ZKSoftware SOAP SDK Development Handbook

Version: 1.0

Date: February, 2008

.

If there is some change of information in this handbook, ZKSoftware won't inform specially.

· ZKSoftware Inc. is our company logo. Biokey is core technology logo of ZKSoftware. Both the two logos have been registered in China and America.

· Other trademark and product name mentioned in this handbook are produced by other companies. ZKSoftware Inc. has no property of them.

· Please solve use problem (instead of development kit problem) during development by yourself.

· ZKSoftware has no responsibility for data loss caused by user or program.

CONTENTS

1. SOAP SDK description

SOAP (Simple Object Access Protocol)is a protocol to exchange information in separated or distributed environment. It is based on XML. SOAP defines a system to transmit commands and parameters between HTTP client terminal and server. Except for HTTP, SOAP has nothing to do with operation system, program language or object model used on server or client terminal. It is simple. Client terminal sends a request to server and calls corresponding object. Then server returns the result. These information is in SML format, and be enveloped into information in accordance with HTTP protocol. SOAP works through internet underwork which has existed before. It is in accordance with any router, fire wall or agent server.

SOAP includes four parts:

1. SOAP envelop: it defines a frame (what is the information content, who sends it, who should receive and deal with it and how to deal with it.)
2. SOAP encoding rules: it is used to express data type example needed by application.
3. SOAP RPC representation: protocol to call and reply remotely.
4. SOAP binding: use bottom protocol to exchange information.

Although the four parts are parts of SOAP, they are overlapped and independent. Especially, envelop and encoding rules are defined in different XML name space, which makes definition simpler.

SOAP has two main design aim: simple and expandable, which mean some characters in traditional information system or distributed object system are not a part of SOAP specification, such as distributed garbage collection, boxcarring or batching of messages, objects-by-reference(which requires distributed garbage collection),Activation(which requires objects-by-reference)and so on.

SOAP, as a common protocol to transmit XML information in WEB service stack, plays a very important part. In internet, the system is incompact, coupling, cross platform. It has nothing to do with language and special interface and needs to provide reliable access to Web application. "software-software dialogue" is used to call each other, and collision among software application, website and various devices has been broken, achieving “seamless integration on WEB”.

ZKSoftware SOAP SDK is a tool to communicate data with offline fingerprint sensor through XML protocol. It can manage user information and fingerprint on offline fingerprint sensor conveniently. It can also download card record and set functions for fingerprint sensor:

1. Download card record on fingerprint sensor.
2. Download and upload user information and fingerprint information.
3. Download and set system information.

2. SOAP SDK installation

how to install

1．Download SOAP SDK3.0 from Microsoft website.

2． Installation file (soap sdk.exe.) Install SOAP development kit to the computer.

3. Communication process description

There are three steps to create SOAP client terminal application:

1- Specify and connect Web server.

2- Prepare and send message.

3- Read message returned from server.

[Example]

Give command to server to restart device. XML format is as the following:

[XML protocol]

Request Xml:

```
        <Restart>
                <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
       </Restart>
```

Response Xml:

```
   <RestartResponse>
          <Row>
             <Result>Succeed! </Result>
       </Row>
   </RestartResponse>
```

