

# Fingerex for Windows 3.8 Programmer's Guide

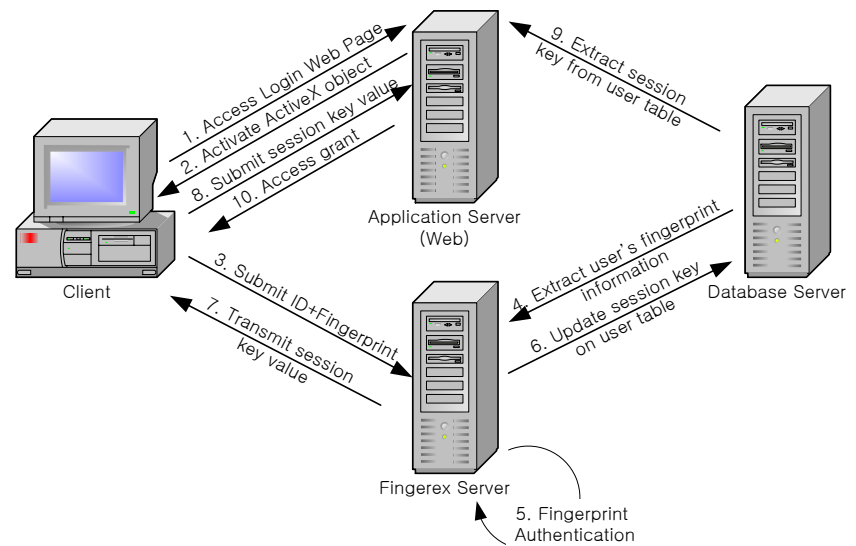
## Table of Contents

<b>BEFORE BEGINNING .....</b>	<b>3</b>
<b>WEB BASED OPERATION FLOW .....</b>	<b>3</b>
<b>CLIENT BASED OPERATION FLOW .....</b>	<b>4</b>
<b>DEVELOPMENT TOOL REQUIREMENT .....</b>	<b>4</b>
<b>CLIENT IDL DESCRIPTION .....</b>	<b>5</b>
<b>WEB BASED SERVICE INTEGRATION .....</b>	<b>9</b>
<b>AUTHUSER.HTM .....</b>	<b>9</b>
<b>CLIENT/SERVER BASED SERVICE INTEGRATION .....</b>	<b>13</b>
<b>ACTIVEX COM OBJECT .....</b>	<b>19</b>
<b>LIBRARY DESCRIPTION .....</b>	<b>19</b>
<b>INTERFACE DFXCFCCTRL_V38 .....</b>	<b>20</b>
<b>INTERFACE IDFXCUTIL1_V38 .....</b>	<b>27</b>
<b>EXAMPLES 1 - CHECK USER .....</b>	<b>29</b>
<b>EXAMPLE2 – AUTHUSER .....</b>	<b>31</b>
<b>EXAMPLE3 – REGISTER USER .....</b>	<b>33</b>
<b>CLIENT ERROR CODE .....</b>	<b>34</b>
<b>TROUBLESHOOTING.....</b>	<b>35</b>

## Before Beginning

Before starting programming with Fingerex server, you should familiar with concept and operation flow of engine. Following diagram explains detailed flow;

### Web Based Operation Flow



- 1 Client access login web page which contain ActiveX object
- 2 ActiveX object asks for fingerprint scan
- 3 Scanned fingerprint information along with user

identification is transfer to Fingerex server

- 4 Fingerex server extracts user's information from database
- 5 Fingerex server matches two fingerprint information
- 6 If match, Fingerex server updates one time password value into database
- 7 Fingerex server sends OTP back to client's ActiveX
- 8 Client submit OTP value along with other user information to web server
- 9 Web server should extract user information from database and compare its OTP value.
- 10 If match, web server grants user access

### **Client Based Operation Flow**

Only difference with web version is that ActiveX object is initiated within client application.

### **Development Tool Requirement**

For client/server development, although Fingerex's ActiveX can be used in any development environment that utilizes ActiveX object, sample project can be opened only with

Microsoft Visual Studio .NET 2005 or later.

For Web development, any web development environment can be used.

