names
BufferSize1	LONG	[in]	Size of the buffer for storing the list of shortcut key IDs and names
FunctionIDName	BSTR*	[out]	List of function IDs and names
BufferSize2	LONG	[in]	Size of the buffer for storing the list of function IDs and names

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. ShortkeyIDName: specifies the list of shortcut key IDs and names, in the format of "ID,Key_Name\r\n1,F1\r\n2,F2...".

2. FunctionIDName: specifies the list of function IDs and names, in the format of "ID,Func_Name\r\n1,adduser\r\n2,userlist...".

3. The values of BufferSize1 and BufferSize2 are est to 4 kB.

Note

This interface is applicable to the new architecture firmware.

5.2.8.7 GetShortkey

VARIANT_BOOL GetShortkey(LONG dwMachineNumber, LONG ShortKeyID, BSTR* ShortKeyName, BSTR* FunctionName, LONG* ShortKeyFun, LONG* StateCode, BSTR* StateName, BSTR* Description, LONG* AutoChange, BSTR* AutoChangeTime)

To query the settings of a functional key.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
ShortKeyID	LONG	[in]	Key ID
ShortKeyName	BSTR*	[out]	Key name
ShortKeyName	BSTR*	[out]	Function name of the key
ShortKeyFun	LONG*	[out]	Key type
StateCode	LONG*	[out]	State code of the status key
StateName	BSTR*	[out]	Name of the status key
Description	BSTR*	[out]	Description of the status key
AutoChange	LONG*	[out]	Whether the status key automatically changes
AutoChangeTime	BSTR*	[out]	Automatic change time of the status key

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The mapping between ShortKeyName and ShortKeyID is as follows: F1-1, F2 -2, F3 -3 ...

2. ShortKeyFun: function of the specified key. The value 0 indicates a functional key and 1 indicates a status key.

3. ShortKeyName: name of the key specified by ShortKeyID.

4. FunctionName: If the specified key is a functional key (ShortKeyFun=0), the function name is returned.

5. FunctionName: If the specified key is a status key (ShortKeyFun=1), the name of the key is returned. In this case, the value of FunctionName is the same as that of StateName.

6. StateCode: If the specified key is a status key (ShortKeyFun=1), the state code of the status key is returned. Otherwise, an invalid value is returned.

7. StateName: If the specified key is a status key (ShortKeyFun=1), the name of the status key is returned. Otherwise, an invalid value is returned.

8. AutoChange: If the specified key is a status key (ShortKeyFun=1), the value of this parameter indicates whether the status key automatically changes. Otherwise, an invalid value is returned.

9. Description: If the specified key is a status key (ShortKeyFun=1), the description of the status key is returned. Otherwise, an invalid value is returned.

10. AutoChangeTime: If the specified key is a status key (ShortKeyFun=1), the automatic change time of the status key is returned, in the format of a character string. Otherwise, an invalid value is returned.

Note

This interface is applicable to the new architecture firmware.

5.2.8.8 SetShortkey

VARIANT_BOOL SetShortkey(LONG dwMachineNumber, LONG ShortKeyID, BSTR ShortKeyName, BSTR FunctionName, LONG ShortKeyFun, LONG StateCode, BSTR StateName, BSTR Description, LONG StateAutoChange, BSTR StateAutoChangeTime)

To set a functional key. It is similar to the functional key definition function on the color-screen machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
ShortKeyID	LONG	[in]	Key ID

ShortKeyName	BSTR	[in]	Key name
FunctionName	BSTR	[in]	Function name of the key
ShortKeyFun	LONG	[in]	Key type
StateCode	LONG	[in]	State code of the status key
StateName	BSTR	[in]	Name of the status key
Description	BSTR	[in]	Description of the status key
AutoChange	LONG	[in]	Whether the status key automatically changes
AutoChangeTime	BSTR	[in]	Automatic change time of the status key

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. When setting ShortKeyName and ShortKeyID, ensure that the mapping between ShortKeyName and ShortKeyID is as follows: F1-1, F2 -2, F3-3 ...

2. ShortKeyFun: function of the specified key. The value 0 indicates a functional key and 1 indicates a status key. Note that the value of this parameter will affect the settings of other four parameters.

3. StateCode: state code of the status key.
If the specified key is not a functional key, that is, the value of ShortKeyFun is not 1, the value of this parameter will be ignored.
If the specified key is a status key, that is, the value of ShortKeyFun is 1, the value of this parameter is the state code of the status key, and ranges from 0 to 255. The state values of status keys cannot be duplicate. Otherwise, function invocation will fail. For example, F2 is a status key and its state code is 2. If you invoke this function to set F3 as a status key and set its state code to 2, the function invocation will fail.

4. StateName: name of the status key.

If the specified key is not a functional key, that is, the value of ShortKeyFun is not 1, the value of this parameter will be ignored.
If the specified key is a status key, that is, the value of ShortKeyFun is 1, the value of this parameter is the name of the status key, and contains at most 18 characters.

5. Description: description of the status key.
If the specified key is not a functional key, that is, the value of ShortKeyFun is not 1, the value of this parameter will be ignored.
If the specified key is a status key, that is, the value of ShortKeyFun is 1, the value of this parameter is the description of the status key.

6. StateAutoChange: whether the status key automatically changes:
If the specified key is not a functional key, that is, the value of ShortKeyFun is not 1, the value of this parameter will be ignored.
If the specified key is a status key, that is, the value of ShortKeyFun is 1, the value of this parameter indicates whether the status key automatically changes. The value 0 indicates that the status key automatically changes and 1 indicates that the status key does not automatically change.

7. StateAutoChangeTime:
If the specified key is not a functional key, that is, the value of ShortKeyFun is not 1, the value of this parameter will be ignored.
If the specified key is a status key, the value of this parameter is the automatic change time of the status key. The details are as follows: 1. 08:30;09:00;08:00;12:00;11:12;00:00;00:00; 2. The hour is separated from the minute with a colon (:), and days are separated with a semicolon (;), free from spaces. 3. You must specify the automatic change time of each day, with a week as a cycle. After the specified time arrives, the attendance status will change to the state specified by StateName and StateCode. If automatic change is disabled on a day, the hour and minute are both set to 0.

8. If the return value of issued status key is -15001 repeatedly, the return value of description will be -15002 repeatedly.

Note

This interface is applicable to the new architecture firmware.

5.2.9 Work Code Functions

5.2.9.1 SetWorkCode

VARIANT_BOOL SetWorkCode(LONG WorkCodeID, LONG AWorkCode)

To define a work code with a specified ID.

Parameters

Parameter description:

name	type	param direction	description of param

WorkCodeID	LONG	[in]	Work code ID
AWorkCode	LONG	[in]	Value of work code described by WorkCodeID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

A work code value within any range can be input on a monochrome machine. After the work code is defined by using this function, the user can input only the defined work code. For example, if the work code is defined as SetWorkCode(1,345) and SetWorkCode(2,567), the user can input only the work code with the values 345 and 567.

Note

Applicable to BW

5.2.9.2 GetWorkCode

VARIANT_BOOL GetWorkCode(LONG WorkCodeID, LONG* AWorkCode)

To obtain the name of a specified work code ID. For details, see the description of SetWorkCode.

Parameters

Parameter description:

name	type	param direction	description of param
WorkCodeID	LONG	[in]	Work code ID
AWorkCode	LONG*	[out]	Obtained value of work code described by WorkCodeID

Returns

Value description:

name	type	description of value

True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW

5.2.9.3 SSR_GetWorkCode

VARIANT_BOOL SSR_GetWorkCode(LONG AWorkCode, BSTR* Name)

To obtain the name of a specified work code ID.

Parameters

Parameter description:

name	type	param direction	description of param
AWorkCode	LONG	[in]	Work code ID
Name	BSTR*	[out]	Obtained value of work code described by WorkCodeID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to TFT

5.2.9.4 SSR_SetWorkCode

VARIANT_BOOL SSR_SetWorkCode(LONG AWorkCode, BSTR Name)

To set a work code with a specified ID.

Parameters

Parameter description:

name	type	param direction	description of param
AWorkCode	LONG	[in]	Work code ID
Name	BSTR	[in]	Value of work code described by WorkCodeID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to TFT

5.2.9.5 SSR_DeleteWorkCode

VARIANT_BOOL SSR_DeleteWorkCode(LONG AWorkCode)

To delete a work code with a specified ID.

Parameters

Parameter description:

name	type	param direction	description of param
AWorkCode	LONG	[in]	Work code ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success

False	BOOL	Function execution failure

See also

Attention

Note

Applicable to TFT

5.2.9.6 SSR_ClearWorkCode

VARIANT_BOOL SSR_ClearWorkCode()

To delete all user-defined work codes.

Parameters

None

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to TFT

5.2.9.7 DeleteWorkCode

VARIANT_BOOL DeleteWorkCode(LONG WorkCodeID)

To delete a work code with a specified work code ID. For details, see the description of SetWorkCode.

Parameters

Parameter description:

name	type	param direction	description of param
WorkCodeID	LONG	[in]	Work code ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW

5.2.9.8 ClearWorkCode

VARIANT_BOOL ClearWorkCode()

To clear all defined work codes on the machine. For details, see the description of SetWorkCode.

Parameters

None

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW

5.2.9.9 SSR_GetWorkCodeIDByName

VARIANT_BOOL SSR_GetWorkCodeIDByName(LONG dwMachineNumber, BSTR WorkCodeName, LONG* WorkCodeId)

To query the interface of work code id based on the work code name.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
WorkCodeName	BSTR	[in]	Work code name
WorkCodeId	LONG*	[out]	Work code ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Together with the interface of SSR_SetWorkCode, to judge whether the issued workname is repeatedly (the same WorkCodeName cannot be issued). When return value of WorkCodeID is greater than zero, the issued workname has existed.

Note

This interface is applicable to the new architecture firmware.

5. 2. 10 SMS Functions

5.2.10.1 SetSMS

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

To add a short message to the machine. To set the short message for a user, invoke SetUserSMS to assign the short message to the user.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
ID	LONG	[in]	Short message ID

Tag	LONG	[in]	Short message type
ValidMinutes	LONG	[in]	Validity period of the short message
StartTime	BSTR	[in]	Effective time of the short message
Content	BSTR	[in]	Content of the short message

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. Tag specifies the short message type. The value 253 indicates public short message, 254 personal short message, and 255 reserved short message.

2. ValidMinutes specifies the validity period of the short message. The value ranges from 0 to 65535. The short message will take effect at the time specified by StartTime and will last for a period of time specified by ValidMinutes.

3. StartTime specifies the effective time of the short message, in the format of yyyy-mm-dd hh:mm:ss.

Note

Applicable to BW and TFT devices

5.2.10.2 SetUserSMS

VARIANT_BOOL SetUserSMS(LONG dwMachineNumber, LONG dwEnrollNumber, LONG SMSID)

To set the short message of a user. Specifically, this function is used to assign a short message with a specific ID to a user.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

dwEnrollNumber	LONG	[in]	User ID
SMSID	LONG	[in]	Short message ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW

5.2.10.3 SSR_SetUserSMS

VARIANT_BOOL SSR_SetUserSMS(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG SMSID)

To set the short message of a user. Specifically, this function is used to assign a short message with a specific ID to a user.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID
SMSID	LONG	[in]	Short message ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success

False	BOOL	Function execution failure

See also

Attention

Note

Applicable to TFT

5.2.10.4 GetSMS

VARIANT_BOOL GetSMS(LONG dwMachineNumber, LONG ID, LONG* Tag, LONG* ValidMinutes, BSTR* StartTime, BSTR* Content)

To obtain details about a short message from the machine based on the short message ID, including the content, effective time, message type, and validity period.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
ID	LONG	[in]	Short message ID
Tag	LONG*	[out]	Short message type
ValidMinutes	LONG*	[out]	Validity period of the short message
StartTime	BSTR*	[out]	Effective time of the short message
Content	BSTR*	[out]	Content of the short message

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. Tag specifies the short message type. The value 253 indicates public short message, 254 personal short message, and 255 reserved short message.

2. ValidMinutes specifies the validity period of the short message. The value ranges from 0 to 65535. The short message will take effect at the time specified by StartTime and will last for a period of time specified by ValidMinutes.

3. StartTime specifies the effective time of the short message, in the format of yyyy-mm-dd hh:mm:ss.

Note

Applicable to BW and TFT devices

5.2.10.5 DeleteSMS.

VARIANT_BOOL DeleteSMS(LONG dwMachineNumber, LONG ID)

To delete a short message with a specified ID from the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
ID	LONG	[in]	Short message ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW and TFT devices

5.2.10.6 DeleteUserSMS

VARIANT_BOOL DeleteUserSMS(LONG dwMachineNumber, LONG dwEnrollNumber, LONG SMSID)

To delete the short message with a specified ID for a specified user. Only the mapping relationship between the user and the short message is deleted, and the short message is not deleted.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	User ID
SMSID	LONG	[in]	Short message ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW

5.2.10.7 SSR_DeleteUserSMS

VARIANT_BOOL SSR_DeleteUserSMS(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG SMSID)

To delete the short message with a specified ID for a specified user. Only the mapping relationship between the user and the short message is deleted, and the short message is not deleted.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

dwEnrollNumber	BSTR	[in]	User ID
SMSID	LONG	[in]	Short message ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to TFT

5.2.10.8 ClearUserSMS

VARIANT_BOOL ClearUserSMS(LONG dwMachineNumber)

To clear all mapping relationships between short messages and users. The short messages are not deleted.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW and TFT devices

5.2.10.9 ClearSMS

VARIANT_BOOL ClearSMS(LONG dwMachineNumber)

To clear all short messages on the machine. All short messages will be deleted from the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW and TFT devices

5. 2. 11 Holiday Functions

5.2.11.1 SetHoliday

VARIANT_BOOL SetHoliday(LONG dwMachineNumber, BSTR Holiday)

To set holidays.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Holiday	BSTR	[in]	Holiday to be set

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The Holiday parameter specifies holidays in the format of mmddmmdd. For example, 04140511 indicates a holiday that lasts from April 14 to May 11.

