

指纹识别开发包使用说明书

介绍 (Introduction)

这个文档用来描述指纹软件开发包及怎么使用这个指纹开发包。

指纹开发包文档包包含以下几个部分

1. 这个文档 (包括 COM API 和 DLL API 的说明及部分样例代码等)
2. COM 组件 (Efatcom.dll) 提供指纹模板的登记, 识别等功能, 该 COM 组件必须在指纹仪的支持下使用, 如果没有指纹设备, COM 组件将不会正常运行。
3. DLL 动态库 (ServerMatcher.dll) 只提供指纹模板的比对识别功能, 不需要指纹设备的支持。
4. 样例代码 (VB.net 样例, Delphi 样例, 服务器 ASP 代码样例)

指纹识别开发包支持本地和 WEB 网络上指纹比对及传输。

本地程序中调用中使用自定义消息的方式来完成一个指纹处理过程。所有处理结果都用自定义消息的方式返回。

在 WEB 网络中, 则通过与服务器端的通讯来完成一个指纹处理过程。在 WEB 网络中, 指纹传输的数据格式是以一个 XML 文档的形式来实现的, XML 文档中的指纹数据都经过加密处理。指纹传输是以 XMLHTTP 协议来实现的。并且 WEB 中传输中, 支持自定义数据的传输。

开发包组件支持平台说明

指纹软件开发包目前只支持 Windows2000、Windows2003、Windows XP、Windows Vista 等 Windows32 位操作系统, 如果需要支持 64 位 Windows 系统或 WindowCE 嵌入式操作系统, 请与我们联系。暂不提供 Unix、Linux 等系统下的开发包。

指纹软件开发包支持所有支持 COM 组件调用的开发工具, 典型的如: Microsoft Visual C++、Microsoft Visual Basic、Delphi 等。指纹软件开发包也支持嵌入到 WEB 页中, WEB 服务器端的 ASP 服务中。

COM 组件注册 (COM Registration)

在你的程序中使用指纹 COM 组件前, 必须先要在操作系统中注册这个 COM 组件。

注册命令如下: Regsvr32.exe Efatcom.dll

如果下取消注册, 则命令如下: Regsvr32.exe Efatcom.dll -u

定义和术语

Binary Large Object (BLOB): 数据库中用于存取模板的可以存储大尺寸数据的数据类型。

Enrollment: 这是在数据库中为用户建立和存储参考模板的过程。首先从指纹传感器上收集一个或多个图像, 提取显著的特征数据, 然后合并结果得到一个模版。

Identification: 一个指纹与数据库中的模板比较来确认身份, 也就是匹配操作, 也就是, 名字和手指头号码和指纹关联。

Image Item: 使用的 SDK 传入的图像数据和接受的自应用程序的图像数据是一个不透明的图像项目。一个图像项目包含像素数据, 尺寸数据和其他 SDK 专有数据。这个 SDK API 包含的函数能在应用程序中使用, 可以从一个图像项目中提取重要数据。

Match Template: 从传感器上获得的一个或者多个图像来建立模板, 用来鉴定操作。在鉴定期间, 匹配模板与参考模板相比较来确定它们是否表现相同的指纹。

Reference Template: 在登记程序期间创建的模板。

Validation: 一个指定的识别情形。一个指纹与 SDK 数据库中的指定的用户进行匹配, 如果它是匹配, 则得到确认信息。

Verification: 逆着应用程序提供的模板列表匹配一个指纹的匹配操作。应用程序将使用一个验证操作, 当它继续参考自己的数据库里的样板时。

指纹识别开发包使用说明书

COM API 接口参考

以下我们将介绍 COM API 的函数，属性，常量定义。具体函数调用及参数在下几章用例子来说明。

函数(Public Member Functions)

HRESULT SetMainWnd(long newVal);

作用： 设置接受自定消息的窗口句柄，这个函数在本地应用程序中必须使用

参数： newVal 窗口的句柄 HWND

返回值： S_OK 成功，否则失败

HRESULT StartTransaction(short workmode);

作用： 开始指纹处理过程

参数： workmode 用户要处理的类型，具体定义在常量列表中

返回值： S_OK 成功，否则失败

HRESULT EndTransaction();

作用： 结束指纹处理

参数： 无

返回值： S_OK 成功，否则失败

HRESULT CalibrateSensor();