## Client IDL Description

```
[
object,
uuid(E9C61B85-E8AF-4558-A3D6-E1311B096D9E),
dual,
helpstring("IIXFPctrl1 Interface"),
pointer_default(unique)
]
interface IIXFPctrl1 : IDispatch
{
typedef enum
{
AUTH_1_1 = 0,
AUTH_1_N = 1
}FXC_EnumAuthType;

typedef enum
{
FX_C_ERROR_SUCCESS           = 0,
FX_C_ERROR_READ              = 201,
FX_C_ERROR_MUSTSCAN          = 202,
FX_C_ERROR_NOTCONNECT        = 203,

```

```
FX_C_ERROR_NO_PASSWORD,  
FX_C_ERROR_NO_USERID,  
  
}FXC_EnumErrorCode;  
  
[propput, id(DISPID_BACKCOLOR)]  
HRESULT BackColor([in]OLE_COLOR clr);  
[propget, id(DISPID_BACKCOLOR)]  
HRESULT BackColor([out,retval]OLE_COLOR* pclr);  
[propput, id(DISPID_BORDERCOLOR)]  
HRESULT BorderColor([in]OLE_COLOR clr);  
[propget, id(DISPID_BORDERCOLOR)]  
HRESULT BorderColor([out, retval]OLE_COLOR* pclr);  
[propput, id(DISPID_BORDERWIDTH)]  
HRESULT BorderWidth([in]long width);  
[propget, id(DISPID_BORDERWIDTH)]  
HRESULT BorderWidth([out, retval]long* width);  
[propput, id(DISPID_FORECOLOR)]  
HRESULT ForeColor([in]OLE_COLOR clr);  
[propget, id(DISPID_FORECOLOR)]  
HRESULT ForeColor([out,retval]OLE_COLOR* pclr);  
[propput, id(DISPID_ENABLED)]  
HRESULT Enabled([in]VARIANT_BOOL vbool);  
[propget, id(DISPID_ENABLED)]  
HRESULT Enabled([out,retval]VARIANT_BOOL* pbool);  
[propput, id(DISPID_BORDERVERISIBLE)]  
HRESULT BorderVisible([in]VARIANT_BOOL vbool);  
[propget, id(DISPID_BORDERVERISIBLE)]  
HRESULT BorderVisible([out, retval]VARIANT_BOOL* pbool);
```

```
[id(1), helpstring("method Scan")] HRESULT Scan([out, retval]FXC_EnumErrorCode
*pVal);
[propget, id(2), helpstring("property userID")] HRESULT userID([out, retval]
BSTR *pVal);
[propput, id(2), helpstring("property userID")] HRESULT userID([in] BSTR
newVal);
[propget, id(3), helpstring("property userPassword")] HRESULT
userPassword([out, retval] BSTR *pVal);
[propput, id(3), helpstring("property userPassword")] HRESULT
userPassword([in] BSTR newVal);
[propget, id(4), helpstring("property AuthType")] HRESULT AuthType([out,
retval] FXC_EnumAuthType *pVal);
[propput, id(4), helpstring("property AuthType")] HRESULT AuthType([in]
FXC_EnumAuthType newVal);
[propget, id(5), helpstring("property UserGUID")] HRESULT UserGUID([out,
retval] BSTR *pVal);
[propput, id(5), helpstring("property UserGUID")] HRESULT UserGUID([in] BSTR
newVal);
[propget, id(6), helpstring("property userName")] HRESULT userName([out,
retval] BSTR *pVal);
[propput, id(6), helpstring("property userName")] HRESULT userName([in] BSTR
newVal);
[id(7), helpstring("method Registration")] HRESULT
Registration([out,retval]FXC_EnumErrorCode *pVal);
[propput, id(8), helpstring("property ServerIPAddress")] HRESULT
ServerIPAddress([in] BSTR newVal);

[propput, id(9), helpstring("property serverPort")] HRESULT serverPort([in]
long newVal);
```

```
[propget, id(10), helpstring("property ResultCode")] HRESULT ResultCode([out,
retval] short *pVal);
[id(11), helpstring("method Authentication")] HRESULT Authentication([out,
retval]FXC_EnumErrorCode *pVal);
};

[
uuid(2669CF85-ED56-4A1C-A05F-1CC05DB01B08),
version(1.0),
helpstring("kFXClient 1.0 Type Library")
]
library KFXCLIENTLib
{
importlib("stdole32.tlb");
importlib("stdole2.tlb");

[
uuid(4635F269-5FB0-4BFE-BCD6-94FA00ABFC1A),
helpstring("_IIXFPCtrl1Events Interface")
]

// Event Interface
dispinterface _IIXFPCtrl1Events
{
properties:
methods:
[id(1), helpstring("method OnAuthFinished")] HRESULT OnAuthFinished();
};
```



```
[
uuid(691FB591-2E17-4244-9FF4-A7FCE8B3AC27),
helpstring("IXFPCtrl1 Class")
]
coclass IXFPCtrl1
{
[default] interface IIXFPCtrl1;
[default, source] dispinterface _IIXFPCtrl1Events;
};

[
uuid(FAFCBBD6-5166-468A-806E-20ACE9458F0E),
helpstring("FXC_Comm1 Class")
]
};
```