Note

Applicable to BW

5.2.11.2 GetHoliday

VARIANT_BOOL GetHoliday(LONG dwMachineNumber, BSTR* Holiday)

To query the holiday that is set on the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
Holiday	BSTR*	[out]	Holiday specified on the machine

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The Holiday parameter specifies holidays in the format of mmddmmdd. For example, 04140511 indicates a holiday that lasts from April 14 to May 11.

Note

Applicable to BW

5.2.11.3 SSR_GetHoliday

VARIANT_BOOL SSR_GetHoliday(LONG dwMachineNumber, LONG HolidayID, LONG* BeginMonth, LONG* BeginDay, LONG* EndMonth, LONG* EndDay, LONG* TimeZoneID)

To query holiday settings on the machine based on the holiday ID.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
HolidayID	LONG	[in]	Holiday ID
BeginMonth	LONG*	[out]	Start date of the holiday
BeginDay	LONG*	[out]	Start date of the holiday
EndMonth	LONG*	[out]	End date of the holiday
EndDay	LONG*	[out]	End date of the holiday
TimeZoneID	LONG*	[out]	Index of the time segment of the holiday

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to TFT and IFACE devices

5.2.11.4 SSR_SetHoliday

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

To set holidays.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
HolidayID	LONG	[in]	Holiday ID
BeginMonth	LONG	[in]	Start date of the holiday
BeginDay	LONG	[in]	Start date of the holiday
EndMonth	LONG	[in]	End date of the holiday
EndDay	LONG	[in]	Start date of the holiday
TimeZoneID	LONG	[in]	Index of the time segment of the holiday

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to TFT and IFACE devices

5. 2. 12 DST Functions

5.2.12.1 SetDaylight

VARIANT_BOOL SetDaylight(LONG dwMachineNumber, LONG Support, BSTR BeginTime, BSTR EndTime)

To set whether to enable DST, and the DST start time and end time.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
Support	LONG	[in]	Whether to enable DST
BeginTime	BSTR	[in]	DST start time
EndTime	BSTR	[in]	DST end time

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The Support parameter specifies whether to enable DST. The value 1 indicates enabling DST and 0 indicates disabling DST.

2. The values of BeginTime and EndTime are in the format of mm-dd hh:mm.

Note

Applicable to BW, TFT, and IFACE devices

5.2.12.2 GetDaylight

VARIANT_BOOL GetDaylight(LONG dwMachineNumber, LONG* Support, BSTR* BeginTime, BSTR* EndTime)

To query the DST settings on the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
Support	LONG*	[out]	Whether to enable DST
BeginTime	BSTR*	[out]	DST start time
EndTime	BSTR*	[out]	DST end time

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The Support parameter specifies whether to enable DST. The value 1 indicates enabling DST and 0 indicates disabling DST.

2. The values of BeginTime and EndTime are in the format of mm-dd hh:mm.

Note

Applicable to BW, TFT and IFACE devices

5. 2. 13 System Data Management Functions

5.2.13.1 ClearKeeperData

VARIANT_BOOL ClearKeeperData(LONG dwMachineNumber)

To clear all data on the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.2.13.2 ClearData

VARIANT_BOOL ClearData(LONG dwMachineNumber, LONG DataFlag)

To clear the records specified by DataFlag on the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
DataFlag	LONG	[in]	This parameter specifies the type of records to be cleared.

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The DataFlag parameter specifies the type of records to be cleared. The value range is 1-5. The meanings are as follows:

1 Attendance records
2 Fingerprint template data
3 None
4 Operation records
5 User information
If the value of this parameter is 5, all users on the machine will be deleted. Note: All fingerprint templates will also be deleted.

Note

Applicable to BW, TFT and IFACE devices

5.2.13.3 GetDataFile

VARIANT_BOOL GetDataFile(LONG dwMachineNumber, LONG DataFlag, BSTR FileName)

To obtain a specified data file from the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
DataFlag	LONG	[in]	Type of the data file to be obtained
FileName	BSTR	[in]	Name of the data file that is stored

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The DataFlag parameter specifies the type of records to be cleared. The values are described as follows:
1 Attendance records

2 Fingerprint template data
3 None
4 Operation records
5 User information
6 Short message data file
7 Data file of short messages and user relationships
8 Extended user information data file
9 Work code information data file

Note

Applicable to BW, TFT and IFACE devices

5.2.13.4 SendFile

VARIANT_BOOL SendFile(LONG dwMachineNumber, BSTR FileName)

To send a file to the machine, usually to the directory /mnt/mtdblock/.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
FileName	BSTR	[in]	Name of the file to be sent

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

If the color-screen machine transfers user pictures or advertisement pictures, name the pictures properly and then the pictures will be automatically saved to the corresponding directory.
Name format of advertisement pictures: ad_Number.jpg. The value of number ranges from 1 to 20. For example, ad_4.jpg.
Name format of user pictures: User ID.jpg. For example, 1.jpg.

Note

Applicable to BW, TFT and IFACE devices

5.2.13.5 ReadFile

VARIANT_BOOL ReadFile(LONG dwMachineNumber, BSTR FileName, BSTR FilePath)

To read a specified file from the machine, usually under the directory /mnt/mtdblock/.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
FileName	BSTR	[in]	Name of the file to be read
FilePath	BSTR	[in]	Path for saving the file

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW

5.2.13.6 RefreshData

VARIANT_BOOL RefreshData(LONG dwMachineNumber)

To refresh the data on the machine. This function is typically invoked after user information or a fingerprint is uploaded, so that the modification takes effect immediately.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 2. 14 User Photo & Attendance Photo

5.2.14.1 UploadUserPhoto

VARIANT_BOOL UploadUserPhoto(LONG dwMachineNumber, BSTR FileName)

To upload a user picture to the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
FileName	BSTR	[in]	User picture name and the absolute path where the picture resides

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The FileName parameter specifies the absolute path where the user picture to be uploaded

resides, for example, C:\Users\HP\Desktop\11.jpg. The user picture is named in the format of User ID.jpg.

2. This function can also be implemented by using the SendFile function.

3. You can invoke this function to upload multiple user pictures by traversing through all file names under a specified directory.

Note

This interface is applicable to the new architecture firmware.

5.2.14.2 DownloadUserPhoto

VARIANT_BOOL DownloadUserPhoto(LONG dwMachineNumber, BSTR FileName, BSTR FilePath)

To download a user picture from the machine to the software.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
FileName	BSTR	[in]	Name of the user picture, with the file name extension .jpg
FilePath	BSTR	[in]	Absolute path where the downloaded user picture is to be saved

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The FileName parameter specifies the name of the user picture to be downloaded. The name format is User ID.jpg, for example, 11.jag. The downloaded picture will be saved in JPG format to the specified path.

2. The FilePath parameter specifies the absolute path where the downloaded picture is to be saved, for example, C:\Users\HP\Desktop\.

Note

This interface is applicable to the new architecture firmware.

5.2.14.3 DeleteUserPhoto

VARIANT_BOOL DeleteUserPhoto(LONG dwMachineNumber, BSTR FileName)

To delete a single user picture or all user pictures on the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
FileName	BSTR	[in]	Name of the user picture to be deleted

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The FileName parameter is used to set to delete one or all user pictures. To delete a single user picture, specify the name of the picture, which is named in the format of User ID.jpg, for example, 11.jpg. To delete all user pictures at a time, set this parameter to ALL, which is case-sensitive.

2. The FilePath parameter specifies the absolute path where the downloaded picture is to be saved, for example, C:\Users\HP\Desktop\.

Note

This interface is applicable to the new architecture firmware.

5.2.14.4 GetAllUserPhoto

VARIANT_BOOL GetAllUserPhoto(LONG dwMachineNumber, BSTR dlDir)

To download all user pictures from the machine to the software.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dlDir	BSTR	[in]	Absolute path where the downloaded user pictures are to be saved

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. The dlDir parameter specifies the absolute path where the downloaded pictures are to be saved, for example, C:\Users\HP\Desktop\.

2. All user pictures downloaded from the machine will be automatically saved in JPG format to the specified path one by one.

Note

This interface is applicable to the new architecture firmware.

5.2.14.5 GetPhotoNamesByTime

VARIANT_BOOL GetPhotoNamesByTime(LONG dwMachineNumber, LONG iFlag, BSTR sTime, BSTR eTime, BSTR* AllPhotoName)

To download attendance pictures from the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
iFlag	LONG	[in]	Flag bit

sTime	BSTR	[in]	Start time
eTime	BSTR	[in]	End time
AllPhotoName	BSTR*	[out]	Names of attendance pictures

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. If the value of iFlag is 0, all attendance pictures on the machine will be downloaded. If the value is 1, attendance pictures between sTime and eTime will be downloaded.

2. sTime: specifies the start time, in the format of YYYY-MM-DD hh:mm:ss.

3. eTime: specifies the end time, in the format of YYYY-MM-DD hh:mm:ss.

4. AllPhotoName: specifies the names of attendance pictures, in the format of verification success pictures (separated with \t)+\n+verification failure pictures (separated with \t).

Note

Applicable to TFT

5.2.14.6 GetPhotoByName

VARIANT_BOOL GetPhotoByName(LONG dwMachineNumber, BSTR PhotoName, BYTE* PhotoData, LONG* PhotoLength)

To download an attendance picture from the machine based on the picture name.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
PhotoName	BSTR	[in]	Picture name

PhotoData	BYTE*	[out]	Picture data in binary format
PhotoLength	LONG*	[out]	Picture size

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to TFT

5.2.14.7 GetPhotoCount.

VARIANT_BOOL GetPhotoCount(LONG dwMachineNumber, LONG* Count, LONG iFlag)

To query the number of attendance pictures on the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
Count	LONG*	[in]	Picture quantity
iFlag	LONG	[in]	Flag bit

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success

False	BOOL	Function execution failure

See also

Attention

If the value of iFlag is 0, the total number of attendance pictures is returned. If the value is 1, the number of pictures that pass verification is returned. If the value is 2, the number of pictures that fail verification is returned.

Note

Applicable to TFT

5.2.14.8 ClearPhotoByTime

VARIANT_BOOL ClearPhotoByTime(LONG dwMachineNumber, LONG iFlag, BSTR sTime, BSTR eTime)

To clear attendance pictures on the machine according to specified conditions.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
iFlag	LONG	[in]	Flag bit
sTime	BSTR	[in]	Start time
eTime	BSTR	[in]	End time

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. If the value of iFlag is 0, all attendance pictures on the machine will be downloaded. If the value is 1, attendance pictures between sTime and eTime will be downloaded.

2. sTime: specifies the start time, in the format of YYYY-MM-DD hh:mm:ss.

3. eTime: specifies the end time, in the format of YYYY-MM-DD hh:mm:ss.

Note

Applicable to TFT

5. 2. 15 Bell Functions

5.2.15.1 GetBellSchDataEx

VARIANT_BOOL GetBellSchDataEx(LONG dwMachineNumber, LONG weekDay, LONG Index, LONG* Enable, LONG* hour, LONG* min, LONG* voice, LONG* way, LONG* InerBellDelay, LONG* ExtBellDelay)

To query bell settings based on the specified weekday and bell index.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
weekDay	LONG	[in]	Weekday
Index	LONG	[in]	Bell index
Enable	LONG*	[out]	Whether to enable the bell
hour	LONG*	[out]	Hour
min	LONG*	[out]	Minute
voice	LONG*	[out]	Ringtone
way	LONG*	[out]	Ringing mode
InerBellDelay	LONG*	[out]	Internal ringing duration
ExtBellDelay	LONG*	[out]	External ringing duration

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success

False	BOOL	Function execution failure

See also

NMS_GetBellSchData

Attention

1. weekDay: specifies the weekday. The value 0 indicates Monday, 1 Tuesday, 2 Wednesday, 3 Thursday, 4 Friday, 5 Saturday, and 6 Sunday.

2. Index: specifies the bell index. The value range is 1-65535.

3. Enable: specifies whether to enable the bell. The value 0 indicates disabling the bell and 1 indicates enabling the bell.

4. voice: specifies the ringtone. The value ranges from 1 to 10, representing bell01.wav to bell10.wav respectively.

5. way: specifies the ringing mode. The value 0 indicates internal ringing, 1 external ringing, and 2 internal and external ringing.

6. InerBellDelay: specifies the internal ringing duration. The value ranges from 1 to 999, in seconds.

7. ExtBellDelay: specifies the external ringing duration. The value ranges from 1 to 999, in seconds.

Note

This interface is applicable to the new architecture firmware.