作用： 校准指纹传感器

参数： 无

返回值： S_OK 成功，否则失败

HRESULT SaveImage(BSTR pFileName);

作用： 保存指纹图像到一个文件

参数： pFileName 指定的文件名

返回值： S_OK 成功，否则失败

HRESULT GetMatchTemplate(BYTE *pMatVal,long *pSizeVal,[out, retval]BOOL *bRet);

作用： 获取匹配模板

参数： pMatVal,用于存储匹配模板的内存空间，必须足够的大，比如设置为 20480

pSizeVal,匹配模板的大小

bRet,函数实例的返回值

返回值： S_OK 成功，否则失败

HRESULT SetMatchTemplate(BYTE *newMatVal,long newSizeVal,[out, retval]BOOL *bRet);

作用： 设置匹配模板

参数： newMatVal,匹配模板的内存地址

newSizeVal,匹配模板的大小

bRet,函数实例的返回值

返回值： S_OK 成功，否则失败

HRESULT GetReferenceTemplate(BYTE *pRefVal,long *pSizeVal,[out, retval]BOOL *bRet);

作用： 获取参考模板

参数： pRefVal,用于存储参考模板的内存空间，必须足够的大，比如设置为 8096

pSizeVal,匹配模板的大小

bRet,函数实例的返回值

返回值： S_OK 成功，否则失败

HRESULT SetReferenceTemplate(BYTE *newRefVal,long newSizeVal,[out,

指纹识别开发包使用说明书

retval]BOOL *bRet);

作用： 设置参考模板

参数： newRefVal, 参考模板的内存地址
newSizeVal, 匹配模板的大小
bRet, 函数实例的返回值

返回值： S_OK 成功， 否则失败

HRESULT **VerifyTemplateInner**([out, retval]BOOL *bRet);

作用： 使用在 SDK 调用时设置的参考模板和匹配模板间进行比对

参数： bRet, 函数实例的返回值

返回值： S_OK 成功， 否则失败

HRESULT **VerifyTemplateOneToOne**(BYTE *RefVal, long RefSize, BYTE

*MatVal, long MatSize, [out, retval]BOOL *bRet);

作用： 从外部指定一个参考模板和一个匹配模板进行比对

参数： RefVal, 参考模板的内存地址

RefSize, 参考模板的大小

MatVal, 匹配模板的内存地址

MatSize, 匹配模板的大小

bRet, 函数实例的返回值

返回值： S_OK 成功， 否则失败

HRESULT **ClearItems**();

作用： 清除添加的自定义项

参数： 无

返回值： S_OK 成功， 否则失败

HRESULT **AddTextItem**([in] BSTR newElement, [in] BSTR newVal);

作用： 添加一个文本类型的数据项

参数： newElement 数据项名称

newVal 数据项值

返回值： S_OK 成功， 否则失败

HRESULT **AddIntItem**([in] BSTR newElement, [in] long newVal);

作用： 添加一个整形类型的数据项

参数： newElement 数据项名称

newVal 数据项值

返回值： S_OK 成功， 否则失败

属性(Properties)

int **ShowStatusText** [Get,Set]

是否在控件窗口中显示状态提示， 1 表示显示， 0 表示不显示

BSTR **UserName** [Get,Set]

指定用户名， 这个属性主要用于 WEB 应用中， 用户传输编号， 下载模板

int **ResultID** [Get]

指纹处理后的返回值

BSTR **StatusText** [Get]

状态提示文本

int **EnrollCount** [Get,Set]

设置登记次数， 最少 3 次， 对多 8 次

int **SoundHint** [Get,Set]

是否启动声音提示， 1 有声音提示， 0 无声音提示， 提示的声音有下面的 Sound 开头的属性来设置

BSTR **SoundProcess** [Get,Set]

设置按手指过程中的声音提示文件

BSTR **SoundEnrollSucceed** [Get,Set]

设置登记成功后的声音提示文件

指纹识别开发包使用说明书

BSTR SoundEnrollFail [Get,Set]	设置登记失败后的声音提示文件
BSTR SoundIdentifySucceed [Get,Set]	设置识别成功后的声音提示文件
BSTR SoundIdentifyFail [Get,Set]	设置识别失败后的声音提示文件
int TimeOut [Get,Set]	设置超时时间, 0 表示不超时, 1 表示 10 秒, 2 表示 20 秒, 以此类推