VC++source code:

```
//Use http protocol to define connection with server pointer.
ISoapConnectorPtr Connector=NULL;
Connector.CreateInstance(__uuidof(HttpConnector30));
//Specify device URL as parameter IP.
Connector->Property["EndPointURL"] = IP;
// Connect device.
Connector->Connect();
//Be about to send SOAP request.
Connector->BeginMessage();
//Create SOAP message to be sent to server.
ISoapSerializerPtr Serializer=NULL;
Serializer.CreateInstance(__uuidof(SoapSerializer30));
//Send request through serializer with MS SOAP Toolkit embedded.
Serializer->Init(_variant_t((IUnknown*)Connector->InputStream));
//Start to deal with SOAP message.
Serializer->StartEnvelope("","","");
//Start body.
Serializer->StartBody("");
//Deal with the subelement of the first layer. The first parameter ---element name. The command
parameter to get user information service--- GetUserInfo,the second parameter---URL.
Serializer->StartElement("GetUserInfo","http://www.zksoftware/Service/message/","","");
//Deal with the subelement of the second layer. the first parameter---element name, port
parameter---ArgComKey,the second parameter---URL.
```

```
Serializer->StartElement("ArgComKey","http://www.zksoftware/Service/message/","","");
//Write element value.
Serializer->WriteString(Com);
//End the subelement of the second layer---ArgComKey.
Serializer->EndElement();
//Deal with the subelement of the second layer. the first parameter ---element name,
parameter---Arg,the second parameter---URL.
Serializer->StartElement("Arg","","","");
//Deal with the third subelement. the first parameter---element name, user number parameter
---PIN.
Serializer->StartElement("PIN","","","");
//Write element value.
Serializer->WriteString(“All”);
//End subelement PIN of the third layer.
Serializer->EndElement();
//Deal with the third subelement. the first parameter---element name, user fingerprint number
parameter---FingerID.
Serializer->StartElement("FingerID","","","");
//Write element value.
Serializer->WriteString("All");
//End subelement of the third layer. (PIN)
Serializer->EndElement();
//End the subelement of the second layer (Arg).
Serializer->EndElement();
//End the first layer’s subelement (GetUserInfo).
Serializer->EndElement();
//End body.
Serializer->EndBody();
//End SOAP message.
Serializer->EndEnvelope();
//Send message to server.
Connector->EndMessage();
//Receive data if the returned value is not blank.
if(Connector->OutputStream!=NULL)
{
//Create SOAPReader object.
ISoapReaderPtr Reader=NULL;
Reader.CreateInstance(__uuidof(SoapReader30));
// Connect outputstream.
Reader->Load(_variant_t((IUnknown*)Connector->OutputStream), "");
//Use text property to get the element property value.
Reader->RPCResult->text,
}
```

4. operation description

4.1 data management

[common parameter]

ComKey: communication password (connection password of corresponding device. Only the local communication password is in accordance with the connection password of device, can device data be sent to the local and be received correctly by the local.)

PIN: user number(enrollment number)

4.1.1 user information

4.1.1.1 obtain user information

[function]

Obtain user information through user enrollment number.

[XML protocol]

Request Xml:

```
      <GetUserInfo>
                  <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
                  <Arg>
                      <PIN xsi:type="xsd:integer”>Job Number</PIN>
                </Arg>
     </GetUserInfo>
```

Response Xml:

```
   <GetUserInfoResponse>
            <Row>
                <PIN>XXXXX</PIN>
                <Name>XXXX</Name>
                <Password>XXX</Password>
                < Group>X</ Group>
                < Privilege>X</ Privilege>
                <Card>XXXX </Card>
                <PIN2>XXXX </PIN2>
                <TZ1>XXX </TZ1>
                <TZ2>XXX </TZ2>
                <TZ3>XXX </TZ3>
         </Row>
   </GetUserInfoResponse>
```

[parameter]

ComKey: communication password

PIN: user number(enrollment number).

[return value]

Return user information, or return blank.

PIN:5 bits code is equal to PIN2 code. 9 bits code is internal code.

Name: user name.

Password: user password.

Group:group

Privilege:user privilege,0,common user 1,administrator.

Card:card number

PIN2: user number(enrollment number).

TZ1:time zone 1

TZ2:time zone 2

TZ3:time zone 3

4.1.1.2 write user information

[function]

Comparatively, enroll a user, and write user's information.

[XML protocol]

Request Xml:

```
<SetUserInfo>
        <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
        <Arg>
            <PIN>XXXXX</PIN>
            <Name>XXXX</Name>
            <Password>XXX</Password>
            < Group>X</ Group>
            < Privilege>X</ Privilege>
            <Card>XXXX </Card>
            <PIN2>XXXX </PIN2>
            <TZ1>XXX </TZ1>
            <TZ2>XXX </TZ2>
            <TZ3>XXX </TZ3>
        </Arg>
</SetUserInfo>
```

Response Xml:

```
<SetUserInfoResponse>
        <Row>
            <Result>Succeed! </Result>
        </Row>
</SetUserInfoResponse>
```

[parameter]

ComKey: communication password

PIN:5 bits code is equal to PIN2 code. 9 bits code is internal code.

Name: user name
Password: user password
Group:group
Privilege:user privilege,0,common user 1,administrator
Card:card number
PIN2: user number(enrollment number)
TZ1:time zone 1
TZ2:time zone 2
TZ3:time zone 3

[return value]

Return True if it is successful, or return False.
Comparatively, enroll user, but hasn’t write fingerprint template yet. SetUsertTemplate function can be used to upload fingerprint template data.

4.1.1.3 delete some user information

[function]

Delete some user.