5.2.15.2 SetBellSchDataEx

VARIANT_BOOL SetBellSchDataEx(LONG dwMachineNumber, LONG weekDay, LONG Index, LONG Enable, LONG hour, LONG min, LONG voice, LONG way, LONG InerBellDelay, LONG ExtBellDelay)

To set bell information.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
weekDay	LONG	[in]	Weekday
Index	LONG	[in]	Bell index
Enable	LONG	[in]	Whether to enable the bell
hour	LONG	[in]	Hour

min	LONG	[in]	Minute
voice	LONG	[in]	Ringtone
way	LONG	[in]	Ringing mode
InerBellDelay	LONG	[in]	Internal ringing duration
ExtBellDelay	LONG	[in]	External ringing duration

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. weekDay: specifies the weekday. At most 7 days in a whole week can be set. Specify the weekday by using the least significant 7 bits of a byte. You can set multiple weekdays.
The first bit indicates Monday (0x00000001), the second bit Tuesday (0x00000002), third bit Wednesday (0x00000004), fourth bit Thursday (0x00000008), fifth bit Friday (0x00000010), sixth bit Saturday (0x00000020), and seventh bit Sunday (0x00000040).
To set multiple days, add the corresponding values. For example, to set Monday and Thursday, add 1 and 8, making 9. To set Tuesday, Friday and Saturday, add 2, 16 and 32, making 50.

2. Index: specifies the bell index. The value range is 1-65535.

3. Enable: specifies whether to enable the bell. The value 0 indicates disabling the bell and 1 indicates enabling the bell.

4. voice: specifies the ringtone. The value ranges from 1 to 10, representing bell01.wav to bell10.wav respectively.

5. way: specifies the ringing mode. The value 0 indicates internal ringing, 1 external ringing, and 2 internal and external ringing.

6. InerBellDelay: specifies the internal ringing duration. The value ranges from 1 to 999, in seconds.

7. ExtBellDelay: specifies the external ringing duration. The value ranges from 1 to 999, in seconds.

Note

This interface is applicable to the new architecture firmware.

5.2.15.3 GetDayBellSchCount

VARIANT_BOOL GetDayBellSchCount(LONG dwMachineNumber, LONG* DayBellCnt)

To query the number of bells that are set on the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
DayBellCnt	LONG*	[out]	Bell quantity

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

This interface is applicable to the new architecture firmware.

5.2.15.4 GetMaxBellIDInBellSchData

VARIANT_BOOL GetMaxBellIDInBellSchData(LONG dwMachineNumber, LONG* MaxBellID)

To obtain the maximum bell index among all bells.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
MaxBellID	LONG*	[out]	Weekday

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

NMS_GetMaxBellIDInBellSchData

Attention

Note

This interface is applicable to the new architecture firmware.

5.2.15.5 ReadAllBellSchData

VARIANT_BOOL ReadAllBellSchData(LONG dwMachineNumber)

To read the information about all bells to the SDK memory.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

This function needs to be used together with GetEachBellInfo to obtain bell information.

Note

This interface is applicable to the new architecture firmware.

5.2.15.6 GetEachBellInfo

VARIANT_BOOL GetEachBellInfo(LONG dwMachineNumber, LONG* weekDay, LONG* Index, LONG* Enable, LONG* hour, LONG* min, LONG* voice, LONG* way, LONG* InerBellDelay, LONG* ExtBellDelay)

To read bell information records from the SDK memory one by one.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
weekDay	LONG*	[out]	Weekday
Index	LONG*	[out]	Bell index
Enable	LONG*	[out]	Whether to enable the bell
hour	LONG*	[out]	Hour
min	LONG*	[out]	Minute
voice	LONG*	[out]	Ringtone
way	LONG*	[out]	Ringing mode
InerBellDelay	LONG*	[out]	Internal ringing duration
ExtBellDelay	LONG*	[out]	External ringing duration

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. This function needs to be used together with ReadAllBellSchData, which is used to read all

bell information to the memory.

2. weekDay: specifies the weekday. At most 7 days in a whole week can be set. Specify the weekday by using the least significant 7 bits of a byte. You can set multiple weekdays.
The first bit indicates Monday (0x00000001), the second bit Tuesday (0x00000002), third bit Wednesday (0x00000004), fourth bit Thursday (0x00000008), fifth bit Friday (0x00000010), sixth bit Saturday (0x00000020), and seventh bit Sunday (0x00000040).
You can learn the weekday settings by checking whether the corresponding bit is set. For example, if the returned value is 50, which is the sum of 0x00000002, 0x00000010 and 0x00000020, the bell is set to ring on Tuesday, Friday and Saturday.

3. Index: specifies the bell index. The value range is 1-65535.

4. Enable: specifies whether to enable the bell. The value 0 indicates disabling the bell and 1 indicates enabling the bell.

5. voice: specifies the ringtone. The value ranges from 1 to 10, representing bell01.wav to bell10.wav respectively.

6. way: specifies the ringing mode. The value 0 indicates internal ringing, 1 external ringing, and 2 internal and external ringing.

7. InerBellDelay: specifies the internal ringing duration. The value ranges from 1 to 999, in seconds.

8. ExtBellDelay: specifies the external ringing duration. The value ranges from 1 to 999, in seconds.

Note

This interface is applicable to the new architecture firmware.

5. 2. 16 UserValidDate Functions

5.2.16.1 SetUserValidDate

VARIANT_BOOL SetUserValidDate(LONG dwMachineNumber, BSTR UserID, LONG Expires, LONG ValidCount, BSTR StartDate, BSTR EndDate)

To set the validity period of a user account.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
UserID	BSTR	[in]	User ID
Expires	LONG	[in]	Validity period type

ValidCount	LONG	[in]	Number of valid usage times
StartDate	BSTR	[in]	Start time of the validity period, in the format of YYYY-MM-DD hh:mm:ss
EndDate	BSTR	[in]	End time of the validity period, in the format of YYYY-MM-DD hh:mm:ss

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. UserID: specifies the ID. The user ID is a string of English letters.

2. Expires: specifies the validity period type. The value ranges from 0 to 3. The value 0 indicates not limiting the the validity period of the user account, 1 indicates limiting the validity period by setting the start time and end time, 2 indicates limiting the validity period by setting the number of usage times, and 3 indicates limiting the validity period by setting the start time, end time, and number of usage times.

3. ValidCount: specifies the number of usage times of the user account. The value is larger than or equal to 0.

4. StartDate: specifies the start time of the validity period. Only the YYYY-MM-DD part is kept because time is accurate only to date on the firmware.

5. EndDate: specifies the end time of the validity period. Only the YYYY-MM-DD part is kept because time is accurate only to date on the firmware.

Note

This interface is applicable to the new architecture firmware.

5.2.16.2 GetUserValidDate

VARIANT_BOOL GetUserValidDate(LONG dwMachineNumber, BSTR UserID, LONG* Expires, LONG* ValidCount, BSTR* StartDate, BSTR* EndDate)

To query the validity period of a user account.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
UserID	BSTR	[in]	User ID
Expires	LONG*	[out]	Validity period type
ValidCount	LONG*	[out]	Number of valid usage times
StartDate	BSTR*	[out]	Start time of the validity period, in the format of YYYY-MM-DD hh:mm:ss
EndDate	BSTR*	[out]	End time of the validity period, in the format of YYYY-MM-DD hh:mm:ss

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. UserID: specifies the ID. The user ID is a string of English letters.

2. Expires: specifies the validity period type. The value ranges from 0 to 3. The value 0 indicates not limiting the the validity period of the user account, 1 indicates limiting the validity period by setting the start time and end time, 2 indicates limiting the validity period by setting the number of usage times, and 3 indicates limiting the validity period by setting the start time, end time, and number of usage times.

3. ValidCount: specifies the number of usage times of the user account. The value is larger than or equal to 0.

4. StartDate: specifies the start time of the validity period. Only the YYYY-MM-DD part is kept because time is accurate only to date on the firmware.

5. EndDate: specifies the end time of the validity period. Only the YYYY-MM-DD part is kept because time is accurate only to date on the firmware.

Note

This interface is applicable to the new architecture firmware.

5. 2. 17 Personalise Functions

5.2.17.1 UploadTheme

VARIANT_BOOL UploadTheme(LONG dwMachineNumber, BSTR FileName, BSTR InDevName)

To upload a theme picture to the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
FileName	BSTR	[in]	Save path + File name
InDevName	BSTR	[in]	File name in the device

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The FileName parameter specifies the file name and save path of the theme picture to be uploaded.

Note

This interface is applicable to the new architecture firmware.

5.2.17.2 UploadPicture

VARIANT_BOOL UploadPicture(LONG dwMachineNumber, BSTR FileName, BSTR InDevName)

To upload a background picture to the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
FileName	BSTR	[in]	Save path + File name
InDevName	BSTR	[in]	File name in the device

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The FileName parameter specifies the file name and save path of the background picture to be uploaded.

Note

This interface is applicable to the new architecture firmware.

5.2.17.3 DownloadPicture

VARIANT_BOOL DownloadPicture(LONG dwMachineNumber, BSTR FileName, BSTR FilePath)

To download the background picture from the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
FileName	BSTR	[in]	Name of the picture file

FilePath	BSTR	[in]	Path where the downloaded picture is to be saved

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

This interface is applicable to the new architecture firmware.

5. 2. 18 APP Info Functions

5.2.18.1 GetAllAppFun

VARIANT_BOOL GetAllAppFun(LONG dwMachineNumber, BSTR* AppName, BSTR* FunofAppName)

To query all App names, all App and corresponding Fun names.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
AppName	BSTR	[out]	App name list. The data is in .txt format and format name is [App name]. Records are separated with \r\n;
FunofAppName	BSTR	[out]	App and App functions lsit. The data is in .txt format and format name is [Fun name, Function name]. Records are separated with \r\n;

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

It is applicable to the new architecture machine

5.2.18.2 GetAllRole

VARIANT_BOOL GetAllRole(LONG dwMachineNumber, BSTR* RoleName)

To query all role names and the corresponding permission names.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
RoleName	BSTR	[out]	To query all role names and the corresponding permission names. The data is in .txt format and format name is [Role name, Permission name]. Records are separated with \r\n;

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

It is applicable to the new architecture machine

5.2.18.3 GetAppOfRole

VARIANT_BOOL GetAppOfRole(LONG dwMachineNumber, LONG Permission, BSTR* AppName)

To query all App names with specified role permission.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
Permission	LONG	[in]	Permisssion of specified role
AppName	BSTR	[out]	AppName with the specified role permission will be returned. The data is in .txt format and format name is [App name]. Records are separated with \r\n;

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

It is applicable to the new architecture machine

5.2.18.4 GetFunOfRole

VARIANT_BOOL GetFunOfRole(LONG dwMachineNumber, LONG Permission, BSTR* FunName)

To query all function names with the specified role permission.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
Permission	LONG	[in]	Permisssion of specified role
FunName	BSTR	[out]	FunName with the specified role permission will be returned. The data is in .txt format and format name is [Function name]. Records are separated with \r\n;

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

It is applicable to the new architecture machine

5.2.18.5 SetPermOfAppFun

VARIANT_BOOL SetPermOfAppFun(LONG dwMachineNumber, LONG Permission, BSTR AppName, BSTR FunName)

To set the corresponding permission of function and App meanwhile.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
Permission	LONG	[in]	Permission value to be set

AppName	BSTR	[in]	App name
FunName	BSTR	[in]	Fun name

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Function of FunName must be included in the functions of AppName

Note

It is applicable to the new architecture machine

5.2.18.6 DeletePermOfAppFun

VARIANT_BOOL DeletePermOfAppFun(LONG dwMachineNumber, LONG Permission, BSTR AppName, BSTR FunName)

To delete the corresponding permission of function and App meanwhile.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
Permission	LONG	[in]	Permission value to be deleted
AppName	BSTR	[in]	App name
FunName	BSTR	[in]	Fun name

Returns

Value description:

name	type	description of value

True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Function of FunName must be included in the functions of AppName

Note

It is applicable to the new architecture machine

5.2.18.7 IsUserDefRoleEnable

VARIANT_BOOL IsUserDefRoleEnable(LONG dwMachineNumber, LONG Permission, VARIANT_BOOL* Enable)

To judge whether the user defined role is enabled.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
Permission	LONG	[in]	Permission value to be deleted
Enable	BOOL*	[out]	Whether the user defined role is enabled: 1 means enable and 0 means disable

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

It is applicable to the new architecture machine

5. 2. 19 Template Integration Functions

5.2.19.1 SSR_SetDeviceData

VARIANT_BOOL SSR_SetDeviceData(LONG dwMachineNumber, BSTR TableName, BSTR Datas, BSTR Options)

This function is applicable to an attendance machine on which the new firmware supports the PULL protocol. This function is used to set data, including the time segment, user information, and holiday settings. The data can be one or more records. If the primary key of an inserted record already exists, the original record will be overwritten.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
TableName	BSTR	[in]	Data table name. For details about the available tables, see attachment 1 PULL Data Dictionary for the New Firmware.
Datas	BSTR	[in]	Data records. The data is in .txt format. Records are separated with \r\n, and fields with values are separated with \t.
Options	BSTR	[in]	It is left blank by default and for extension.

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

This interface is applicable to the new architecture firmware.

5.2.19.2 SSR_GetDeviceData

VARIANT_BOOL SSR_GetDeviceData(LONG dwMachineNumber, BSTR* Buffer, LONG BufferSize, BSTR TableName, BSTR FiledNames, BSTR Filter, BSTR Options)

This function is applicable to an attendance machine on which the new firmware supports the PULL protocol. This function is used to read data from the machine, including the punch records, time segments, user information, and holiday settings. The data can be one or more records.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
Buffer	BSR*	[out]	Buffer for storing returned data. The returned data is in .txt format and may contain multiple records, which are separated with \r\n.
BufferSize	LONG	[in]	Size of the buffer for storing returned data
TableName	BSTR	[in]	Data table name. For details about the available tables, see attachment 1 PULL Data Dictionary for the New Firmware.
FiledNames	BSTR	[in]	Field name list. The fields are separated with \t. * indicates all fields. The field names are on the first line in the returned data.
Filter	BSTR	[in]	Filter criteria for reading data. If the data is a character string in the format of "field name operator value", multiple filter criteria separated with a comma (,) are supported. The details are as follows:
Options	BSTR	[in]	Options

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

This interface is applicable to the new architecture firmware.