## Web based service integration

### AuthUser.htm

Following is sample html file that comes with product.

```
<HTML>
<HEAD>
<TITLE>Fingerex Authentication test form</TITLE>
<script language="JavaScript">
```

```
function FX_scan()
{
    var sRet = AUTHFORM.IXFPCtrl1.Scan();
    txtResult.value = sRet;
}
function FX_Auth()
{
    AUTHFORM.IXFPCtrl1.ServerIPAddress = AUTHFORM.SIP.value;
    AUTHFORM.IXFPCtrl1.ServerPort = AUTHFORM.SPORT.value;
    AUTHFORM.IXFPCtrl1.userID = AUTHFORM.uid.value;
    var sRet = AUTHFORM.IXFPCtrl1.Authenticate();
    txtResult.value = sRet;
}
</script>

<SCRIPT ID=clientEventHandlersJS LANGUAGE=javascript>
function IXFPCtrl1_OnCommFinished()
{
    var resultcode = AUTHFORM.IXFPCtrl1.ResultCode;
    txtResult.value = resultcode;
    txtResult.value = txtResult.value + ", " + AUTHFORM.IXFPCtrl1.userID +
    ", " + AUTHFORM.IXFPCtrl1.userName + ", Session:" +
    AUTHFORM.IXFPCtrl1.UserSessionKey;

    alert("Successfully communicate with server\nresult code exist below");
}
</SCRIPT>

<SCRIPT LANGUAGE=javascript FOR=IXFPCtrl1 EVENT=OnCommFinished>
```

```
IXFPCtrl1_OnCommFinished();
</SCRIPT>

</HEAD>
<BODY text=white bgColor=midnightblue>
<P align=center><STRONG><FONT size=4>Fingerex Authentication Test
Form</FONT></STRONG>
<BR></P>
<center>
<form name="AUTHFORM">
<table>
  <TR>
    <td colspan=2 align=middle>
      <OBJECT id=IXFPCtrl1
      classid=CLSID:3CBF2B19-4251-48e4-93AF-86FE28F79096
      codebase="download/FingerexClientV38.cab#version=2006,11,29,321"
      VIEWASTEXT></OBJECT></td>
    </TR>
    <tr>
      <td colspan=2 align=middle><input type=button
onclick="JavaScript: FX_scan();" value="Scan"><td></td>
    <tr>
      <td>User ID</td>
      <td><INPUT name="uid" maxlength=40 size=40 ></td>
    </tr>
    <tr>
      <td colspan=2 align=middle>
        <input type=button value="Authenticate" onclick="JavaScript:
```

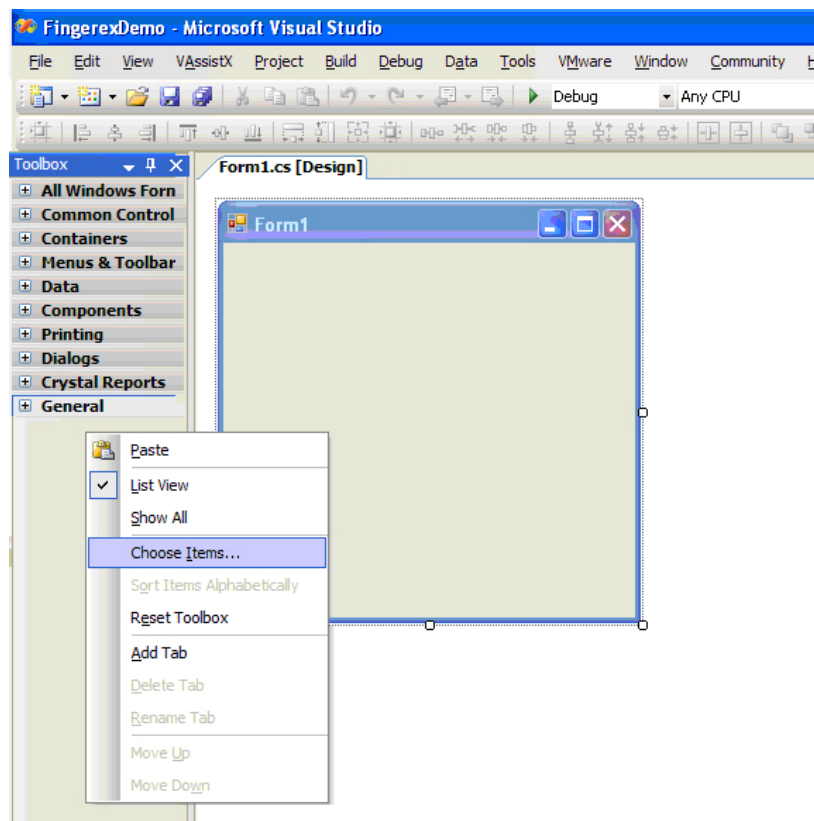
Object Description

```
FX_Auth();" >
        </td>
    </tr>
    <tr>
        <td>ServerIP</td>
        <td><input name="SIP" value="127.0.0.1" ></td>
    </tr>
    <tr>
        <td>Server Port</td>
        <td><input name="SPORT" value="2000" ></td>
    </tr>
</table>
</form>
<HR>Result<BR><input name="txtResult" size=100 ><hr>
</center>
</BODY>
</HTML>
```

