

Topaz Systems, Inc.
SigIDp™ Fingerprint Capture Imaging ActiveX Control
Version 1.0.0.7 - 5/24/2006

Topaz Systems, Inc.
650 Cochran Street, Suite 6
Simi Valley, CA, 93065
805 520-8282 (Main)
805 520-0867 fax

Tech support:

805 520-8286

805 520-8282

www.topazsystems.com

support@topazsystems.com

Copyright 2005, all rights reserved

SigIDp™ Fingerprint Capture System

INSTALLATION NOTE: The control should be properly setup and registered when running the setup program. To register the control manually, copy SigIDp.ocx into the c:\windows\sigplus directory. Then run regsvr32 c:\windows\sigplus\SigIDp.ocx. Or, copy SigIDp.ocx into the c:\winnt\sigplus directory. Then run regsvr32 c:\winnt\sigplus\SigIDp.ocx

DESCRIPTION:

SigIDp.ocx is an ActiveX control that allows the developer to capture fingerprint images (bmp, tif and jpg) at high resolution for crisp fingerprint imaging, and also provides fingerprint verification.

Fingerprint verification can be done using the built-in database, or can be extracted to file or ASCII hex string, for storage in your own file system or database. Bear in mind, one-to-many searches can only be performed using the built-in database system. Extracted fingerprint files and fingerprint strings can only be used to compare on a one-to-one basis.

TECH SUPPORT

Call (805) 520-8286 or (805) 520-8286 for Technical support

Be sure to see <http://www.topazsystems.com/Software/faq.htm> for the most frequently asked questions and answers to software and hardware related issues.

FOR A VB6 DEMO USING SIGIDP.OCX, PLEASE SEE YOUR Win\SigPlus\SigIDp\Proj DIRECTORY AFTER INSTALLATION.

CONTENTS:

General Methods (Initialization): p. 1

Imaging Methods: p. 3 – 5

Fingerprint Verification Methods, Built-in Database: p. 6 – 7

Fingerprint Verification Methods, Portable Format: p. 8

Installation Notes and Important Notices: 9 – 10

Properties:

GENERAL METHODS (HARDWARE INITIALIZATION):

.InitDevice() As Short

Initializes the fingerprint device. Must be called successfully before using any other methods.

Example:

```
Dim intRet As Integer  
intRet = SigIDp1.InitDevice
```

RETURN: integer. 0 – Success, 1 – Failure (most likely due to device being unplugged)

.CloseDevice() As Short

Closes the fingerprint device. May be called only after a successful return from InitDevice(), when you are ready to close the application.

Example:

```
Dim intRet As Integer  
intRet = SigIDp1.CloseDevice
```

RETURN: integer. 0 – Success, 1 – Failure, 2 – Device already initialized

.SetDialogDelay(long milliseconds)

Sets the length of time (in milliseconds) the capture dialog will remain after a fingerprint is successfully captured (for use with imaging methods only—CreateBmp(), CreateJpg(), CreateTif()).

Example:

```
SigIDp1.SetDialogDelay 2000 'sets delay for 2 seconds
```

.GetDialogDelay() As Long

Returns the current length of time (in milliseconds) the fingerprint dialog will display after a successful capture. (for use with imaging methods only—CreateBmp(), CreateJpg(), CreateTif()).

RETURN: long (time in milliseconds)

Example:

```
Dim myRet As Long  
myRet = SigIDp1.GetDialogDelay
```

IMAGING METHODS

Images are created at a resolution of 500x500 pixels.

.CreateBmp(String filepath) As Short

This method creates a bitmap image of a fingerprint at the location specified through its argument, filepath. The user is guided by an automatic capture dialog. This method takes a string argument, the path to the file you wish to save to.

RETURN: integer 1 – Success, 2 - User closed capture dialog window (pressed the 'X')

Example:

```
Dim intRetVal As Integer  
intRetVal = SigIDp1.CreateBmp("C:\myimage.bmp")
```

```
If intRetVal = 1 Then  
    'saved to file ok  
Elseif intRetVal = 2 Then  
    MsgBox "You Pressed 'X'", vbOKOnly + vbExclamation, "User Canceled"  
End If
```

.CreateJpg(String filepath) As Short

This method creates a jpeg image of a fingerprint at the location specified through its argument, filepath. The user is guided by an automatic capture dialog. This method takes a string argument, the path to the file you wish to save to.

RETURN: integer 1 – Success, 2 - User closed capture dialog window (pressed the 'X')

Example:

```
Dim intRetVal As Integer  
intRetVal = SigIDp1.CreateJpg("C:\myimage.jpg")
```

```
If intRetVal = 1 Then  
    'saved to file ok  
Elseif intRetVal = 2 Then  
    MsgBox "You Pressed 'X'", vbOKOnly + vbExclamation, "User Canceled"  
End If
```

.CreateTif(String filepath) As Short

This method creates a tif image of a fingerprint at the location specified through its argument, filepath. The user is guided by an automatic capture dialog. This method takes a string argument, the path to the file you wish to save to.