5.2.19.3 SSR_GetDeviceDataCount

VARIANT_BOOL SSR_GetDeviceDataCount(BSTR TableName, BSTR Filter, BSTR Options)

Query the number of The unification of multiple Biometric Templates within the device.

Parameters

Parameters are shown below:

name	type	param direction	description of param
TableName	BSTR	[in]	Name of Table
Filter	BSTR	[in]	Filter condition
Options	BSTR	[in]	parameter

Returns

returned value specification:

name	type	description of value
True	BOOL	Success
False	BOOL	Fault

See also

Attention

Note

This interface is applicable to BW,TFT devices,For The unification of multiple Biometric Template Tablestructure, see the SSR_SetDeviceData interface Attention

5.2.19.4 SSR_DeleteDeviceData

VARIANT_BOOL SSR_DeleteDeviceData(LONG dwMachineNumber, BSTR TableName, BSTR Datas, BSTR Options)

Delete The unification of multiple Biometric Templates within the device.

Parameters

Parameters are shown below:

name	type	param direction	description of param
TableName	BSTR	[in]	Name of Table
Datas	BSTR	[in]	Filter condition
Options	BSTR	[in]	parameter

Returns

returned value specification:

name	type	description of value
True	BOOL	Success
False	BOOL	Fault

See also

Attention

Note

This interface is applicable to BW,TFT devices,For The unification of multiple Biometric Template Tablestructure, see the SSR_SetDeviceData interface Attention

5.2.19.5 Variable description : BiometricType

BiometricType.

Parameters

BiometricType Get the supported biometric type by retrieving the parameter and return 8-bit numeric string. Each represents a type of biometric type. (0 not support ; 1 support;e.g.:BiometricType=01100000 ,device support FP and Face)

Character index	0	1	2	3	4	5	6	7	8

Related Type	General	FP	Face	Voice	Iris	Retina	Palmprint	FingerVein	Palmvein

See also

Attention

Note

Applicable to interface of unification of multiple Biometric Template

5.2.19.6 Variable description : BiometricVersion

BiometricVersion.

Parameters

BiometricVersion Get the version of the supported biometric type Get the version of the supported biometric type of the device by retrieving the parameter and return the version of the supported biometric type seperated by ":" (e.g.:BiometricVersion=0:10.0:7.0:::::device support FP10.0 and Face7.0)

Character index	0	1	2	3	4	5	6	7	8
Related Type	General	FP	Face	Voice	Iris	Retina	Palmprint	FingerVein	Palmvein

See also

Attention

TableName

Note

Applicable to interface of unification of multiple Biometric Template

5.2.19.7 Variable description : BiometricMaxCount

BiometricMaxCount.

Parameters

BiometricMaxCount Get the supported biometric type data capacity by retrieving the parameter and return the supported biometric type data capacity seperated by ":" (e.g.:BiometricMaxCount=0:3000:1000:::::,device support 3000 FP and 1000 Face templates)

Character index	0	1	2	3	4	5	6	7	8
Related Type	General	FP	Face	Voice	Iris	Retina	Palmprint	FingerVein	Palmvein

The new device is directly returned by the firmware, and the old firmware is SDK compatible.(supported)

See also

Attention

TableName

Note

Applicable to interface of unification of multiple Biometric Template

5.2.19.8 Variable description : BiometricUsedCount

BiometricUsedCount.

Parameters

BiometricUsedCount Get the number of supported biometric type data by retrieving the padameter and return the number of supported biometric type data seperated by ":" (e.g.:BiometricUsedCount=0:100:10:::::,indicate there are 100 FP templates, 10 faces)

Character index	0	1	2	3	4	5	6	7	8
Related Type	General	FP	Face	Voice	Iris	Retina	Palmprint	FingerVein	Palmvein

See also

Attention

TableName

Note

Applicable to interface of unification of multiple Biometric Template

5.3 Access Control Functions(Time Slot, Group, Open Door Combination)

5. 3. 1 GetUserGroup

VARIANT_BOOL GetUserGroup(LONG dwMachineNumber, LONG dwEnrollNumber, LONG* UserGrp)

To obtain the ID of the group to which a specified user belongs.

Parameters

Parameter description:

name	type	param direction	description of

			param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	User ID
UserGrp	LONG*	[out]	Returned group ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The value of UserGrp is the ID of the group to which the user specified by dwEnrollNumber belongs. The value range is 1-5.

Note

Applicable to BW, TFT, and IFACE devices

5. 3. 2 SetUserGroup

VARIANT_BOOL SetUserGroup(LONG dwMachineNumber, LONG dwEnrollNumber, LONG UserGrp)

To set the group to which a specified user belongs.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

dwEnrollNumber	LONG	[in]	User ID
UserGrpl	LONG	[in]	Group ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The value of UserGrp is the ID of the group to which the user specified by dwEnrollNumber belongs. The value range is 1-5.

Note

Applicable to BW, TFT, and IFACE devices

5. 3. 3 GetTZInfo

VARIANT_BOOL GetTZInfo(LONG dwMachineNumber, LONG TZIndex, BSTR* TZ)

To obtain the information about a time segment with a specified index.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
TZIndex	LONG	[in]	Time segment index
TZ	BSTR*	[out]	The value is the information about the time segment with the index specified by TZIndex.

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The value of the TZ parameter is the information about the time segment with the index specified by TZIndex. Every eight digits indicate a time segment in the format of hhmmhhmm. For example, 10111223000023590000235900002359000023590000235900002359 indicates the time segment from 10:11 to 12:23 on Sunday and the whole day from Monday to Saturday.

Note

Applicable to BW, TFT, and IFACE devices

5. 3. 4 SetTZInfo

VARIANT_BOOL SetTZInfo(LONG dwMachineNumber, LONG TZIndex, BSTR TZ)

To set the information about a time segment with a specified index.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
TZIndex	LONG	[in]	Time segment index
TZ	BSTR	[in]	Time segment information to be set

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success

False	BOOL	Function execution failure

See also

Attention

The TZ parameter specifies the time segment information to be set. Every eight digits indicate a time segment in the format of hhmmhhmm.For example, 10111223000023590000235900002359 000023590000235900002359 indicates the time segment from 10:11 to 12:23 on Sunday and the whole day from Monday to Saturday.

Note

Applicable to BW, TFT, and IFACE devices

5. 3. 5 GetUnlockGroups

VARIANT_BOOL GetUnlockGroups(LONG dwMachineNumber, BSTR* Grps)

To obtain unlock combinations of the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
Grps	BSTR*	[out]	Current unlock combination of the machine

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The value of the Grps parameter is an unlock combination. There are totally 10 unlock combinations, separated with a colon (:). For example, 12:23:14:15 represents four valid combinations, which are combination 1 (12 represents groups 1 and 2), combination 2 (23 represents groups 2 and 3), combination 3 (14 represents groups 1 and 4), and combination 4 (15 represents groups 1 and 5).

Note

Applicable to BW

5. 3. 6 SetUnlockGroups

VARIANT_BOOL SetUnlockGroups(LONG dwMachineNumber, BSTR Grps)

To set unlock combinations.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
Grps	BSTR	[in]	Unlock combination

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Totally 10 unlock combinations need to be set, and separate these combinations with a colon (:). For example, 12:23:14:15 represents four valid combinations, which are combination 1 (12 represents groups 1 and 2), combination 2 (23 represents groups 2 and 3), combination 3 (14 represents groups 1 and 4), and combination 4 (15 represents groups 1 and 5).

Note

Applicable to BW

5.3.7 SSR_SetUnLockGroup

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

To set unlock combinations.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
CombNo	LONG	[in]	Unlock combination ID ranging from 1 to 10. The machine supports at most 10 unlock combinations.
Group1	LONG	[in]	Unlock combination group 1
Group2	LONG	[in]	Unlock combination group 2
Group3	LONG	[in]	Unlock combination group 3
Group4	LONG	[in]	Unlock combination group 4
Group5	LONG	[in]	Unlock combination group 5

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Group1, Group2, Group3, Group4 and Group5 are group IDs of an unlock combination. Each unlock combination contains five group IDs. The group ID ranges from 1 to 99. For example, SSR_SetUnLockGroup(1,1,2,23,14,0,56) indicates that the personnel of groups 2, 23, 14 and 56 need to verify together to open the door.

Note

Applicable to TFT and IFACE devices

5. 3. 8 SSR_GetUnLockGroup

VARIANT_BOOL SSR_GetUnLockGroup(LONG dwMachineNumber, LONG CombNo, LONG* Group1, LONG* Group2, LONG* Group3, LONG* Group4, LONG* Group5)

To obtain unlock combination information based on the group ID.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
CombNo	LONG	[in]	Unlock combination ID ranging from 1 to 10
Group1	LONG*	[out]	Unlock combination group 1
Group2	LONG*	[out]	Unlock combination group 2
Group3	LONG*	[out]	Unlock combination group 3
Group4	LONG*	[out]	Unlock combination group 4
Group5	LONG*	[out]	Unlock combination group 5

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success

False	BOOL	Function execution failure

See also

Attention

The values of Group1, Group2, Group3, Group4, and Group5 are group IDs of the specified unlock combination. The group IDs are returned. Each unlock combination contains a maximum of five groups. The group ID ranges from 1 to 99.

Note

Applicable to TFT and IFACE devices

5.3.9 GetGroupTZs

VARIANT_BOOL GetGroupTZs(LONG dwMachineNumber, LONG GroupIndex, LONG* TZs)

To obtain the time segments of a specified group. This function differs from GetGroupTZStr in the format of the returned values.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
GroupIndex	LONG	[in]	Group index, ranging from 1 to 5
TZs	LONG*	[out]	Time segment index

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The TZs parameter is a LONG pointer. Its value is the indexes of the three time segments used by the group specified by GroupIndex, indicated by TZs[0], TZs[1] and TZs[2] respectively.

Note

Applicable to BW

5. 3. 10 SetGroupTZs

VARIANT_BOOL SetGroupTZs(LONG dwMachineNumber, LONG GroupIndex, LONG* TZs)

To set the three time segments of a specified group. This function differs from SetGroupTZStr in the format of the specified values.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
GroupIndex	LONG	[in]	Group index, ranging from 1 to 5
TZs	LONG*	[in]	Time segment index

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The TZs parameter is a LONG pointer. The three time segments are specified by TZs[0], TZs[1], and TZs[2] respectively.

Note

Applicable to BW

5. 3. 11 GetGroupTZStr

VARIANT_BOOL GetGroupTZStr(LONG dwMachineNumber, LONG GroupIndex,BSTR* TZs)

To obtain the time segments of a specified group. This function differs from GetGroupTZs in the format of the returned values.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
GroupIndex	LONG	[in]	Group index, ranging from 1 to 5
TZs	BSTR*	[out]	Time segment index

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The value of TZs is the indexes of the time segments of the group specified by GroupIndex. Each group contains three time segments separated with a colon (:). For example, if the returned value is 1:23:13, the indexes of the three time segments are 1, 23 and 13 respectively.

Note

Applicable to BW

5. 3. 12 SetGroupTZStr

VARIANT_BOOL SetGroupTZStr(LONG dwMachineNumber, LONG GroupIndex, BSTR TZs)

To set the time segments of a specified group. This function differs from SetGroupTZs in the format of the specified values.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
GroupIndex	LONG	[in]	Group index, ranging from 1 to 5
TZs	BSTR	[in]	Time segment index

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The TZs parameter specifies the indexes of the time segments of the group described by GroupIndex. Each group contains three time segments separated with a colon (:). For example, if the value is set to 1:23:13, the indexes of the three time segments are 1, 23 and 13 respectively.

Note

Applicable to BW

5. 3. 13 SSR_SetGroupTZ

VARIANT_BOOL SSR_SetGroupTZ(LONG dwMachineNumber, LONG GroupNo, LONG Tz1, LONG Tz2, LONG Tz3, LONG ValidHoliday, LONG VerifyStyle)

To set the time segments of a group.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
GroupNo	LONG	[in]	Group ID, ranging from 1 to 99
Tz1	LONG	[in]	Time segment 1
Tz2	LONG	[in]	Time segment 2
Tz3	LONG	[in]	Time segment 3
ValidHoliday	LONG	[in]	Whether the time segments are valid on holidays
VerifyStyle	LONG	[in]	Group verification mode

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The VerifyStyle parameter specifies the group verification mode. 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).

Note

Applicable to TFT and IFACE devices

5. 3. 14 SSR_GetGroupTZ

VARIANT_BOOL SSR_GetGroupTZ(LONG dwMachineNumber, LONG GroupNo, LONG Tz1, LONG Tz2, LONG Tz3, LONG ValidHoliday, LONG VerifyStyle)

To obtain the time segments of a group.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Fingerprint machine ID
GroupNo	LONG	[in]	Group ID, ranging from 1 to 99
Tz1	LONG*	[out]	The indexes of the three time segments of the specified group are returned. The index ranges from 1 to 50.
Tz2	LONG*	[out]	The indexes of the three time segments of the specified group are returned. The index ranges from 1 to 50.
Tz3	LONG*	[out]	The indexes of the three time segments of the specified group are returned. The index ranges from 1 to 50.
ValidHoliday	LONG*	[out]	Whether the time segments are valid on holidays. The value 1 indicates that the time segments are valid on holidays and 0 indicates that the time segments are invalid on holidays.
VerifyStyle	LONG*	[out]	Verification mode of the fingerprint machine

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The VerifyStyle parameter specifies the group verification mode. 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).