## Client/Server based service integration

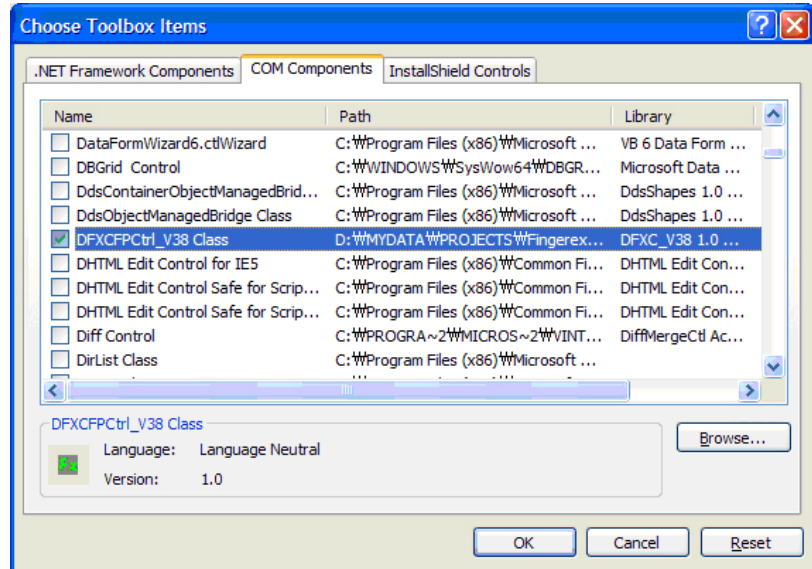
In this chapter, we describe how to make C# application step by step.

First, start Microsoft Visual Studio 2005 or later and create new project with C#.

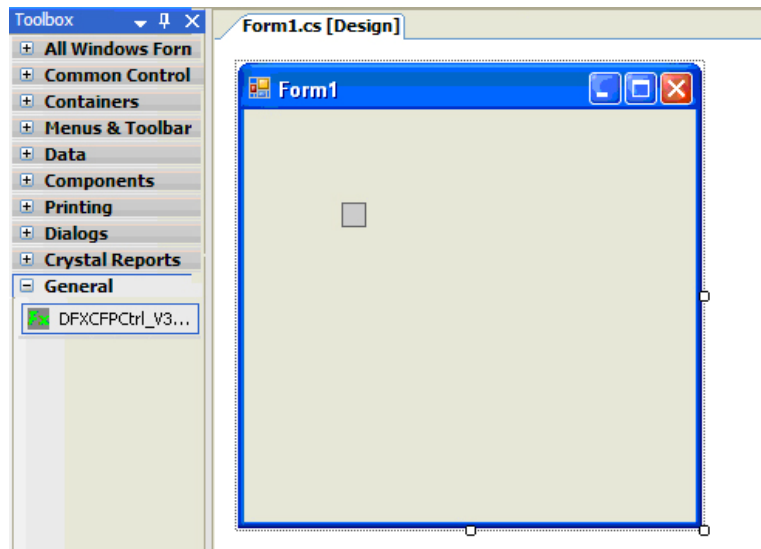


Select “Choose Items” then click. Following dialog box will appear.

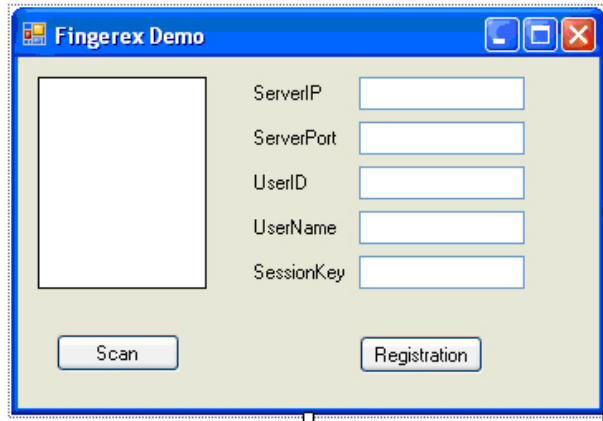
Select DFXCFCtrl\_V38 from COM components.



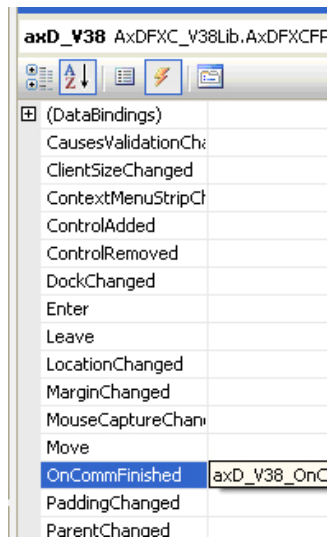
You can now see DFXCFCtrl\_V38 class in toolbox. Drag component into project.