常量(Const)

以下为 Workmode	
#define TRANS_CAPTUREIMAGE	0 采集指纹图像
#define TRANS_ENROLLTOWEB	1 登记指纹模板, 然后上传至 WEB 服务器(会要求登记主从两个手指的指纹模板)
#define TRANS_VALIDATEFORMWEB	2 从服务器下载指定用户的模板, 然后在本机计算机识别
#define TRANS_IDENTIFYTOWEB	3 上传匹配模板至服务器, 在 WEB 服务器端进行识别
#define TRANS_ENROLLINNER	4 在本机计算机登录指纹模板, 用户自己决定模板存储到的位置
#define TRANS_VALIDATEINNER	5 在本地计算机上识别(验证)指纹模板, 参考模板必须在开始处理之前设置
以下为自定义消息	
#define WM_FPSMSG	WM_USER+800 Windows 窗口自定消息
#define FM_SHOW_STATUS	0 WPARAM 参数, 状态更新提示, LPARAM 是状态文本
#define FM_SHOW_NEWIMAGE	1 WPARAM 参数, 显示指纹图像
#define FM_CAPTURE_NEWIMAGE	2 WPARAM 参数, 采集到新的指纹图像
#define FM_ENROLL_COMPLETE	3 WPARAM 参数, 登记完成后, LPARAM 参数低字节为 1 时表示登记成功, 否则登记失败
#define FM_IDENTIFY_COMPLETE	4 WPARAM 参数, 识别完成后, LPARAM 参数低字节为 1 时表示识别成功, 否则识别失败
#define FM_VERIFY_COMPLETE	5 WPARAM 参数, 验证完成后, LPARAM 参数低字节为 1 时表示验证成功, 否则验证失败
#define FM_FOCUS_LOST	14 设备被其它程序占用
#define FM_TIMEOUT	15 设备超时

DLL API 接口参考

以下我们将介绍 DLL API 的函数。

BOOL StartMatcher ();	说明: 开始指纹匹配器 参数: 无 返回值: TRUE, 成功, FALSE 失败
VOID CloseMatcher ();	说明: 关闭指纹匹配器 参数: 无 返回值: TRUE, 成功, FALSE 失败

指纹识别开发包使用说明书

```
BOOL VerifyTemplateOneToOne(BYTE pMatchTemp[],UINT nMatchSize,BYTE  
pRefTemp[],UINT nRefSize);
```