Note

Applicable to TFT and IFACE devices

5. 3. 15 GetUserTZs

VARIANT_BOOL GetUserTZs(LONG dwMachineNumber, LONG UserID, LONG TZs)

To obtain the time segments of a user. Each user has three time segments. This function differs from GetUserTZStr in the format of the returned time segments.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
UserID	LONG	[in]	User ID
TZs	LONG*	[out]	Time segments during which the user can open the door

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The TZs pointer has three values that store three time segment indexes, which can be read from TZs[0], TZs[1] and TZs[2] respectively.

Note

Applicable to BW, TFT, and IFACE devices

5. 3. 16 SetUserTZs

VARIANT_BOOL SetUserTZs(LONG dwMachineNumber, LONG UserID, LONG* TZs)

To set the time segments of a user. At most three time segments can be set for a user. This function differs from SetUserTZStr in the format of specified time segments.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
UserID	LONG	[in]	User ID
TZs	LONG*	[in]	Time segment index

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The TZs parameter specifies the indexes of the time segments of the user described by UserID. The three indexes can be specified by TZs[0], TZs[1] and TZs[2] respectively. When TZs[0] is 0, the group setting takes effect. When it is 1, a user-defined setting takes effect.

Note

Applicable to BW, TFT, and IFACE devices

5. 3. 17 GetUserTZStr

VARIANT_BOOL GetUserTZStr(LONG dwMachineNumber, LONG UserID, BSTR* TZs)

To obtain the time segments of a user. This function differs from GetUserTZs in the format of returned time segments.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
UserID	LONG	[in]	User ID
TZs	BSTR*	[out]	Unlock time segments of the user

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The TZs parameter specifies the unlock time segments of the user. The details are as follows: Monochrome access control panel: X1:X2:X3. X1, X2 and X3 are indexes of user-defined time segments. If X1 is 0, the group time segments are used. To query whether a user uses the group time segments, invoke the UseGroupTimeZone function and check the returned value. If user A uses user-defined time segments 1 and 2, the fingerprint machine will return 1:2:0. If user B uses the group time segments, the fingerprint machine will return 0:0:0.

Color-screen access control panel: X1:X2:X3:X4. X4 specifies whether to use the group time segments. The value 0 indicates using the group time segments and 1 indicates using user-defined time segments. X1, X2 and X3 are indexes of used time segments. For example, if user A uses user-defined time segments 1 and 2, the fingerprint machine will return 1:2:0:1. If user B uses the group time segments defined as 1:1:1:0, the fingerprint machine will return 0:0:0:0.

Note

Applicable to BW, TFT and IFACE devices

5.3.18 SetUserTZStr

VARIANT_BOOL SetUserTZStr(LONG dwMachineNumber, LONG UserID, BSTR TZs)

To set the time segments of a user. The time segments are separated with a colon (:). This function differs from SetUserTZs in the format of specified time segments.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
UserID	LONG	[in]	User ID
TZs	BSTR	[in]	Unlock time segments of the user

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The TZs parameter specifies the unlock time segments of the user. The details are as follows: Monochrome access control panel: X1:X2:X3. X1, X2 and X3 are indexes of user-defined time segments. If X1 is 0, the group time segments are used. To query whether a user uses the group time segments, invoke the UseGroupTimeZone function and check the returned value. If user A uses user-defined time segments 1 and 2, the fingerprint machine will return 1:2:0. If user B uses the group time segments, the fingerprint machine will return 0:0:0.Color-screen access control panel: X1:X2:X3:X4. X4 specifies whether to use the group time segments. The value 0 indicates using the group time segments and 1 indicates using user-defined time segments. X1, X2 and X3 are indexes of used time segments. For example, if user A uses user-defined time segments 1 and 2, the fingerprint machine will return 1:2:0:1. If user B uses the group time segments defined as 1:1:1:0, the fingerprint machine will return 0:0:0:0.

Note

Applicable to BW, TFT and IFACE devices

5. 3. 19 ACUnlock

VARIANT_BOOL ACUnlock(LONG dwMachineNumber, LONG Delay)

To enable the access control panel to output door opening level and close the door after Delay/10 seconds.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
Delay	LONG	[in]	Delay in closing the door

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 3. 20 GetACFun

VARIANT_BOOL GetACFun(LONG* ACFun)

To check whether the machine has the access control function.

Parameters

Parameter description:

name	type	param direction	description of param
ACFun	LONG*	[out]	Flag of the access control function

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The ACFun parameter is an access control flag. The value 0 indicates that access control is unavailable, 1 indicates simple access control, 2 indicates medium-level access control, 6 indicates advanced access control, and 14 indicates advanced access control+normal open.

Note

Applicable to BW, TFT and IFACE devices

5. 3. 21 GetDoorState

VARIANT_BOOL GetDoorState(LONG MachineNumber, LONG* State)

To query the current door status. The value 1 indicates that the door is opened and 0 indicates that the door is closed.

Parameters

Parameter description:

name	type	param direction	description of param
MachineNumber	LONG	[in]	Machine ID
State	LONG*	[out]	Door status

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 3. 22 UseGroupTimeZone

VARIANT_BOOL UseGroupTimeZone()

To query whether a user uses the group time segments.

Parameters

none

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

This function must be used together with GetUserTZs or GetUserTZStr. Specifically, invoke the GetUserTZs or GetUserTZStr function to obtain the time segment information about a user, and then invoke UseGroupTimeZone to check whether the user uses the group time segments.

Note

Applicable to BW, TFT and IFACE devices

5. 3. 23 TurnOffAlarm

VARIANT_BOOL TurnOffAlarm(LONG dwMachineNumber)

To turn off alarms.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

This interface is applicable to the new architecture firmware.

5.4 Device Management Functions

5. 4. 1 IsTFTMachine

VARIANT_BOOL IsTFTMachine(LONG dwMachineNumber)

To check whether the machine is a color-screen one.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value

True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.4.2 GetDeviceStatus

VARIANT_BOOL GetDeviceStatus(LONG dwMachineNumber, LONG dwStatus, LONG* dwValue)

To query the data storage status on the machine, such as the number of administrators and number of users.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwStatus	LONG	[in]	Data to be obtained
dwValue	LONG*	[out]	Content of the data specified by dwStatus

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The dwStatus parameter specifies the data to be obtained. The value range is 1-22. The values are described as follows:

1 Number of administrators
2 Number of registered users
3 Number of fingerprint templates on the machine
4 Number of passwords
5 Number of operation records
6 Number of attendance records
7 Fingerprint template capacity
8 User capacity
9 Attendance record capacity
10 Remaining fingerprint template capacity
11 Remaining user capacity
12 Remaining attendance record capacity
21 Number of faces
22 Face capacity
0 Other conditions

Note

Applicable to BW, TFT, and IFACE devices

5. 4. 3 GetDeviceInfo

VARIANT_BOOL GetDeviceInfo(LONG dwMachineNumber, LONG dwInfo, LONG* dwValue)

To obtain machine information, such as the language and baud rate.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwInfo	LONG	[in]	Information type
dwValue	LONG*	[out]	Informatin of the type specified by dwInfo

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

dwInfo: specifies the type of information to be obtained. The value ranges from 1 to 68, and cannot be 65. The values are described as follows:

1. Maximum number of administrators, which is fixed at 500.
2. Machine ID.
3. Language: If the value of dwValue is 0, the language is English. The value 1 indicates other conditions. The value 2 indicates Traditional Chinese, and 3 indicates Thai.
4. Idle duration (in minutes): After the specified idle duration elapses, the machine will enter the standby state or be shut down.
5. Lock control duration, that is, the lock drive duration.
6. Attendance record quantity alarm: When the specified attendance record quantity is reached, the machine will raise an alarm to remind the user.
7. Operation record quantity alarm: When the specified operation record quantity is reached, the machine will raise an alarm to remind the user.
8. Duplicate record time: minimum interval for a user to record the same attendance state.
9. Baud rate for RS232/485 communication: The value 0 indicates the baud rate of 1200 bps, 1 indicates 2400 bps, 2 indicates 4800 bps, 3 indicates 9600 bps, 4 indicates 19200 bps, 5 indicates 38400 bps, 6 indicates 57600 bps, and Others indicates 115200 bps.
10. Parity check bit, return value of which is fixed at 0.
11. Stop bit, return value of which is fixed at 0.
12. Date separator, return value of which is fixed at 1.
13. Whether to enable network functions: The value 1 indicates enabling network functions and 0 indicates disabling network functions.
14. Whether to enable RS232.
15. Whether to enable RS485.
16. Whether to enable announcements.
17. Whether to perform high-speed comparison.

18. Idle mode, that is, the state of the machine during idle hours. The value 87 indicates shutdown and 88 indicates hibernation.
19. Automatic shutdown time: The default value is 255, which means that the machine will not automatically shut down.
20. Automatic startup time: The default value is 55, which means that the machine will not automatically start.
21. Automatic hibernation time: The default value is 255, which means that the machine will not automatically enter the hibernation state.
22. Automatic ringing time 1: The default value is 65535, which means that the bell will not automatically ring.
23. 1:N comparison threshold.
24. Registration threshold.
25. 1:1 comparison threshold.
26. Whether to display the matching score during verification.
27. Number of people that unlock the door concurrently.
28. Verify the card number only.
29. Network speed: The value 1 indicates 100M-H, 4 indicates 10M-F, 5 indicates 100M-F, 8 indicates AUTO, and Others indicates 10M-H.
30. Whether a card must be registered.
31. Waiting time before the machine automatically returns to the initial state if no operation is performed.
32. Waiting time before the machine automatically returns to the initial state if no response is returned after the PIN is input.
33. Waiting time before the machine automatically returns to the initial state if no operation is performed after entering the menu.
34. Time format.
35. Whether 1:1 comparison is mandatory.
    36 40. Automatic ringing time 2, 3, 4, 5, and 6: The default value is 65535, which means that the bell will not automatically ring.
    41 56. Automatic status change time 1 16: The default values are all -1, which means that the status will not change automatically.
36. Wiegand failure ID.
37. Wiegand duress ID.
38. Wiegand zone bit.
39. Pulse width of Wiegand outputs.
40. Pulse interval of Wiegand outputs.
41. ID of the start sector on the MIFARE card where fingerprints are stored.
42. Total number of sectors on the MIFARE card where fingerprints are stored.
43. Number of fingerprints stored on the MIFARE card.

44. Whether to display the attendance status.
    67-68. Meaningless.
45. TCP Comm Port.
46. UDP port.
47. Fingerprint algorithm version.
48. Face algorithm version.
49. Finger vein version
50. FaceFunOn.
51. PIN2Width.
52. IsSupportABCPin.
53. IMEFunOn.
54. IsSupportAlarmExt.
55. ~DCTZ.
56. ~DOTZ.
57. dwValue serves the input and output parameter. The input indicates the name of another option to be obtained and the output is the value of this option. This function is similar to GetSysOption in this case.
    [Note] The values of the preceding time points are all digits, which can be converted into actual time points. Specifically, convert a value into binary format, the least significant eight bits indicate the minute and the most significant eight bits indicate the hour. For example, if the value is 2860, it is 101100101100 in binary, the least significant eight bits are 00101100, which is 44 in decimal, and the most significant eight bits are 00001011, which is 11 in decimal. That is, the actual time point is 11:44.

Note

Applicable to BW, TFT, and IFACE devices

5. 4. 4 SetDeviceInfo

VARIANT_BOOL SetDeviceInfo(LONG dwMachineNumber, LONG dwInfo, LONG dwValue)

To set machine information, such as the language and duplicate record time.

Parameters

Parameter description:

name	type	param direction	description of param

dwMachineNumber	LONG	[in]	Machine ID
dwInfo	LONG	[in]	Information type, ranging from 1-20, 80, 81. For details about the meanings of values, see the description of GetDeviceInfo
dwValue	LONG	[in]	Information of the type specified by dwInfo

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

dwInfo specifies the information type, ranging from 1-20, 80, 81. For details about the meanings of values, see the description of GetDeviceInfo.

Note

Applicable to BW, TFT and IFACE devices

5. 4. 5 SetDeviceTime

VARIANT_BOOL SetDeviceTime(LONG dwMachineNumber)

To set the time of the machine to be the same as that of the local computer. To set specified time, see the description of SetDeviceTime2.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 4. 6 SetDeviceTime2

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

To set the time of the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwYear	LONG	[in]	Year
dwMonth	LONG	[in]	Month
dwDay	LONG	[in]	Date
dwHour	LONG	[in]	Hour
dwMinute	LONG	[in]	Minute
dwSecond	LONG	[in]	Second

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 4. 7 GetDeviceTime

VARIANT_BOOL GetDeviceTime(LONG dwMachineNumber, LONG* dwYear, LONG* dwMonth, LONG* dwDay, LONG* dwHour, LONG* dwMinute, LONG* dwSecond)

To query the time of the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwYear	LONG*	[out]	Year
dwMonth	LONG*	[out]	Month
dwDay	LONG*	[out]	Date
dwHour	LONG*	[out]	Hour
dwMinute	LONG*	[out]	Minute
dwSecond	LONG*	[out]	Second

Returns

Value description:

name	type	description of value

True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.4.8 GetSerialNumber

VARIANT_BOOL GetSerialNumber(LONG dwMachineNumber, BSTR* dwSerialNumber)

To query the serial number of the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwSerialNumber	BSTR*	[out]	Serial number

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT, and IFACE devices

5.4.9 GetProductCode

VARIANT_BOOL GetProductCode(LONG dwMachineNumber, BSTR* lpszProductCode)

To query the product code of the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
lpszProductCode	BSTR*	[out]	Product code

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 4. 10 GetFirmwareVersion

VARIANT_BOOL GetFirmwareVersion(LONG dwMachineNumber, BSTR* strVersion)

To query the firmware version of the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
strVersion	BSTR*	[out]	Firmware version of the machine

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 4. 11 GetSDKVersion

VARIANT_BOOL GetSDKVersion(BSTR* strVersion)

To query the SDK version.