Create following form. You need Server IP, Port, User ID, Name and SessionKey to make it work.



Handle “OnCommFinished” event for DFXCFCtrl\_V38 Class.



```

private void Form1_Load(object sender, EventArgs e)
{
    axD_V38.Size = new Size(102, 128);
}

private void butScan_Click(object sender, EventArgs e)
{
    axD_V38.Scan();
}

private void butReg_Click(object sender, EventArgs e)
{
    axD_V38.ServerIP = txtServerIP.Text;
    axD_V38.ServerPort = int.Parse(txtServerPort.Text);
    axD_V38.UserID = txtUserID.Text;
    axD_V38.UserName = txtUserName.Text;
    axD_V38.UserSessionKey = txtSessionKey.Text;

    axD_V38.Registration();
}

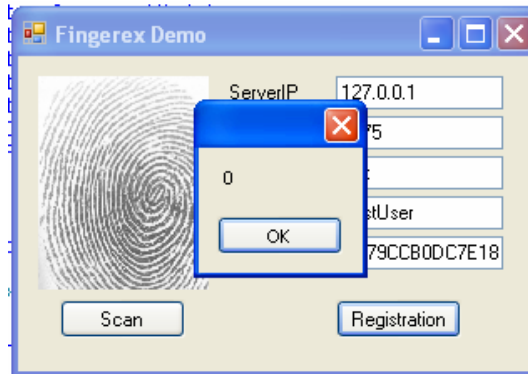
private void axD_V38_OnCommFinished(object sender, EventArgs e)
{
    AxDFXCFCtrl_V38 axD = (AxDFXCFCtrl_V38)sender;
    MessageBox.Show(axD.ResultCode.ToString());
}

```

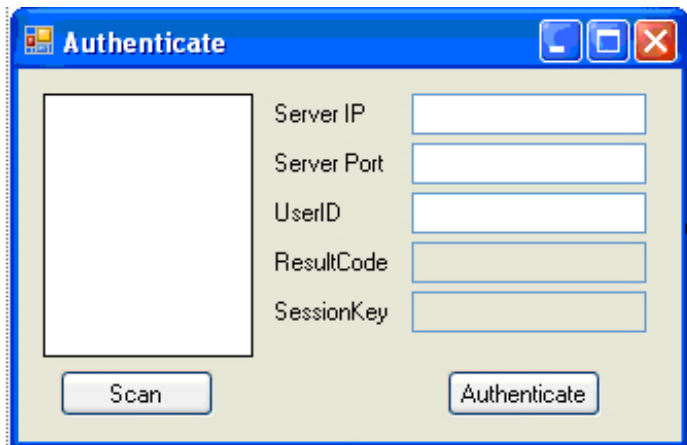
For testing, fill configuration string.







Now we do registration. Let's create authentication utility. Create new application with same as registration.



Handle event as follows;

```
private void Form1_Load(object sender, EventArgs e)
{
    axD_V38.Size = new Size(102, 128);
}

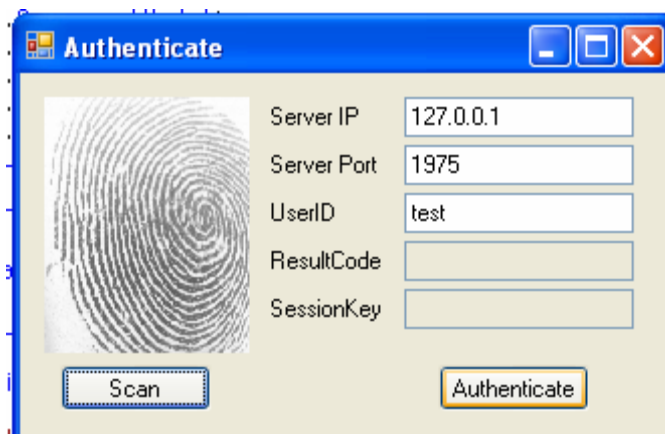
private void butScan_Click(object sender, EventArgs e)
{
    axD_V38.Scan();
}

private void butReg_Click(object sender, EventArgs e)
{
    axD_V38.ServerIP = txtServerIP.Text;
    axD_V38.ServerPort = int.Parse(txtServerPort.Text);
    axD_V38.UserID = txtUserID.Text;
    axD_V38.UserName = txtUserName.Text;
    axD_V38.UserSessionKey = txtSessionKey.Text;

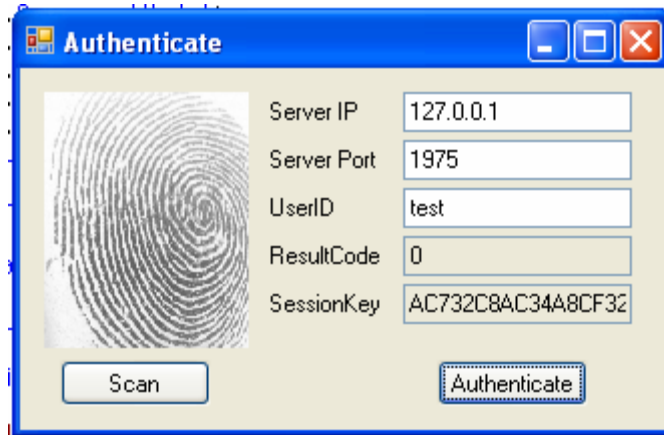
    axD_V38.Registration();
}

private void axD_V38_OnCommFinished(object sender, EventArgs e)
{
    AxDFXCFPCtrl_V38 axD = (AxDFXCFPCtrl_V38)sender;
    MessageBox.Show(axD.ResultCode.ToString());
}
```