说明：1：1 识别参考模板和匹配模板

参数：pMatchTemp[]，匹配模板 nMatchSize，匹配模板大小
pRefTemp[]，参考模板 nRefSize，参考模板大小

返回值：TRUE，识别成功，FALSE 识别失败

建立指纹应用程序的基本步骤

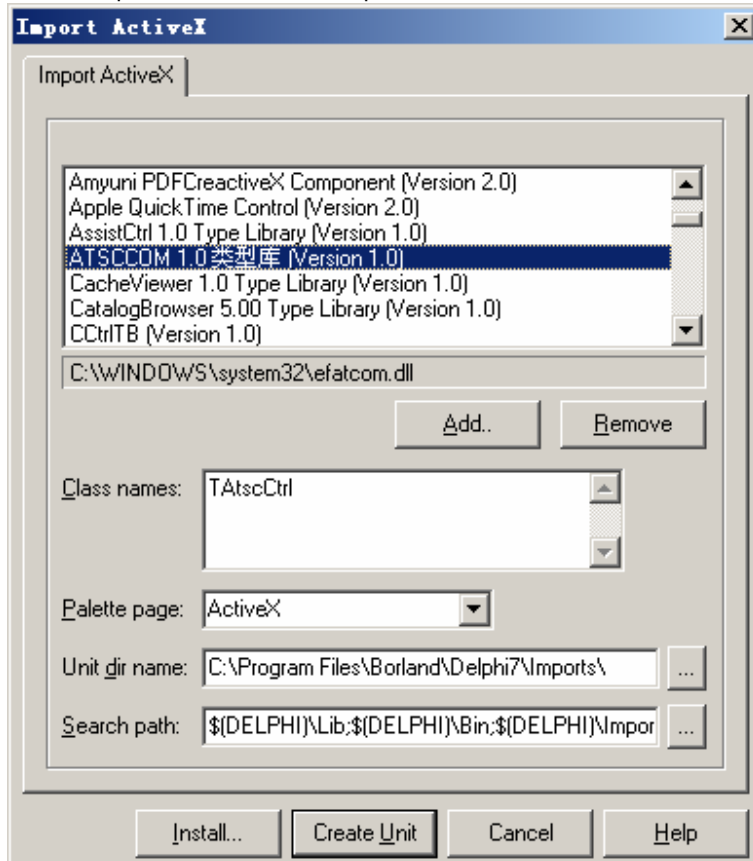
1. 导入类型库(ActiveX),开发工具自动产生调用代码
2. 新建一个工程
3. 在主窗体创建或主窗体载入事件中，添加代码
`SetMainWnd(hWnd)`
设置组件发送自定义消息的目标窗口
4. 添加这个窗口消息的自定义消息处理代码
`LRESULT CALLBACK WindowProc(HWND hwnd,UINT uMsg,WPARAM wParam,LPARAM
lParam);`
Win32 SDK 中直接加入添加到上面函数中，其他的开发工具中，可能需要重载或映射
5. 开始一个指纹处理过程
`StartTransaction(0)`
运行这个函数后，设备等待按手指，采集到指纹图像后，将发送自定义消息到指定的窗口，窗口收到自定义消息后，用户即可以按自己的意愿来处理数据了。收到自定义消息后，这个指纹处理过程就自动完成。
6. 结束一个指纹处理过程
`EndTransaction()`
结束一个正在工作的指纹处理过程
7. 以上是基本指纹处理流程,运用得当，可以完成任意的指纹应用。

指纹识别开发包使用说明书

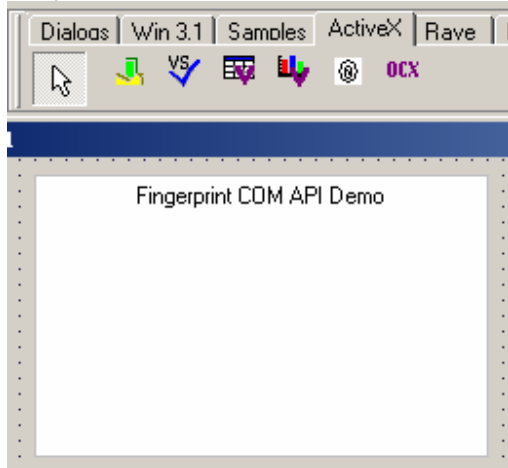
Delphi 应用样例 (Delphi 7 版本)

1. 安装组件

点击 Delphi7 菜单中的 Component->Install ActiveX Control ...弹出



选中 ATSCCOM 1.0 类型库，按后点击 Install ..., 根据提示完成 Delphi 的组件安装，安装完后在 Delphi 的组件面板 ActiveX 栏会出现图标