Parameters

Parameter description:

name	type	param direction	description of param
strVersion	BSTR*	[out]	SDK version

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 4. 12 GetDeviceIP

VARIANT_BOOL GetDeviceIP(LONG dwMachineNumber, BSTR* IPAddr)

To query the IP address of the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
IPAddr	BSTR*	[out]	IP address

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 4. 13 SetDeviceIP

VARIANT_BOOL SetDeviceIP(LONG dwMachineNumber, BSTR IPAddr)

To set the IP address of the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

IPAddr	BSTR	[in]	IP address

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 4. 14 GetDeviceMAC

VARIANT_BOOL GetDeviceMAC(LONG dwMachineNumber, BSTR* sMAC)

To query the MAC address of the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
sMAC	BSTR*	[out]	MAC address

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 4. 15 SetDeviceMAC

VARIANT_BOOL SetDeviceMAC(LONG dwMachineNumber, BSTR sMAC)

To set the MAC address of the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
sMAC	BSTR	[in]	MAC address

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 4. 16 GetWiegandFmt

VARIANT_BOOL GetWiegandFmt(LONG dwMachineNumber, BSTR* sWiegandFmt)

To query the Wiegand format of the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
sWiegandFmt	BSTR*	[out]	Wiegand format

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 4. 17 SetWiegandFmt

VARIANT_BOOL SetWiegandFmt(LONG dwMachineNumber, BSTR sWiegandFmt)

To set the Wiegand format of the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
sWiegandFmt	BSTR	[in]	Wiegand format

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 4. 18 GetCardFun

VARIANT_BOOL GetCardFun(LONG dwMachineNumber, LONG* CardFun)

To query whether the machine supports the RF card.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
CardFun	LONG*	[in]	Whether the RF card is supported

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

If the value of CardFun is 1, the machine supports only the RF card. If the value is 2, both the RF card and fingerprints are supported. If the value is 0, the RF card is not supported.

Note

Applicable to BW, TFT and IFACE devices

5. 4. 19 SetDeviceCommPwd

VARIANT_BOOL SetDeviceCommPwd(LONG dwMachineNumber, LONG CommKey)

To set the communication password of the machine, which will be saved on the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
CommKey	LONG	[in]	Communication password

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 4. 20 SetCommPassword

VARIANT_BOOL SetCommPassword(LONG CommKey)

To set the communication password of the PC. A connection can be set up between the machine and the PC only if their communication passwords are the same.

Parameters

Parameter description:

name	type	param direction	description of param
CommKey	LONG	[in]	Communication password

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 4. 21 QueryState

VARIANT_BOOL QueryState(LONG* State)

To query the current status of the machine.

Parameters

Parameter description:

name	type	param direction	description of param
State	LONG*	[out]	Current status of the machine

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

State specifies the current status of the machine. The meanings of the values are as follows:
0 Waiting state
1 Fingerprint registration state
2 Fingerprint identification state
3 Menu access state
4 Busy state (handling other work)
5 State of waiting for card writing

Note

Applicable to BW, TFT and IFACE devices

5. 4. 22 GetVendor

VARIANT_BOOL GetVendor(BSTR* strVendor)

To query the manufacturer of the machine.

Parameters

Parameter description:

name	type	param direction	description of param
strVendor	BSTR*	[in]	Manufacturer of the machine

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 4. 23 GetDeviceStrInfo

VARIANT_BOOL GetDeviceStrInfo(LONG dwMachineNumber, LONG dwInfo, BSTR* Value)

To query the delivery time of the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwInfo	LONG	[in]	It can be set only to 1
Value	BSTR*	[out]	Delivery time of the machine

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

dwInfo can be set only to 1.

Note

Applicable to BW, TFT and IFACE devices

5. 4. 24 GetPlatform

VARIANT_BOOL GetPlatform(LONG dwMachineNumber, BSTR* Platform)

To query the platform of the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
Platform	BSTR*	[out]	Platform name

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 4. 25 ReadAOptions

VARIANT_BOOL ReadAOptions (LONG dwMachineNumber, BSTR Option, BSTR Value)

Read the values of specified configuration parameters from the device. The parameters beginning with "~" are skipped.

Parameter

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine equipment number
Option	BSTR	[in]	parameter name
Value	BSTR	[in]	Parameter value

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success

False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT devices

5. 4. 26 GetSysOption

VARIANT_BOOL GetSysOption (LONG dwMachineNumber, BSTR Option, BSTR Value)

Get parameter values

Parameter

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine equipment number
Option	BSTR	[in]	parameter name
Value	BSTR	[in]	Parameter value

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Parameter name. When the parameter is the character string "~ZKFPVersion", if the returned Value is 10, the current device uses ZKFinger10.0; if the returned Value is null or 9, the current device uses ZKFinger9.0.

Note

Applicable to BW, TFT devices

5. 4. 27 SetSysOption

VARIANT_BOOL SetSysOption(LONG dwMachineNumber, BSTR Option, BSTR Value)

To set parameter value.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
Option	BSTR	[in]	Parameter name
Value	BSTR	[in]	Parameter value

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW and TFT devices

5. 4. 28 GetDeviceStatusEx

LONG GetDeviceStatusEx(LONG dwMachineNumber)

To get the status of the P2P devices.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
1	LONG	Machine is reading data
2	LONG	Machine is writing data
3	LONG	Machine is working
greater than 3	LONG	The machine is free and the software can operate the machine

See also

Attention

P2P devices could not do multi-threaded work. Must get the device status and when the device isfree the software can operate the machine.

Note

Applicable to some P2P devices, such as the K Pro series attendace machine

5.5 Others

5. 5. 1 Device Control Functions

5.5.1.1 ClearAdministrators

VARIANT_BOOL ClearAdministrators(LONG dwMachineNumber)

To clear the operation rights of all administrators.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.5.1.2 EnableDevice

VARIANT_BOOL EnableDevice(LONG dwMachineNumber, VARIANT_BOOL bFlag)

To enable or disable the machine. After the machine is disabled, the fingerprint, keyboard, and card modules are unavailable.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
bFlag	BOOL	[in]	Flag that indicates whether the machine is enabled. The value 1 indicates that the machine is enabled and 0 indicates that the machine is disabled.

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.5.1.3 EnableClock

VARIANT_BOOL EnableClock(LONG Enabled)

To enable or disable the display of the colon (:) in the time of the machine. The colon is displayed on the main screen of the machine if enabled, and not displayed if disabled.

Parameters

Parameter description:

name	type	param direction	description of param
Enabled	LONG	[in]	Whether to enable the display of the colon. The value 1 indicates enabling the display of the colon and 0 indicates disabling the display of the colon.

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.5.1.4 DisableDeviceWithTimeOut

VARIANT_BOOL DisableDeviceWithTimeOut(LONG dwMachineNumber, LONG TimeOutSec)

To disable the machine for a period of time.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
TimeOutSec	LONG	[in]	Time period during which the machine is disabled

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.5.1.5 PowerOffDevice

VARIANT_BOOL PowerOffDevice(LONG dwMachineNumber)

To shut down the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.5.1.6 RestartDevice

VARIANT_BOOL RestartDevice(LONG dwMachineNumber)

To restart the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 5. 2 Online Registration Functions

5.5.2.1 StartEnroll

VARIANT_BOOL StartEnroll(LONG UserID, LONG FingerID)

To register a user. The machine will then enter the user registration state and waits for the user to scan fingerprints.

Parameters

Parameter description:

name	type	param direction	description of param
UserID	LONG	[in]	ID of the user to be registered
FingerID	LONG	[in]	Index of the fingerprint of the user, ranging from 0 to 9

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

After this function is invoked and the user scans fingerprints for three times, the machine may not respond when the user scans fingerprints again. In this case, invoke StartIdentify to enable the machine to enter the waiting state.

Note

Applicable to BW

5.5.2.2 StartEnrollEx

VARIANT_BOOL StartEnrollEx(BSTR UserID, LONG FingerID, LONG Flag)

To register a user. The machine will then enter the user registration state and waits for the user to scan fingerprints.

Parameters

Parameter description:

name	type	param direction	description of param
UserID	BSTR	[in]	ID of the user to be registered
FingerID	LONG	[in]	Index of the fingerprint of the user, ranging from 0 to 9
Flag	LONG	[in]	Flag that indicates whether the fingerprint template is valid or a

			duress fingerprint

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

1. After this function is invoked and the user scans fingerprints for three times, the machine may not respond when the user scans fingerprints again. In this case, invoke StartIdentify to enable the machine to enter the waiting state.

2. The Flag parameter indicates whether the fingerprint template is valid, or the fingerprint template is a duress fingerprint. The value 0 indicates that the fingerprint template is invalid, 1 indicates that the fingerprint template is valid, and 3 indicates that the fingerprint template is a duress fingerprint.

Note

Applicable to TFT

5.5.2.3 StartVerify

VARIANT_BOOL StartVerify(LONG UserID, LONG FingerID)

To start 1:1 comparison.

Parameters

Parameter description:

name	type	param direction	description of param
UserID	LONG	[in]	ID of the user to be verified
FingerID	LONG	[in]	Index of the fingerprint of the user, ranging from 0 to 9

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW and TFT devices

5.5.2.4 StartIdentify

VARIANT_BOOL StartIdentify()

To start 1:N comparison. Then the machine will enter the 1:N verification state.

Parameters

None

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW and TFT devices

5.5.2.5 CancelOperation

VARIANT_BOOL CancelOperation()

To cancel the current fingerprint registration state of the machine.

Parameters

None

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW and TFT devices

5. 5. 3 Card Operation Functions

5.5.3.1 WriteLCD

VARIANT_BOOL WriteLCD(LONG Row, LONG Col, BSTR Text)

To write data. Specifically, this function is used to write a character string to any row of any column on the machine.

Parameters

Parameter description:

name	type	param direction	description of param
Row	LONG	[in]	Start row
Col	LONG	[in]	Start column
Text	BSTR	[in]	Content to be written on the screen of the machine

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW

5.5.3.2 ClearLCD

VARIANT_BOOL ClearLCD()

To clear the screen of the machine. Specifically, this function is used to clear all information displayed on the LCD screen of the machine.

Parameters

None

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW

5.5.3.3 WriteCard

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

To write a specified user and the fingerprint template of the user to the MF card. After this function is invoked, the user needs to punch the MF card on the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	LONG	[in]	User ID

dwFingerIndex1	LONG	[in]	Fingerprint index, ranging from 0 to 3
TmpData1	BYTE*	[in]	Fingerprint template corresponding to the fingerprint index
dwFingerIndex2	LONG	[in]	Fingerprint index, ranging from 0 to 3
TmpData2	BYTE*	[in]	Fingerprint template corresponding to the fingerprint index
dwFingerIndex3	LONG	[in]	Fingerprint index, ranging from 0 to 3
TmpData3	BYTE*	[in]	Fingerprint template corresponding to the fingerprint index
dwFingerIndex4	LONG	[in]	Fingerprint index, ranging from 0 to 3
TmpData4	BYTE*	[in]	Fingerprint template corresponding to the fingerprint index

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

TmpData1 cannot be left blank

Note

Applicable to BW, TFT and IFACE devices

5.5.3.4 EmptyCard

VARIANT_BOOL EmptyCard(LONG dwMachineNumber)

To clear the MF card.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5. 5. 4 Others

5.5.4.1 GetLastError

GetLastError(LONG* dwErrorCode)

To query information about the last error.

Parameters

Parameter description:

name	type	param direction	description of param
dwErrorCode	LONG*	[out]	Error code

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The dwErrorCode parameter specifies the error code. The values are described as follows:

1. During connection, the following error codes may be returned:
0 Connected successfully
-1 Failed to invoke the interface
-2 Failed to initialize
-3 Failed to initialize parameters
-5 Data mode read error
-6 Wrong password
-7 Reply error
-8 Receive timeout
-307 Connection timeout
In invoking other interfaces, the following error codes may be returned:
-201 Device is busy
-199 New Mode
-103 device send back error of face version error
-102 face template version error, like 8.0 face template send to 7.0 device
-101 malloc memory failed
-100 Not supported or the data does not exist
-10 The length of transmitted data is incorrect
-5 Data already exists
-4 Insufficient space
-3 Wrong size
-2 File read/write error
-1 The SDK is not initialized and needs to be reconnected
0 Data not found or duplicate data
1 Correct operation
4 Parameter error
101 Buffer allocation error
102 repeat invoking

2. Underlying error codes:
-12001 Socket creation timeout (connection timeout)
-12002 Insufficient memory
-12003 Wrong Socket version
-12004 Not TCP protocol

-12005 Waiting timeout
-12006 Data transmission timeout
-12007 Data reading timeout
-12008 Failed to read Socket
-13009 Waiting event error
-13010 Exceeded retry attempts
-13011 Wrong reply ID
-13012 Checksum error
-13013 Waiting event timeout
-13014 DIRTY_DATA
-13015 Buffer size too small
-13016 Wrong data length
-13017 Invalid data read1
-13018 Invalid data read2
-13019 Invalid data read3
-13020 Data loss
-13021 Memory initialization error
-15001 Invoking return value of status key issued by SetShortkey interface repeatedly
-15002 Invoking return value of description issued by SetShortkey interface repeatedly
-15003 The two level menu is not opened in the device, and the data need not be issued

3. getdevicedata and setdevicedata invocation error codes
-15100 Error occurs in obtaining table structure
-15101 The condition field does not exist in the table structure
-15102 Inconsistency in the total number of fields
-15103 Inconsistency in sorting fields
-15104 Memory allocation error
-15105 Data parsing error
-15106 Data overflow as the transmitted data exceeds 4M
-15108 Invalid options
-15113 Data parsing error: table ID not found
-15114 A data exception is returned as the number of fields is smaller than or equal to 0
-15115 A data exception is returned as the total number of table fields is inconsistent with the total number of fields of the data

4. Firmware error codes:
2000 Return OK to execute
-2001 Return Fail to execute command
-2002 Return Data
-2003 Regstered event occorred
-2004 Return REPEAT Command
-2005 Return UNAUTH Command
0xffff Return Unknown Command
-4999 Device parameter read error
-4998 Device parameter write error
-4997 The length of the data sent by the software to the device is incorrect

-4996 A parameter error exists in the data sent by the software to the device
-4995 Failed to add data to the database
-4994 Failed to update the database
-4993 Failed to read data from the database
-4992 Failed to delete data in the database
-4991 Data not found in the database
-4990 The data amount in the database reaches the limit
-4989 Failed to allocate memory to a session
-4988 Insufficient space in the memory allocated to a session
-4987 The memory allocated to a session overflows
-4986 File does not exist
-4985 File read failure
-4984 File write failure
-4983 Failed to calculate the hash value
-4982 Failed to allocate memory

Note

This interface is applicable to the new architecture firmware.