[XML protocol]

Request Xml:

```
        <DeleteUser>
                <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
                <Arg>
                 <PIN xsi:type="xsd:integer”>Job Number</PIN>
            </Arg>
       </DeleteUser>
```

Response Xml:

```
   <DeleteUserResponse>
          <Row>
             <Result>Succeed! </Result>
        </Row>
   </DeleteUserResponse>
```

[parameter]

ComKey: communication password
PIN: user number(enrollment number)

[return value]

Return True if it is successful,or return False..

4.1.1.4 obtain all user information

[function]

Obtain all user information.

[XML protocol]

Request Xml:

```
            <GetAllUserInfo>
                        <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
          </GetAllUserInfo>
```

Response Xml:

```
    <GetAllUserInfoResponse>
            <Row>
               <PIN>XXXXX</PIN>
               <Name>XXXX</Name>
               <Password>XXX</Password>
               < Group>X</ Group>
               < Privilege>X</ Privilege>
               <Card>XXXX </Card>
               <PIN2>XXXX </PIN2>
               <TZ1>XXX </TZ1>
               <TZ2>XXX </TZ2>
               <TZ3>XXX </TZ3>
         </Row>
    </GetAllUserInfoResponse>
```

[parameter]

ComKey: communication password

[return value]

Return user information, or return blank.

PIN:5 bits code is equal to PIN2 code. 9 bits code is internal code.

Name: user name

Password: user password

Group:group

Privilege:user privilege,0,common user 1,administrator

Card:card number

PIN2: user number(enrollment number)

TZ1:time zone 1

TZ2:time zone 2

TZ3:time zone 3

4.1.1.5 clear user password

[function]

clear user password.

[XML protocol]

Request Xml:

```
            <ClearUserPassword>
                        <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
                        <Arg>
                        <PIN xsi:type="xsd:integer”>Number</PIN>
                        </Arg>
```

```
       </ClearUserPassword>
```

Response Xml:

```
    <ClearUserPasswordResponse>
            <Row>
               <Result>Succeed! </Result>
         </Row>
    </ClearUserPassword>
```

[parameter]

ComKey: communication password

PIN:user number(enrollment number).

[return value]

Return True if it is successful,or return False..

clear all user information

[function]

clear all user information.

[XML protocol]

Request Xml:

```
     <ClearData>
               <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
               <Arg>
               <Value xsi:type="xsd:integer”>3</Value>
               </Arg>
    </ClearData>
```

Response Xml:

```
    <ClearDataResponse>
            <Row>
               <Result>Succeed! </Result>
         </Row>
    </ClearData>
```

[parameter]

ComKey: communication password

Value:operation value

[return value]

Return True if it is successful,or return False.

obtain user state

[function]

Obtain user state information.

[XML protocol]

Request Xml:

```
     <AttState>
               <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
```

```
                    <Arg>
                    <Action>“Get”</Action>
                    </Arg>
         </AttState>
```

Response Xml:

```
   <AttStateResponse>
           <Row>
              <Uid>XXXXX</Uid>
              <State>XXXX</State>
           </Row>
   </AttStateResponse>
```

[parameter]

ComKey: communication password

[return value]

Return user information, or return blank.

Uid:5 bits code is equal to PIN2 code. 9 bits code is internal code.

State: user state

set user state

[function]

Set user state information.

[XML protocol]

Request Xml:

```
         <AttState>
                    <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
                    <Arg>
                    <Action >“Set”</Action>
                    <Uid> XXXXX </Uid>
                    <State> XXXXX </State>
                    </Arg>
         </AttState>
```

Response Xml:

```
   <AttStateResponse>
           <Row>
                <Result>Succeed! </Result>
           </Row>
   </AttStateResponse>
```

[parameter]

ComKey: communication password

Uid　user ID number

State　user state

[return value]

Return True if it is successful, or return False.

delete user state

[function]

Delete user state information.

[XML protocol]

Request Xml:

```
<AttState>
        <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
        <Arg>
        <Action >“Del”</Action>
        <Uid> XXXXX </Uid>
        </Arg>
</AttState>
```

Response Xml:

```
<AttStateResponse>
    <Row>
        <Result>Succeed! </Result>
    </Row>
</AttStateResponse>
```

[parameter]

ComKey: communication password

Uid user ID number

[return value]

Return True if it is successful, or return False.

4.1.2 fingerprint management

4.1.2.1 obtain user fingerprint template information

[function]

Obtain user fingerprint template information in string.