RETURN: integer 1 – Success, 2 - User closed capture dialog window (pressed the 'X')

Example:

```
Dim intRetVal As Integer  
intRetVal = SigIDp1.CreateTif("C:\myimage.tif")
```

```
If intRetVal = 1 Then  
    'saved to file ok  
Elseif intRetVal = 2 Then  
    MsgBox "You Pressed 'X'", vbOKOnly + vbExclamation, "User Canceled"  
End If
```

.BmpBufferBytes() As Variable

This method creates a bmp buffer, an array of bytes, of the fingerprint image. The user is guided by an automatic capture dialog.

RETURN: Variable - byte array holding the fingerprint image bmp.

Example:

```
Dim myArr() As Byte 'create a byte array to hold the bmp  
myArr = SigIDp1.BmpBufferBytes
```

.SetLowResInclusion(Bool state)

This method allows the user to create a copy of the display fingerprint image to file (what shows up on screen when the finger is placed on the device) along with the high-res image, if desired, when CreateBmp(), CreateTif() or CreateJpg() are called. True=include the lower-res image, False=do not include the low-res image. This is set to False by default. User should use the SetLowResInclusionFilepath() method to choose the save path for this low-res image file.

RETURN: None.

Example:

```
SigIDp1.SetLowResInclusion True 'will now save a low-res image with the high-res image
```

.GetLowResInclusion() As Boolean

This method lets the user know the current state of the low-resolution image inclusion (see SetLowResInclusion above). True=low res image will be generated on a call to create a fingerprint image, False=low res image will not be created (default).

RETURN: Boolean, current state of SetLowResInclusion().

Example:

```
Dim blnReturn As Boolean  
blnReturn = SigIDp1.GetLowResInclusion True 'will now save a low-res image with the  
'high-res image
```

.SetLowResInclusionFilepath(String filepath)

This method allows the user to choose the filepath to which the low res image is to be written when SetLowResInclusion() is set to True (see SetLowResInclusion() above). The developer should NOT include the file extension in the filepath, as the extension type is automatically inferred in the call to CreateBmp(), CreateTif() and CreateJpg(). SetLowResInclusion must be set to True for this method to take effect.

RETURN: None.

Example:

```
SigIDp1.SetLowResInclusion True 'will now save a low-res image with the high-res image  
Dim myPath As String  
myPath = "C:\myfolder\mynewlowresimage"  
SigIDp1.SetLowResInclusionFilepath myPath  
'developer can now call CreateBmp(), CreateTif() or CreateJpg()
```

.GetLowResInclusionFilepath() As String

Returns the current filepath (no file extension included) as set in the SetLowResInclusionFilepath() method.

RETURN: String, filepath set in SetLowResInclusionFilepath().

Example:

```
Dim myFilepath As String  
myFilepath = SigIDp1.GetLowResInclusionFilepath
```

.BmpBufferBytesLowRes() As Variant

Returns the bmp image buffer holding the low-res image when SetLowResInclusion() is set to True. Prior to making this call, the user needs to call SetLowResInclusion() to True, and also call BmpBufferBytes() to create the bmp buffer arrays.

RETURN: Variant, bmp image buffer holding low-res fingerprint image

Example:

```
Dim myHiResArr() As Byte
```

```
Dim myLowResArr() As Byte
```

```
SigIDp1.SetLowResInclusion True 'will now save a low-res image with the high-res image  
myHiResArr = SigIDp1.BmpBufferBytes  
myLowResArr = SigIDp1.BmpBufferBytesLowRes
```

FINGERPRINT VERIFICATION METHODS, BUILT-IN DATABASE

.DeleteUser(String name)

Takes in a name as a string and removes that entry from the database of users that have previously been enrolled using EnrollUser().

RETURN: None.

Example:

```
Dim name As String  
SigIDp1.DeleteUser(name)
```

.EnrollUser(String name) As Integer

While the database is open, takes in a name as a string and adds it into the database of users.

RETURN: Integer, 0 – Success, 1 – User is already in the database, 2 – Database isn't open, 3 – The user cancelled the signature capture, 4 – The user used multiple fingers, else – Unhandled exception.

Example:

```
Dim name As String  
Dim retVal As Integer  
retVal = SigIDp1.EnrollUser(name)
```

.GetNumberOfUsers() As Integer

Returns the number of users in the currently open database.

RETURN: Integer, The number of users in the currently open database.