5.5.4.2 GetHIDEventCardNumAsStr

VARIANT_BOOL GetHIDEventCardNumAsStr(BSTR* strHIDEventCardNum)

To query the number of the card that is punched most recently.

Parameters

Parameter description:

name	type	param direction	description of param
strHIDEventCardNum	BSTR*	[out]	Number of the punched card

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.5.4.3 CaptureImage

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

To capture an image of the finger of which the fingerprint is being scanned.

Parameters

Parameter description:

name	type	param direction	description of param
FullImage	BOOL	[in]	Whether to capture an image of the entire finger. The value True indicates capturing an image of the entire finger and False indicates capturing an image of only the fingerprint.
Width	LONG*	[in]	Width of the captured image
Height	LONG*	[in]	Height of the captured image
Image	BYTE*	[in]	Fingerprint image in binary format
ImageFile	BSTR	[in]	Name of the file of captured image to be saved (including the path)

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.5.4.4 UpdateFirmware

VARIANT_BOOL UpdateFirmware(BSTR FirmwareFile)

To upgrade the firmware. To use this function, obtain the firmware from the technical support engineers of our company beforehand.

Parameters

Parameter description:

name	type	param direction	description of param
FirmwareFile	BSTR	[in]	Name of the firmware file to be upgraded (including the path)

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.5.4.5 BeginBatchUpdate

VARIANT_BOOL BeginBatchUpdate(LONG dwMachineNumber, LONG UpdateFlagl)

To get ready for uploading data in batches. For example, if you invoke this function before uploading data such as fingerprint templates or user information, the SDK will temporarily store the data in the buffer. Then you can invoke BatchUpdate to upload the data to the machine.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

UpdateFlag1	LONG	[in]	Whether to overwrite the original fingerprint template

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The UpdateFlag1 parameter specifies whether to overwrite the original fingerprint template if a fingerprint with the same index already exists when you upload a fingerprint template. The value 1 indicates overwriting the original fingerprint template and 0 indicates not overwriting the original fingerprint template.

Note

Applicable to BW, TFT and IFACE devices

5.5.4.6 BatchUpdate

VARIANT_BOOL BatchUpdate(LONG dwMachineNumber)

To upload data in batches. Generally you are advised to invoke BeginBatchUpdate to upload the data to the buffer before invoking this function.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value

True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.5.4.7 CancelBatchUpdate

VARIANT_BOOL CancelBatchUpdate(LONG dwMachineNumber)

To cancel bulk data uploading. You can invoke this function after invoking BeginBatchUpdate but before invoking BatchUpdate. This function aims to release the buffer allocated for bulk data uploading.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.5.4.8 PlayVoice

VARIANT_BOOL PlayVoice(LONG Position, LONG Length)

To play announcements with specified consecutive indexes. The indexes depend on the machine. You can view the indexes, which range from 0 to 11, in voice testing on the machine.

Parameters

Parameter description:

name	type	param direction	description of param
Position	LONG	[in]	Index of the start announcement
Length	LONG	[in]	Index of the end announcement

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.5.4.9 PlayVoiceByIndex

VARIANT_BOOL PlayVoiceByIndex(LONG Index)

To play an announcement with the specified index. The index depends on the machine. You can view the index, which ranges from 0 to 11, in voice testing on the machine.

Parameters

Parameter description:

name	type	param direction	description of param
Index	LONG	[in]	Index of the announcement to be played

Returns

Value description:

name	type	description of value

True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW, TFT and IFACE devices

5.5.4.10 ReadAttRule

VARIANT_BOOL ReadAttRule(LONG dwMachineNumber)

To read the attendance rule of the machine. This function supports OP1000.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW devices

5.5.4.11 SaveTheDataToFile

VARIANT_BOOL SaveTheDataToFile(LONG dwMachineNumber, BSTR TheFilePath, LONG FileFlag)

To save the data in the buffer as a file. This function supports OP1000.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
TheFilePath	BSTR	[in]	Path for saving the file
FileFlag	LONG	[in]	File type flag

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The FileFlag parameter specifies the file type. The values are described as follows:
1 Attendance records
2 User
3 Attendance rule
4 Department list
5 Shift

Note

Applicable to BW devices

5.5.4.12 ReadTurnInfo

VARIANT_BOOL ReadTurnInfo(LONG dwMachineNumber)

To read the shift information on the machine. This function supports OP1000.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Note

Applicable to BW devices

5.5.4.13 SSR_OutPutHTMLRep

VARIANT_BOOL SSR_OutPutHTMLRep(LONG dwMachineNumber, BSTR dwEnrollNumber, BSTR AttFile, BSTR UserFile, BSTR DeptFile, BSTR TimeClassFile, BSTR AttruleFile, LONG BYear, LONG BMonth, LONG BDay, LONG BHour, LONG BMinute, LONG BSecond, LONG EYear, LONG EMonth, LONG EDay, LONG EHour, LONG EMinute, LONG ESecond, BSTR TempPath, BSTR OutFileName, LONG HTMLFlag, LONG resv1, BSTR resv2)

To generate the attendance report of a user within the specified time period in HTML format. This function supports OP1000.

Parameters

Parameter description:

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine ID
dwEnrollNumber	BSTR	[in]	User ID
AttFile	BSTR	[in]	Name of the attendance record file
UserFile	BSTR	[in]	Name of the user information file
DeptFile	BSTR	[in]	Name of the department information file
TimeClassFile	BSTR	[in]	Name of the shift information file
AttruleFile	BSTR	[in]	Name of the attendance rule file
BYear	LONG	[in]	Start time of the time period

BMonth	LONG	[in]	End time of the time period
BDay	LONG	[in]	Start time of the time period
BHour	LONG	[in]	Start time of the time period
BMinute	LONG	[in]	Start time of the time period
BSecond	LONG	[in]	Start time of the time period
EYear	LONG	[in]	End time of the time period
EMonth	LONG	[in]	End time of the time period
EDay	LONG	[in]	End time of the time period
EHour	LONG	[in]	End time of the time period
EMinute	LONG	[in]	End time of the time period
ESecond	LONG	[in]	End time of the time period
TempPath	BSTR	[in]	Path where other exception files are saved
OutFileName	BSTR	[in]	Attendance report file name (including the path)
HTMLFlag	LONG	[in]	HTML report format
resv1	LONG	[in]	Reserved
resv2	BSTR	[in]	Flow report name

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

The HTMLFlag parameter specifies the HTML report type. The values are described as follows:

1 Flow report
2 Exception report
3 Statistics report

Note

Applicable to BW devices

5.5.4.14 SendFileByType

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine equipment number
FileName	BSTR	[in]	File full path name
iType	LONG	[in]	File type

5.5.4.15 SetCommProType

VARIANT_BOOL SetCommProType(LONG, proType)

Set the priority to use PULL or Standlone SDK to connect the device.

Parameters

Parameter description:

name	type	param direction	description of param
proType	LONG	[in]	Priority protocol type

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

proType=1 Priority to use the Standlone SDK protocol proType=2 Priority to use the PULL SDK protocol Default to use Standlone SDK protocol to connect devices. If the customer does not develop PULL devices, this interface is not recommended by the developer.

Note

No dependency with device.

5.5.4.16 SetCommProType

VARIANT_BOOL SetCommProType(LONG proType)

Set the priority to use PULL or Standlone SDK to connect the device.

Parameters

Parameter description:

name	type	param direction	description of param
proType	LONG	[in]	Priority protocol type

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

proType=1 Priority to use the Standlone SDK protocol proType=2 Priority to use the PULL SDK protocol Default to use Standlone SDK protocol to connect devices. If the customer does not develop PULL devices, this interface is not recommended by the developer.

Note

No dependency with device.

5.5.4.17 GetConnectStatus

VARIANT_BOOL GetConnectStatus(LONG* dwErrorCode)

Get the error code return from the function BatchUpdate()

Parameters

Parameter description:

name	type	param direction	description of param
dwErrorCode	LONG*	[out]	Error code

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

When (dwErrorCode >= -11004 && dwErrorCode <= -10004), the software will disconnect and re-connect the device.

Note

Only used to get the error code return from the function BatchUpdate(), without the limitation of devices.

5.5.4.18 SetDeviceTableData

VARIANT_BOOL SendFileByType(LONG dwMachineNumber, BSTR TableName, BSTR Datas, BSTR Options, LONG* Count)

This function applies only to the time and attendance applications that comply with the PULL protocol in the new firmware. Used to set and insert data (such as time segments, user information, and leaves settings) into a device. The data can be one or more records. If the primary key of an inserted record already exists in the device, the original record is overwritten.

Parameters

name	type	param direction	description of param
dwMachineNumber	LONG	[in]	Machine equipment number
TableName	BSTR	[in]	Data table name
Datas	BSTR	[in]	Data record representation

Options	BSTR	[in]	Default empty, use for expansion
Count	LONG*	[out]	The number of records which were issued successfully

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

Indication of data records. Data is in the text format. Multiple records are separated by \r\n. Various pairs of "field=value" are separated by \t.

Note

5.5.4.19 SearchDevice

VARIANT_BOOL SearchDevice(BSTR commType, BSTR address, BSTR* DevBuffer, LONG DevBufferSize)

Use web search devices.

Parameters

Parameter description:

name	type	param direction	description of param
commType	BSTR	[in]	Communication type
address	BSTR	[in]	Communication data
DevBuffer	BSTR*	[out]	Device list
DevBufferSize	LONG	[in]	Size of device list

Returns

Value description:

name	type	description of value
True	BOOL	Function execution success
False	BOOL	Function execution failure

See also

Attention

commType=UDP
address=255.255.255.255 The return data include the IP address, serial number, firmware version, etc.

Note

Applicable to some customized devices

6 FAQs

6.1 How to Download Attendance Records?

First, use ReadGeneralLogData to read all attendance records and write them into the memory. Then, use SSR_GetGeneralLogData repeatedly to obtain attendance records. When SSR_GetGeneralLogData returns False, it means that all attendance records are obtained. Then, you can write the obtained records into database or display them in other forms to finish downloading. You can follow the same steps to down operation records.

Note

BW device use GetGeneralLogData instead of SSR_GetGeneralLogData

6.2 How to Create a User Online?

First, use SSR_SetuserInfo to write user information (such as enrollment number, password, and name) into the device. Then, use SSR_SetUserTmpStr/SSR_SetUerTmp to set fingerprint templates for the user. This method improves enrollment efficiency and is suitable when user information has been collected and stored in media such as database.

To upload user information and corresponding fingerprint templates in batches, use BeginBatchUpdata and SetUserInfo together with BarchUpdata, EnableDevice, or RefreshData. For details, see demo program.

Note

BW device use SetuserInfo instead of SSR_SetuserInfo, SetUserTmpStr instead of SSR_SetUserTmpStr, SetUerTmp instead of SSR_SetUerTmp.

6.3 How to Import or Download Data from USB Disk?

Many existing offline products support data download from USB disks. Many customers are concerned about data formats of USB disks. As the downloaded data formats are complex, ZKSoftware provides a tool for importing data from USB disks to database. Database is open for customers to download data. In addition, ZKSoftware also provides examples on how to process data files (*.dat, etc.) collected from USB disks and how to write the data into specified data files. All structs adopt 1 byte alignment mode.

Data in USB disks include user information, fingerprint templates, face templates, attendance records, and short messages. Detailed data structures are used in demo program. They are described briefly below:

User data structure:

```
typedef struct _User_{
  U16 PIN; //Internal number of a user
  U8 Privilege;
  char Password[8];
```

```
    char Name[24];
    U8 Card[4];
    U8 Group;
    U16 TimeZones[4];
    char PIN2[24];      //User ID
   }
```

Data structure of 9.0 fingerprint template:

```
typedef struct _Template_{
U16 Size;               //fingerprint template length
U16 PIN;                      //internal user ID, which corresponds to PIN2 in user table
BYTE FingerID;      // fingerprint backup index
BYTE Valid;
BYTE Template[MAXTEMPLATESIZE]; //maximize template length
}       //MAXTEMPLATESIZE           602 Bytes
```

Data structure of template.fp10:

```
typedef struct _Template_{
        U16 Size;                               //entire structure data size
        U16 PIN;                                 //user ID
        BYTE FingerID;                           //fingerprint index
        BYTE Valid;                              //flag
        BYTE *Template; //template
}
```

Attendance record,

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

Exported as a text file:

attlog.dat format explanation:

segment:

BadgeNumber(employee number),

checktime,

DeviceID,

checktype(check status),

VerifyCode(verification ways:password or fingerprint or other)

Workcode

There is an Ascii code #9(Tab) between each segment. When development, move to the segment value you want to choose by "Tab".