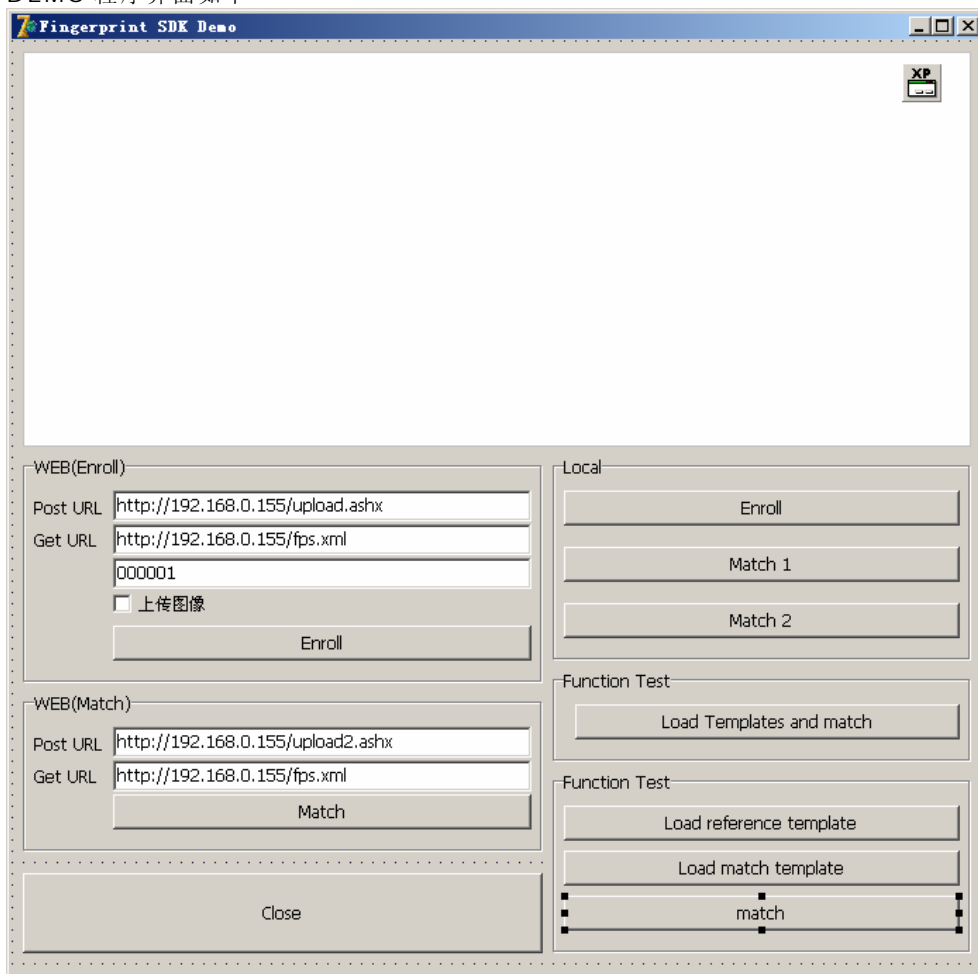


拖动这个图标到窗体上即可。

要查看 Delphi 中类型库的详细内容，可以打开 ATSCCOMLib_TLB.pas 文件，该文件由 Delphi 自动生成，在 Delphi 的 Imports 目录下可以找到；

指纹识别开发包使用说明书

2. DEMO 程序界面如下



3. DEMO 程序指纹处理部分

添加自动消息，用于接收指纹处理消息

在 Unit1.pas 文件的 Type 下的 Private 出添加代码

```
procedure WMFPSMessage(var Msg:TMessage); message WM_FPSMSG;
```

在 Unit1.pas 文件的 Implementation 下添加

```
procedure TForm1.WMFPSMessage(var Msg:TMessage);
```

```
var
```

```
  FpsMsg: DWORD;
```

```
  RetHigh: WORD;
```

```
  RetLow: WORD;
```

```
  fs: TFileStream;
```

```
  revbuf: array[0..20480]of BYTE;
```

```
  revsize: integer;
```

```
begin
```

```
  FpsMsg:=Msg.WParam;
```

```
  RetLow:=msg.LParamLo;
```

```
  RetHigh:=msg.LParamHi;
```

```
  case FpsMsg of
```

```
    FM_CAPTURE_NEWIMAGE:
```

```
      begin
```

```
        atsctrl1.GetMatchTemplate(revbuf[0],revsize);
```

```
        fs:=TFileStream.Create(m_apptah+'match.dat',fmCreate);
```

```
        fs.Position:=0;
```

```
        fs.Write(revbuf,revsize);
```

```
        fs.Free;
```

指纹识别开发包使用说明书

```
    if AtscCtrl1.VerifyTemplateInner=1 then
    begin
        ShowMessage('Identify succeed');
    end else
    begin
        ShowMessage('Identify fail');
    end;
end;
FM_ENROLL_COMPLETE:
begin
    if RetLow=1 then
    begin
        atscctrl1.GetReferenceTemplate(revbuf[0],revsize);
        fs:=TFileStream.Create(m_appptah+'reference.dat',fmCreate);
        fs.Position:=0;
        fs.Write(revbuf,revsize);
        fs.Free;
    end;
end;
FM_VERIFY_COMPLETE:
begin
    if RetLow=1 then
    begin
        ShowMessage('Identify succeed');
    end else
    begin
        ShowMessage('Identify fail');
    end;
end;
end;
end;
```