Testing



And result. Please note that result code is 0 and we gets session key.



## ActiveX COM Object

### Library Description

```
Library: library DFXCFPCtrl_V38  
  
Interface Object  
interface IDFXCSPCFPCtrl  
interface IDFXCSPCUtill  
  
Event DISPInterface  
dispinterface _DIDFXCFPCtrlEvents_V38  
  
COCLASS  
coclass DFXCSPCFPCtrl  
{
```

```
[default] interface IDFXCFPCtrl_V38;
    [default, source] dispinterface _DIDFXCFPCtrlEvents_V38;
};

coclass DFXCSPCUtl1
{
    [default] interface IDFXCUtl1_V38;
};
```

### Interface DFXCFPCtrl\_V38

Handles low-level communication with Fingerex server and extracts feature information from fingerprint image.

This ActiveX supports IObjectSafety, CComControl, IOleControlImpl and can run on WEB and secure container.

#### Properties

BackColor	OLE_COLOR Background color of fingerprint authentication control
BorderColor	OLE_COLOR Border color of fingerprint authentication control
BorderWidth	Numeric

	Width of fingerprint authentication control's border
Enabled	VARIANT_BOOL Enable/Disable fingerprint authentication control
BorderVisible	VARIANT_BOOL Enable/Disable visibility of fingerprint authentication control's border
ServerIP	IP or DNS Name Server's IP or DNS qualified name
ServerPort	Numeric Port number or server for communication
UserID	Character User ID (As USER_ID on DB)
UserPassword	Character User Password (As USER_PW on DB)
SubResultCode	Internal Error Code Reserved for internal use. Do not use.
UserSessionKey	Character

	R.F.U
UserName	Character R.F.U.
FPFeature	Binary Contains feature list after reading fingerprint.
FPFeatureBSTR	Character Return hex code of feature list. (Require 960Bytes)
ChangeFP	Short Indicate deletion of previous fingerprint information if new one registered.
UserNewPassword	Character R.F.U
FPDesc	Character Description of fingerprint
Info1	Character Return contents of Info1 field in database if authentication is success.
Info2	Character

Return contents of Info2 field in database if authentication is success.

Info3

Character

Return contents of Info3 field in database if authentication is success.

ResultCode

Internal Error Code

Result code for operation

### Methods

Authenticate

FXC\_V38\_COM\_EnumErrorCode

Result code for authentication operation

Registration

FXC\_V38\_COM\_EnumErrorCode

Result code for registration operation

Scan

FXC\_V38\_COM\_EnumErrorCode

Result code for feature extraction of fingerprint operation

MatchingTest

FXC\_V38\_COM\_EnumErrorCode

Result code for fingerprint matching operation

CheckUserExist      FXC\_V38\_COM\_EnumErrorCode  
Result code for checking on user  
existence operation

IsDeviceExisted      0 – Exist, Other-Error  
Result code for device status check  
operation

FXC\_V38\_COM\_EnumErrorCode

FX\_COM\_ERROR\_SUCCESS  
0: Operation successful

FX\_COM\_ERROR\_READ  
201: Error reading fingerprint

FX\_COM\_ERROR\_MUSTSCAN  
202: Fingerprint must be scanned

FX\_COM\_ERROR\_NOCONNECT  
203: Cannot connect Fingerex server

FX\_COM\_ERROR\_NO\_PASSWORD  
204: This operation require password



FX\_COM\_ERROR\_NO\_USERID

205: This operation require user ID

FX\_COM\_ERROR\_CUR\_PROCESSING

206: Operation is already started

FX\_COM\_ERROR\_NETWORK\_CLOSE

207: Server is disconnected

FX\_COM\_ERROR\_NETWORK\_TIMEOUT

208: Server has close connection due to the timeout

FX\_COM\_ERROR\_NO\_HW

210: Cannot find fingerprint scanning device

FX\_COM\_ERROR\_NO\_APPCODE

210: R.F.U.