If the device has the output data protection function, the serial number of the current device is displayed in first line and the hash value in the last line of the file to which attendance records are exported from USB disk.

Data structure of SMS

```
typedef struct _SMS_{
BYTE Tag;             //category
U16 ID;          //data content flag. 0 indicates that the record is invalid.
U16 ValidMinutes;  //valid minutes. 0 indicates that the record is permanently valid.
U16 Reserved;
U32 StartTime;       //start time
BYTE Content[MAX_SMS_CONTENT_SIZE+1];     //short message content
}       // MAX_SMS_CONTENT_SIZE    60 Bytes
```

Data structure between SMS and user pin//user->sms,udata.dat

```
typedef struct _UData_{
U16 PIN;              //0 indicates that the record is invalid
U16 SmsID;
}GCC_PACKED TUData, *PUData;    //4Bytes
```

Data structure of face template:

```
typedef struct _FaceTmp_{
        U16 Size;//face template size
        U16 PIN;//user ID
        BYTE FaceID;//Face ID
        BYTE Valid;//flag
        U16 Reserve;//reserve
        U32 ActiveTime;
        U32 VfCount;//Verify Count
        BYTE FaceTmp[FACE_TMP_MAXSIZE]
    }   //FACE_TMP_MAXSIZE=1024*2+512
```

Note

BW device use User data structure:

```
    typedef struct _User_{
        U16 PIN; //Internal number of a user
        U8 Privilege;
        char Password[5];
        char Name[8];
        U8 Card[5];        //ID No which used for store the relevant ID No
        U8 Group;          //the Group user belongs to
        U16 TimeZones;     //user can use time zone
        U32 PIN2;          //User ID
    }
```

6.4 How to Use Biokey to Write the Collected Fingerprint Templates Offline?

When a fingerprint is collected, Biokey usually obtains the fingerprint template during enrollment. For example, the currently enrolled fingerprint template can be obtained via OnEnroll. After obtaining the fingerprint template, Biokey converts it into an offline fingerprint template. Then, the template can be written into the device.

6.5 How to Obtain All Information of All Users?

Use ReadAllUserID to read IDs of all users and write them into memory. Then, use SSR_GetAllUserInfo repeatedly to obtain EnrollNumber of users, and use SSR_GetUserInfo to obtain user information. If necessary, you can also use SSR_GetUserTmpStr to obtain the fingerprint templates in string form.

Note

BW device use GetAllUserInfo instead of SSR_GetAllUserInfo, GetUserInfo instead of SSR_GetUserInfo, GetUserTmpStr instead of SSR_GetUserTmpStr.

6.6 How to Connect to the Device?

During connection, the device can be regarded as an independent PC. However, the IP address of the device must match the IP address to be connected. Some devices, for example F4, support serial port connection and network connection. Therefore, during different connections, you need to set the device differently, modify communication mode, and set the controller switch to TCP/IP or RS232/485. Otherwise, the connection may fail. Sometimes, if the device fails to be connected due to busy serial ports, you can restart the program. If the application keeps connecting to the device without being manually disconnected, you can use DisableDeviceWithTimeOut to set the automatic disconnection time of the device. If some connections are used to download or modify data via serial ports or network, you can use EnableDevice to keep the device working and restore the connections after communication finishes, so as to maintain data consistency and avoid unknown errors.

6.7 Password Is Invalid After SetUserInfo Is Used.

After SSR_SetUserInfo is called, Password may be set to null. If so, password verification will fail. To keep the password unchanged when writing user information, use SSR_GetUserInfo to obtain user password and transmit the password value to the Password parameter of SSR_SetUserInfo before using SSR_SetUserInfo.

Note

BW device use SetUserInfo instead of SSR_SetUserInfo, GetUserInfo instead of SSR_GetUserInfo.

6.8 How to Convert an Online Template into an Offline Template?

Use FPTempConvertNew to convert the collected templates into offline fingerprint templates. See related descriptions of Biokey SDK for how to obtain the templates collected by Biokey. FPTempConvertNew is used to convert binary fingerprint templates. Parameters temp1 and temp2 are binary parameters. You can also use FPTempConvertNewStr to convert Biokey fingerprint templates of string type into offline fingerprint templates.

6.9 Demo Program Fails to Connect to the Device.

After the attendance management program is installed, users can connect to the device by using the attendance management program, but cannot connect to the device by using demo program. The reason is that DLL is copied to the directory of the attendance management program but registered in the installation directory during program installation. Generally, SDK loads controls from the system directory. Therefore, if the SDK version in the system directory is different from that in the attendance software directory, conflicts occur. (DLL function addresses of different versions are different, but OCX functions are the same in programming. Therefore, the problem is found only during runtime.)

Caution: The common procedure for registering the SDK in the system is as follows:

1. If an SDK has been already registered in the system, run **regsvr32 /u zkemkeeper.dll** to unregister the original SDK.

2. Copy all DLLs to the system directory, for example, win2000 is located in winnt\system32.

3. Run regsvr32 "registration path\zkemkeeper.dll" to register the SDK.

4. Correctly load controls in development environment (learn the usage of development tool by yourself. Relevant details are omitted here).

5. Try to use the SDK of the same version in development or running environment.

6.10 Offline Fingerprint Device Keeps Working After Being Connected.

After connecting the SDK to the offline fingerprint device, use EnableDevice to keep the offline fingerprint device working (see EnableDevice), so as to maintain data consistency and avoid unknown errors. After the offline fingerprint device is working, the keypad and fingerprint sensor stop working. After communication finishes, disconnect the SDK from the device or use EnableDevice again to restore the offline fingerprint device to normal state.

DisableDeviceWithTimeOut is recommended.

6.11 Illegal Characters Are Displayed or Screen Display Is Abnormal After Non-English Names or Short Messages Are Uploaded to the Device.

First, check whether the device supports the specified language. For example, if the current language of the device is English, but an Arabic name is uploaded to the device, the name cannot be displayed normally. If the device already supports the language, but the name still cannot be displayed, use related functions to convert the user name into UTF-8 format (for example, use AnsiToUTF8() in Dephi), and then use SSR_SetUserInfo to upload the user name.

Note

BW device use SetUserInfo instead of SSR_SetUserInfo.

6.12 Card Management Problems

How to register a card in the device? How to obtain the user card?

The SDK has the cardnumber parameter. If this parameter is invisible in development environment, use GetStrCardNumber and SetStrCardNumber instead.

For a user enrolled in the device, the card number is a kind of user information. When SSR_SetUserInfo is used to set user information, cardnumber is automatically used as the card number and set for the user.

The procedure for registering a card is as follows:

Set cardnumber -> Upload user information

The procedure for obtaining the card number of a user is as follows:

Obtain information of the specified user -> Obtain cardnumber

Note: The card number is internally defined as four unsigned bytes of long type. If VB does not support four unsigned bytes, verification can be started after the last three bytes of the card number are input (if the last three bytes are different from each other).

Note

BW device use SetUserInfo instead of SSR_SetUserInfo.

6.13 Firewall or Router Traversal

In most cases, the device to be connected needs to traverse firewalls or routers, and UDP socket and port 4370 are used for SDK communication. Therefore, UDP and port 4370 must be enabled on firewalls or routers. If the device traverses gateways via port mapping, the device can be accessed via port number and IP address of routers. Generally, if UDP and port 4370 are enabled and can be pinged, the device can be connected. Certainly, in the case of data download, network connection must be considered. In addition, some devices that support SOAP ports can be accessed via embedded Web Server and SOAP.

Caution: The zem100 series products can traverse internet via port mapping. For zem200 products, as the devices run on Linux, they can be accessed after the gateway is configured if the local network environment supports gateway communication. Certainly, there are still some other methods for accessing the device, for example, VPN and IP mapping. The connection scheme should be selected according to specific network environments.

6.14 Difference between ZKFinger10.0 and ZKFinger9.0 and Comparison between Templates

Algorithm performance: Compared with ZKFinger9.0, ZKFinger10.0 achieves better false acceptance rate (FAR), false rejection rate (FRR), and enrollment rejection rate (ERR), better image processing effect of low-quality fingerprints (for example, fingerprints are too dry or too wet, or users have worn or injury), and 10 times faster comparison.

Template size: The size of a ZKFinger10.0 fingerprint template is about 1.6 KB. The size of a ZKFinger9.0 fingerprint template is about 512B. When ZKFinger10.0 is used, a Mifare card with at least 2 KB capacity should be used for data storage.

Template compatibility: The ZKFinger10.0 and ZKFinger9.0 fingerprint templates are incompatible with each other. If a user who have already registered ZKFinger9.0 fingerprint templates wants to use ZKFinger10.0, the user has to register fingerprint templates again, and vice versa.

6.15 Uploading a Large Capacity of Fingerprints

Large capacity usually means over 1500 fingerprints. Some devices can hold 8000 fingerprints or more. Fingerprint templates must be uploaded in batches. In this mode, the upload is much faster. For how to upload fingerprint templates in batches, see descriptions of batch process function.

6.16 Differences between High-speed Upload and Ordinary Upload

In an ordinary upload, each time upload functions (such as SSR_SetUserinfo and SSR_SetUserTmpStr) are used, the SDK communicates with the device and uploads related data to the device.

In a high-speed upload, BeginBatchUpdata is used to create a temporary buffer to store the data to be uploaded in subsequent operations. Then, BatchUpdata can be used to upload all the data in the buffer to the device at a time. This mode greatly reduces communications between the SDK and the device, and raises the speed of large-capacity upload in particular.

Note

BW device use SetUserinfo instead of SSR_SetUserinfo, SetUserTmpStr instead of SSR_SetUserTmpStr.

6.17 How to Determine Whether the Device Uses ZKFinger10.0 or ZKFinger9.0?

Use the following function:

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

The Option parameter constantly "~ZKFPVersion". If the returned Value is 10, the device uses ZKFinger10.0. If the returned Value is 9 or null (null is returned as old TFT devices do not have this value), the device uses ZKFinger9.0.

For example:

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

If verSionFp='10', the device uses ZKFinger10.0

If verSionFp='9' or verSionFp='', the device uses ZKFinger9.0

6.18 How to Upload, Download, and Delete ZKFinger10.0 Templates?

ZKFinger10.0 provides faster comparison, but the template size and storage mode are different from those of older versions:

When ZKFinger10.0 is used, the size of a fingerprint template is about 1.6 KB. When an older version is used, the size of a fingerprint template is smaller than 608B.

You can use follow function to upload,download and delete ZKFinger10.0 Templates:

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)

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)

The above four functions can upload and download both ordinary fingerprint templates (Flag=1) and threatened fingerprint templates (Flag=3). Additionally, they can be used for both ZKFinger10.0 templates and ZKFinger9.0 templates.

To delete ZKFinger10.0 templates, you can use the following functions (these two functions are used on TFT devices to delete fingerprint templates):

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)

Note:

Ver6.60 is the internal version of device firmware. You can obtain it by using GetFirmwareVersion after SDK connects to the device (or by using the attendance software). Caution: The internal version obtained is different from the firmware version that you view on the device.

6.19 How to Upload, Download, and Delete ZKFinger9.0 Templates?

The functions used to upload and download ordinary ZKFinger9.0 templates (that is, SSR_GetUserTmp, SSR_GetUserTmpStr, SSR_SetUserTmp, and SSR_SetUserTmpStr) in earlier SDK versions are continuously used in the present SDK version.

New functions (that is, SetUserTmpEx, GetUserTmpEx, SetUserTmpExStr, and GetUserTmpExStr) are also added in the present SDK version. These functions can upload and download both ordinary fingerprint templates (Flag=1) and threatened fingerprint templates (Flag=3). But they are used only on the TFT devices with firmware Ver6.60 or later. Additionally, they can be used for both ZKFinger10.0 templates and ZKFinger9.0 templates.

To delete ZKFinger9.0 templates, you can use SSR_DelUserTmp or SSR_DelUserTmpExt.

6.20 How to Download a Face Template?

1. The transmission mode of face templates is the same as that of ZKFinger10.0 fingerprint templates.

2. One user has about 15 face templates in different angles. Each template consists of 2576 bytes. The third and fourth bytes of each template indicate the ID, corresponding to the first and second bytes of user structure. For devices that support face identification, the last 24 bytes of user structure indicate the user ID. Therefore, the total size of face templates of each user is about 37 KB. You are not recommended to upload or download data via serial ports. When the value of dwFaceIndex is 50, all face templates of a user are uploaded or downloaded.

3. To upload and download ZKFace templates, use the following functions:

SetUserFace(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG dwFaceIndex, BYTE* TmpData, LONG TmpLength, VARIANT_BOOL* pVal);

GetUserFace(LONG dwMachineNumber, BSTR dwEnrollNumber, LONG dwFaceIndex, BYTE* TmpData, LONG * TmpLength, VARIANT_BOOL* pVal);

DelUserFace (LONG dwMachineNumber, BSTR dwEnrollNumber, LONG dwFaceIndex, VARIANT_BOOL* pVal);

For details, see description of the functions.