这部分代码用于接收自定义消息，指纹处理的各种状态通过消息来实现。

4. WEB 部分,要设置两个地址,用于 XMLHTTP 协议的 Post 和 Get。服务器端为 Windows Server 2003+IIS+ASP 2.0。

服务端基本代码(部分)

```
<%@ WebHandler Language="VB" Class="upload" %>
Imports System.Web
Imports System.Xml
Imports System.Web.Services

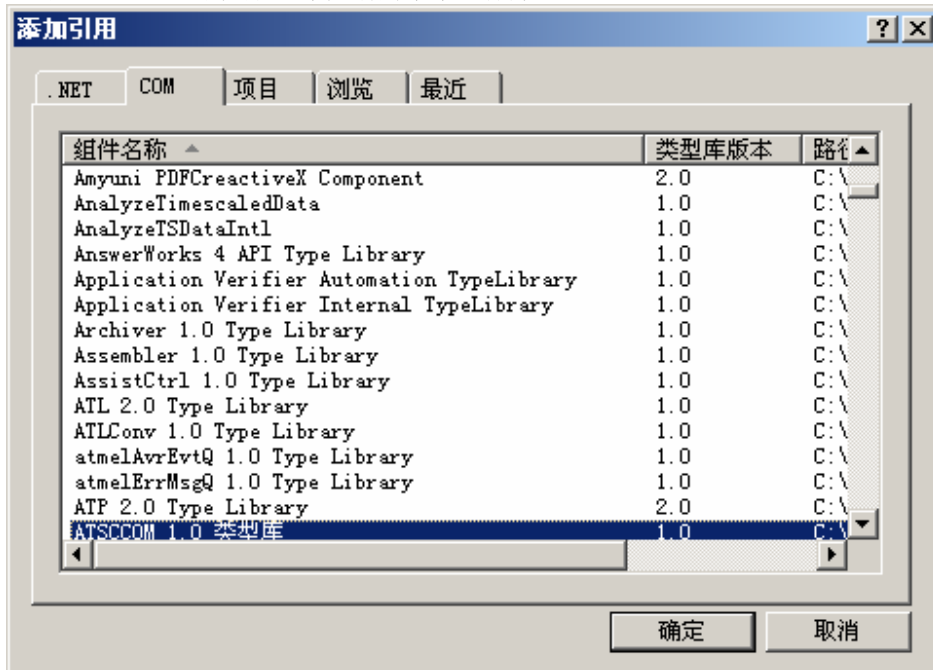
Public Class upload Implements System.Web.IHttpHandler
    Sub ProcessRequest(ByVal context As HttpContext) Implements
    IHttpHandler.ProcessRequest
        Dim strType As String
        Dim xmlDoc As New XmlDocument()
        xmlDoc.Load(context.Request.InputStream)
        xmlDoc.Save("c:\inetpub\wwwroot\fps.xml")
        '成功必须返回 succeed,失败 fail
        context.Response.Write("succeed")
        xmlDoc = Nothing
    End Sub
    ReadOnly Property IsReusable() As Boolean Implements
    IHttpHandler.IsReusable
        Get
            Return False
        End Get
    End Property
End Class
```