FX\_COM\_ERROR\_MUSTBESESSIONKEY

220: R.F.U.

Server Communication Error Code

FXSERVER\_ERROR\_SUCCESS

0: Success

FXSERVER\_ERROR\_DB\_NORMAL

1: Database error

FXSERVER\_ERROR\_UNKNOWN\_APPCODE

2: R.F.U.

FXSERVER\_ERROR\_NOTMATCH

3: Fingerprint mismatch

FXSERVER\_ERROR\_NOSUCH\_USER

4: No user exist

FXSERVER\_ERROR\_SESSIONKEYNOTMATCH

5: R.F.U.

FXSERVER\_ERROR\_MAXFINGERREGISTERED

6: R.F.U.

FXSERVER\_ERROR\_PWDNOTMATCH

7: Password mismatch

FXSERVER\_ERROR\_NOFINGERPRINT

8: R.F.U.

FXSERVER\_ERROR\_NOSUCHAPPUSERID

9: R.F.U.

FXSERVER\_ERROR\_ALREADYREGISTEREDBYOTHER  
USER

10: R.F.U.

FXSERVER\_ERROR\_SUCHUSER\_NOFP

20: R.F.U.

### **Interface IDFXCUtl1\_V38**

This interface gives you more high-level functionality. Also gives you fingerprint registration and authentication dialog box.

#### Properties

ServerIP	IP or DNS Name Server's IP or DNS qualified name
ServerPort	Numeric

Port number or server for communication

UserID                      Character  
User ID (As USER\_ID on DB)

UserPWD                    Character  
User Password (As USER\_PW on DB)

DBGubunCode              Number  
Custom database indicator. Must consult  
with Digent in order to use it.

#### Methods

RegisterUser              User registration dialog  
Display user registration dialog and  
insert information into database.  
0: Success

AuthUser                    User Authentication dialog  
Display user Authentication dialog and  
compare information with database.  
0: Success

## Examples 1 - Check User

```
<HTML>
<HEAD>
<meta http-equiv="content-type" content="text/html; charset=euc-kr">
<SCRIPT LANGUAGE="JavaScript">
function CheckUser()
{
    FPCTL1.ServerIPAddress = "127.0.0.1";
    FPCTL1.ServerPort = 1978;
    FPCTL1.AuthType = 0;
    FPCTL1.DBGubunCode=1;
    var result = FPCTL1.CheckUserExist(FRM.USERID.value,
FRM.USERPWD.value);
}
</SCRIPT>
<SCRIPT ID=clientEventHandlersJS LANGUAGE=javascript>
function FPCTL1_OnAuthFinished()
{
    var resultcode = FPCTL1.ResultCode;

    alert("Communication end");
    alert(resultcode + ", " + "SABUN:" + FPCTL1.UserName);
}
</SCRIPT>

<SCRIPT LANGUAGE=javascript FOR=FPCTL1 EVENT=OnCommFinished>
    FPCTL1_OnAuthFinished();
```

```
</SCRIPT>
</HEAD>
<BODY>
BASIC Function1: Check User<BR>
<form name="FRM">
USERID:<input type="text" NAME="USERID"><BR>
PASSWORD:<input type="text" NAME="USERPWD"><BR>
Check User : <input type="button" value="Check User" OnClick="JavaScript:
CheckUser();">
</form>

<object ID=FPCTL1 classid=CLSID:31C3D480-F087-4a52-9866-CB60DA2A6516></object>
</BODY>
</HTML>
```

## Example2 – AuthUser

```
<HTML>
<HEAD>
<meta http-equiv="content-type" content="text/html; charset=euc-kr">
<SCRIPT LANGUAGE="JavaScript">
function AuthUser()
{
    FPCTLUTIL1.ServerIP = "127.0.0.1";
    FPCTLUTIL1.ServerPort = 1978;
    FPCTLUTIL1.DBGubunCode = 1;

    // Following value set to authentication dialog box to default
    FPCTLUTIL1.UserID = "ABC";

    var result = FPCTLUTIL1.AuthUser();
    if(result == 0)
    {
        alert("Authentication Success");
        alert("Result:" + FPCTLUTIL1.UserID + ", " + FPCTLUTIL1.UserPWD
+ ", " + FPCTLUTIL1.UserSabunNM);
    }
}
</SCRIPT>

</HEAD>
<BODY>
BASIC Function2: Authenticate User<BR>
<form name="FRM">
```

```
Auth User : <input type=button value="User Check" OnClick="JavaScript:
AuthUser();" >
</form>

<object ID=FPCTLUTIL1 classid=CLSID:B7B26320-B7AD-45B5-A473-
59A172C23265></object>

</BODY>
</HTML>
```