[XML protocol]

Request Xml:

```
<GetUserTemplate>
        <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
        <Arg>
            <PIN xsi:type="xsd:integer”>Job Number</PIN>
            <FingerID xsi:type="xsd:integer”>Finger Number</FingerID>
      </Arg>
</GetUserTemplate>
```

Response Xml:

```
<GetUserTemplateResponse>
```

```
                <Row>
                    <PIN>XXXXX</PIN>
                    <FingerID>XX</FingerID>
                    <Size>XXX</Size>
                    <Valid>X</Valid>
                    <Template>XXXXXXXXXXXXXXXX….</Template>
            </Row>
        </GetUserTemplateResponse>
```

[parameter]

ComKey: communication password

PIN: user number(enrollment number)

FingerID: user fingerprint number

[return value]

Return specified user fingerprint information

Pin: user enrollment number

Finger ID:fingerprint enrollment number

Size:fingerprint size

Template:fingerprint information

Valid:validity

4.1.2.2 write user fingerprint template information

[function]

Write user fingerprint template in string, namely, upload No.dwFingerIndex fingerprint template TmpData of dwPin corresponding user to the connected device.

[XML protocol]

Request Xml:

```
          <SetUserTemplate>
                      <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
                      <Arg>
                          <PIN>XXXXX</PIN>
                          <FingerID>XX</FingerID>
                          <Size>XXX</Size>
                          <Valid>X</Valid>
                          <Template>XXXXXXXXXXXXXXXX….</Template>
                    </Arg>
          </SetUserTemplate>
```

Response Xml:

```
        <SetUserTemplateResponse>
                <Row>
                    <Result>Succeed! </Result>
                </Row>
        </SetUserTemplateResponse>
```

[parameter]

ComKey: communication password
PIN: user number(enrollment number)
Finger ID:fingerprint enrollment number
Size:fingerprint size
Template:fingerprint information
Valid:validity

[return value]

Return True if it is successful, or return False..

4.1.2.3 delete user fingerprint template information

[function]

Delete user fingerprint template information.

[XML protocol]

Request Xml:

```
        <DeleteTemplate>
                 <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
                 <Arg>
                  <PIN xsi:type="xsd:integer”>Job Number</PIN>
            </Arg>
       </DeleteTemplate>
```

Response Xml:

```
   <DeleteTemplateResponse>
          <Row>
             <Result>Succeed! </Result>
         </Row>
   </DeleteTemplateResponse>
```

[parameter]

ComKey: communication password
PIN: user number(enrollment number).

[return value]

Return True if it is successful, or return False..

4.1.2.4 clear all user fingerprint template information

[function]

clear all user fingerprint template information .

[XML protocol]

Request Xml:

```
        <ClearData>
                 <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
                 <Arg>
                 <Value xsi:type="xsd:integer”>2</Value>
                 </Arg>
```

```
          </ClearData>
```

Response Xml:

```
   <ClearDataResponse>
            <Row>
                <Result>Succeed! </Result>
        </Row>
   </ClearData>
```

[parameter]

ComKey: communication password

Value:operation value

[return value]

Return True if it is successful, or return False..

4.1.3 record management

The record mainly includes attendance log, which can only be downloaded, and every record is read from attendance log.

4.1.3.1 obtain all log information

[function]

Read attendance log from attendance recorder.

[XML protocol]

Request Xml:

```
        <GetAttLog>
                    <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
                    <Arg>
                        <PIN xsi:type="xsd:integer”>Job Number</PIN>
                  </Arg>
          </GetAttLog>
```

Response Xml:

```
   <GetAttLogResponse>
            <Row>
                <PIN>XXXXX</PIN>
                <DateTime >YYYY-MM-DD HH:MM:SS</DateTime>
                <Verified>X</Verified>
                <Status>X</Status>
                <WorkCode>XXXXX</WorkCode>
        </Row>
   </GetAttLogResponse>
```

[parameter]

ComKey: communication password

PIN: user number(enrollment number).

[return value]

Return log information if it is successful, or return blank.

Pin: user number(enrollment number).

DateTime:date time .

Verified:verification mode.

Status:attendance state .

WorkCode:work code

4.1.3.2 obtain log information at the real time

[function]

Read attendance log from attendance recorder at the real time.

[XML protocol]

Request Xml:

```
<GetLastAttLog>
        <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
</GetLastAttLog>
```

Response Xml:

```
<GetAttLogResponse>
        <Row>
            <PIN>XXXXX</PIN>
            <DateTime >YYYY-MM-DD HH:MM:SS</DateTime>
            <Verified>X</Verified>
            <Status>X</Status>
            <WorkCode>XXXXX</WorkCode>
     </Row>
</GetAttLogResponse>
```

[parameters]

ComKey: communication password

[return value]

Return log information if it is successful, or return blank.