5. 详细代码见 DEMO 源代码

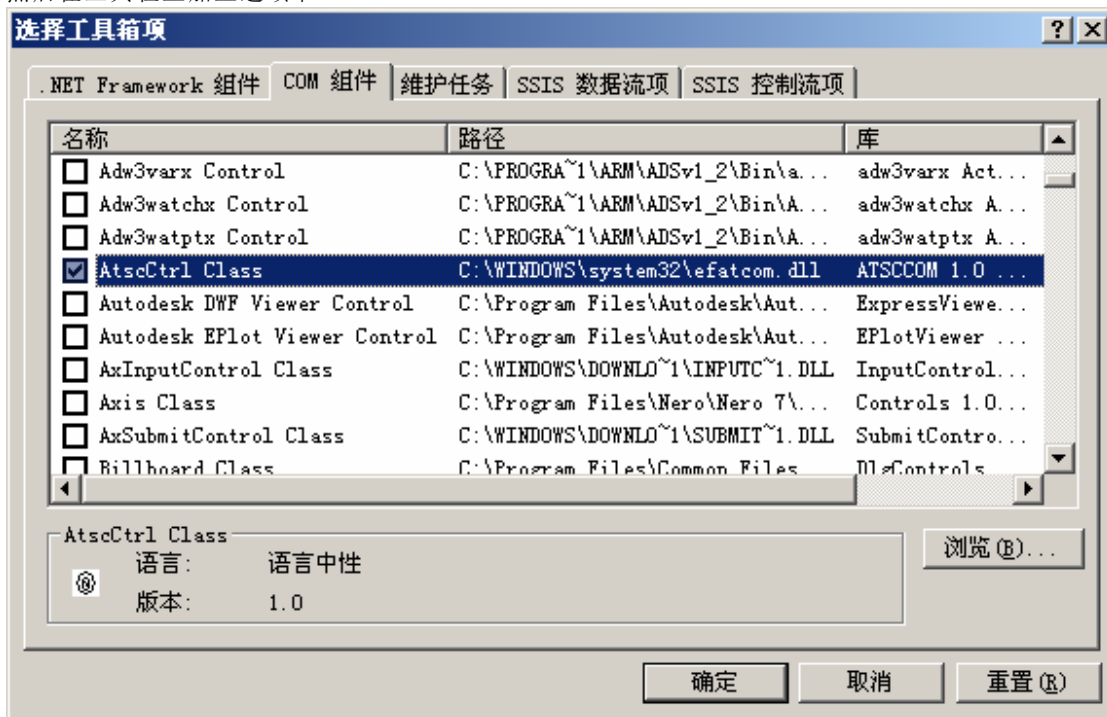
指纹识别开发包使用说明书

Microsoft Visual Basic 样例 (VS 2005)

1. 安装组件
点击 VS 2005 的项目 -> 添加引用菜单, 打开



然后在工具栏上加上选项卡



指纹识别开发包使用说明书

2. 界面如下



3. 代码

自定义消息处理部分代码

```
Protected Overrides Sub WndProc(ByRef m As Message)
    Dim fmsg As Integer = m.WParam.ToInt32
    Dim retl As Integer = LOWORD(m.LParam.ToInt32)
    Dim reth As Integer = HIWORD(m.LParam.ToInt32)
    If m.Msg = WM_FPSMSG Then
        Select Case fmsg
            Case FM_CAPTURE_NEWIMAGE
                MessageBox.Show("Capture Image succeed")
            Case FM_ENROLL_COMPLETE
                If retl = 1 Then
                    MessageBox.Show("Enroll fingerprint template succeed")

                    Dim tpsize As Integer
                    Dim tpbuf(7500) As Byte
                    If AxAtscCtrl1.GetReferenceTemplate(tpbuf(0), tpsize) = 1
Then
                        Dim fpdat(tpsize) As Byte
                        CopyMemory(fpdat, tpbuf, tpsize)

                        Dim sql As String = "SELECT * FROM FingerDB"
                        Dim conn As OleDbConnection = New OleDbConnection
conn.ConnectionString = connectionString
                        If conn.State <> ConnectionState.Open Then
                            conn.Open()
                        End If
                        Dim adapter As OleDbDataAdapter = New
OleDbDataAdapter(sql, conn)
                        Dim cmdBuilder As OleDbCommandBuilder = New
OleDbCommandBuilder(adapter)
                        Dim ds As DataSet = New DataSet("FingerDB")
                        adapter.MissingSchemaAction =
MissingSchemaAction.AddWithKey
                        adapter.Fill(ds, "FingerDB")
                        Dim row As DataRow = ds.Tables("FingerDB").NewRow()
                        row("ID") = TextBox1.Text
                        row("Template") = fpdat
                        ds.Tables("FingerDB").Rows.Add(row)
                    End If
                End If
        End Select
    End If
End Sub
```

指纹识别开发包使用说明书

```
        adapter.Update(ds, "FingerDB")
        If conn Is Nothing Then
            If conn.State = ConnectionState.Open Then
                conn.Close()
            End If
            conn.Dispose()
        End If
        MsgBox.Show("Save to database")
    End If
Else
    MsgBox.Show("Enroll Fail")
End If

        Case FM_VERIFY_COMPLETE
    If ret1 = 1 Then
        MsgBox.Show("Identify succeed")
    Else
        MsgBox.Show("Identify fail")
    End If
    End Select
Else
    MyBase.WndProc(m)
End If
End Sub
```

4. 详细代码见 DEMO 源代码