### Example3 – Register User

```
<HTML>
<HEAD>
<meta http-equiv="content-type" content="text/html; charset=euc-kr">
<SCRIPT LANGUAGE="JavaScript">
function RegisterUser()
{
    FPCTLUTIL1.ServerIP = "127.0.0.1";
    FPCTLUTIL1.ServerPort = 1978;
    FPCTLUTIL1.DBGubunCode = 1;

    var result = FPCTLUTIL1.RegisterUser();
}
</SCRIPT>

</HEAD>
<BODY>
BASIC Function1: Register User<BR>
<form name="FRM">
Register User : <input type=button value="User Register" OnClick="JavaScript:
RegisterUser();">
</form>

<object ID=FPCTLUTIL1 classid=CLSID:B7B26320-B7AD-45B5-A473-
59A172C23265></object>
</BODY>
</HTML>
```

## Client Error Code

Value	Definition
1001	Database busy
1002	Database Connection Failure
1003	Password Mismatch
1004	Connection string error when connecting to Database
1005	Authentication Failure. User ID does not exist. Matching Fingerprint does not exist.
1006	Maximum User Over
0	Success (Registration & Authentication)
201	Fingerprint Reading Failure
202	Missing Fingerprint Information
203	Fingerex Server Connection Failure (Check IP or PORT)
204	Missing Password (Registration)
205	Missing User ID (Registration & Authentcation-1:1 matching)
206	Server is working.
207	Network Error
208	Server Time Out
1006	Reached Max Capacity

## Troubleshooting

Q. How to check if Fingerex server is started?

A. Check if port is in use. See following screenshot.

```
O:\W>netstat -an | FIND "1975"
O:\W>netstat -an | FIND "80"
TCP    0.0.0.0:80           0.0.0.0:0           LISTENING
TCP    0.0.0.0:19801       0.0.0.0:0           LISTENING
TCP    0.0.0.0:29801       0.0.0.0:0           LISTENING
TCP    127.0.0.1:1941      127.0.0.1:19801     ESTABLISHED
TCP    127.0.0.1:19801     127.0.0.1:1941      ESTABLISHED
TCP    210.122.226.163:2317 66.249.89.147:80    CLOSE_WAIT
TCP    210.122.226.163:2742 207.46.16.246:80    TIME_WAIT
TCP    210.122.226.163:2751 218.239.45.195:80    CLOSE_WAIT
TCP    210.122.226.163:2752 218.239.45.195:80    CLOSE_WAIT
TCP    210.122.226.163:2754 211.115.123.215:80    CLOSE_WAIT
TCP    210.122.226.163:2757 58.229.121.166:80    CLOSE_WAIT
TCP    210.122.226.163:2759 218.239.45.195:80    CLOSE_WAIT
TCP    210.122.226.163:2765 207.46.16.246:80    ESTABLISHED
TCP    210.122.226.163:2766 207.46.16.246:80    ESTABLISHED
```

Q. Cannot read user from Fingerex server.

A. Make sure database connection is correct.

Q. Cannot register user.

A. Check if Database is connected correctly.

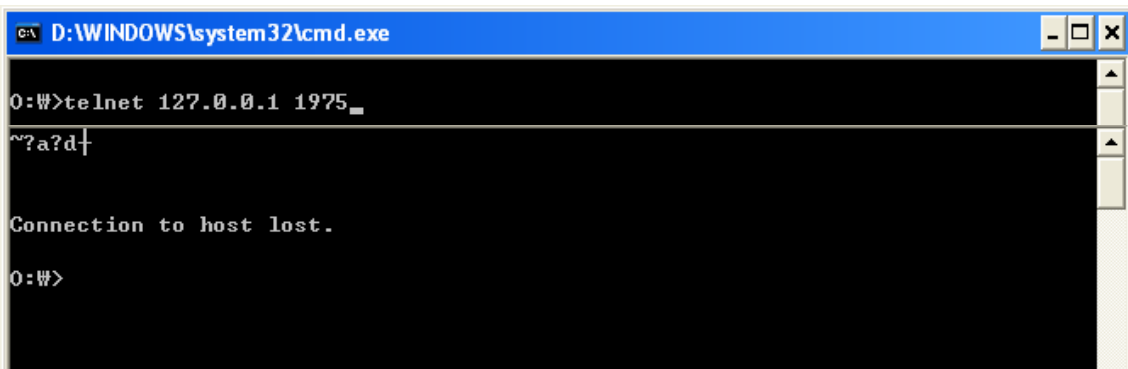
Q. Can't connect to server.

A. Check server's IP and port from client setting. Also, check

if there is firewall.

Q. I can connect to server's host but cannot connect to server.

A. Check if server is working. See following screenshot.



Q. Cannot get fingerprint scanned.

A. Check if device driver is installed. See following screenshot.

From

“MyComputer->Properties->Hardware->Drive Manager”,