Pin: user number(enrollment number).

DateTime:date time .

Verified:verification mode.

Status:attendance state .

WorkCode:work code

4.1.3.3 clear all log information

[function]

clear all log information .

[XML protocol]

Request Xml:

```
<ClearData>
        <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
```

```
                        <Arg>
                        <Value xsi:type="xsd:integer”>1</Value>
                        </Arg>
           </ClearData>
```

Response Xml:

```
        <ClearDataResponse>
                  <Row>
                      <Result>Succeed! </Result>
              </Row>
        </ClearData>
```

[parameter]

ComKey: communication password

Value:operation value

[return value]

Return True if it is successful,or return False..

4.1.4 system data management

4.1.4.1 update information

[function]

Refresh new data.

[XML protocol]

Request Xml:

```
            <RefreshDB>
                        <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
           </RefreshDB>
```

Response Xml:

```
        <DeleteUserResponse>
                  <Row>
                      <Result>Succeed! </Result>
              </Row>
        </DeleteUserResponse>
```

[parameters]

ComKey: communication password

[return value]

Return True if it is successful, or return False..

Explanation:

Call the function after user information or fingerprint is uploaded to make all modifications come into effect.

4.2 set device

4.2.1 restart device

[function]

Restart device.

[XML protocol]

Request Xml:

```
        <Restart>
                 <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
       </Restart>
```

Response Xml:

```
     <RestartResponse>
            <Row>
               <Result>Succeed! </Result>
          </Row>
     </RestartResponse>
```

[parameter]

ComKey: communication password

[return value]

Return True if it is successful,or return False..

4.2.2 obtain device parameter

[function]

Obtain device parameter .

[XML protocol]

Request Xml:

```
        <GetOption>
                 <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
                 <Arg>
                     <Name xsi:type="xsd:string”>Option Item Name</Name>
            </Arg>
       </GetOption>
```

Response Xml:

```
     <GetOptionResponse>
            <Row>
               <Name>XXXX</Name>
               <Value>XXXX</Value>
          </Row>
      </GetOptionResponse>
```

[parameters]

ComKey: communication password

Name:parameter name.

[return value]

Return parameter name and value.

Name:parameter name,

Value:parameter value.

4.2.3 obtain device\ terminal time

[function]

obtain device\ terminal time.

[XML protocol]

Request Xml:

```
        <GetDate>
                  <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
      </GetDate>
```

Response Xml:

```
   <GetDateResponse>
            <Row>
               <Date>YYYY-MM-DD</Date>
               <Time>HH:MM:SS</Time>
         </Row>
     </GetDateResponse>
```

[parameter]

ComKey: communication password

[return value]

Return device\ terminal time.

Date:date

Time:time

4.2.4 set device parameter

[function]

set device parameter.

[XML protocol]

Request Xml:

```
        <SetOption>
                  <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
                  <Arg>
                   <Name xsi:type="xsd:string”>Item Name</Name>
```

```
                <Value xsi:type="xsd:string”>Item Value</Value>
            </Arg>
        </SetOption>
```

Response Xml:

```
    <SetOptionResponse>
            <Row>
                <Result>Succeed! </Result>
          </Row>
    </SetOptionResponse>
```

[parameter]

ComKey: communication password

Name:parameter name,

Value:parameter value.

[return value]

Return True if it is successful,or return False..

4.2.5 set device\ terminal time

[function]

set device\ terminal time.

[XML protocol]

Request Xml:

```
        <SetDate>
                <ArgComKey xsi:type="xsd:integer”>ComKey</ArgComKey>
                <Arg>
               <Date xsi:type="xsd:string”>YYYY-MM-DD</Date>
               <Time xsi:type="xsd:string”>HH:MM:SS</Time>
            </Arg>
        </SetDate>
```

Response Xml:

```
    <SetDateResponse>
            <Row>
                <Result> Succeed! </Result>
          </Row>
    </SetDateResponse>
```

[parameter]

ComKey: communication password

Date:date

Time:time .

[return value]

Return True if it is successful, or return False..

5. solutions to common problems

5.1 how to create user online

Use function SetuserInfo to write user record for device, such as enrollment number, password, name and so on.

5.2 obtain all user all information

Use GetUserInfo function to get user information. And user GetUserTemplate function to get fingerprint template in string.

5.3 device connection

The device can be regarded as an independent PC during connection. Device IP address must be in accordance with that of connected device.