Example:

```
Dim users As Integer  
users = SigIDp1.GetNumberOfUsers
```

.Identify() As String

Takes a fingerprint from the user and checks to see if it is present in the currently open database. A string is returned based on the result.

RETURN: String, "cancel" – The user cancelled the fingerprint capture, "" – The user is not in the currently open database, else - the name of the user in the currently open database to whom the fingerprint belongs.

Example:

```
Dim strUserFound As String
```

```
strUserFound = SigIDp1.Identify
```

.InitDeviceDb(String name, String path) As Integer

Opens an existing user specified database or creates one and then initializes the fingerprint capture device. An integer is returned based on the result.

RETURN: Integer, 0 – Database successfully opened, 1 – Failure, device is probably unplugged, 2 – Failure, a database is currently open.

Example:

```
Dim name As String  
Dim path As String  
Dim retVal As Integer
```

```
retVal = SigIDp1.InitDeviceDb(name, path)
```

.ListAllUsers() As String

Returns a string representation of the users in the currently open database in the following format: ID#<username>ID#<username>ID#<username>...

RETURN: String, all of the users in the currently open database.

Example:

```
Dim userList as String
```

```
userList = SigIDp1.ListAllUsers
```

.ValidateID(String name) As Integer

Takes in the name of an entry in the database as a string, gets a fingerprint from the user, and then compares that entry's fingerprint to the new fingerprint from the user and returns an integer based on the result. InitDeviceDb() must be called prior to this method in order for this method to work.

RETURN: integer 0 – Successful match, 1 – Unsuccessful match, 3 – Database not open, 4 – The user cancelled fingerprint capture, 5 – The user not in database, else – Some unhandled exception occurred.

Example:

```
Dim strFilePath As String  
Dim strFileName As String  
SigIDp1.InitDeviceDb(strFilePath, strFileName)
```

```
Dim intRet As Integer  
Dim name As String  
intRet = SigIDp1.ValidateID(name)
```

FINGERPRINT VERIFICATION METHODS, PORTABLE FORMAT

.GetFingerprintString() As String

Takes a set of 4 fingerprints (all of the same finger) and returns the string representation.

RETURN: String, "3" – The user closed the signature capture window, "4" – The user used multiple fingers, else - Current fingerprint string.

Example:

```
Dim fString As String  
fString = SigIDp1.GetFingerprintString
```

.GetFingerprintFile(String strFile) As Integer

Takes a set of 4 fingerprints (all of the same finger) and creates a .dat file with a user specified name.

RETURN: integer 0 – Success, 3 – The user cancelled the fingerprint capture, 4 – The user used multiple fingers, else – Not initialized properly.

Example:

```
Dim strFile As String  
Dim retVal As Integer  
retVal = SigIDp1.GetFingerprintFile(strFile)
```

.ValidateFingerprintString(String strFile) As Integer

Captures a fingerprint from the user and compares it against the current fingerprint stored in the SigIDp object. An integer is returned based on the result.

RETURN: Integer, 0 – Fingerprint matches, 1 – Fingerprint does not match, 2 – There was an error loading the file, 3 – The user cancelled the fingerprint capture.

Example:

```
Dim strFile As String  
Dim retVal As Integer  
retVal = SigIDp1.ValidateFingerprintString(strFile)
```

.ValidateFingerprintFile(String strFile) As Integer

Takes in a string representation of a file path to a .dat file to compare against, and then captures the user's fingerprint. An integer is returned based on the result.

RETURN: Integer, 0 - Fingerprint matches, 1 – Fingerprint does not match, 2 – There was an error loading the file, 3 – The user cancelled the fingerprint capture.

Example:

```
Dim strFile As String  
Dim retVal as Integer  
retVal = SigIDp1.ValidateFingerprintFile(strFile)
```

Installation notes

Topaz Systems SigIDp post-installation directory structure is:

WIN

\SigPlus

[SigIDp.ocx \(Fingerprint Capture ActiveX Control\)](#)

\SigIDp

[sigidpclient.exe \(The SigIDp demo using SigIDp.ocx\)](#)

\Docs [sigidp.pdf \(this development documentation\)](#)

\Proj [sigidp_vb.zip\(VB6 demo using SigIDp ActiveX control\)](#)

Important Notices

License Agreement and Limited Warranty

IMPORTANT: Please read this document before continuing the software load procedure. By loading the software enclosed with this agreement, you are indicating acceptance of the terms of this legal agreement between you (herein call Licensee) and Topaz Systems, Inc. (herein called Topaz). If you do not agree to the terms of this agreement, do not load the enclosed software and promptly return the product.

1. Limited use License: Topaz and it's suppliers (if any) grant you the right to use the software for use with Topaz Gem-Series tablets only. The software is owned for distribution exclusively by Topaz and is protected by the United States Patent and Trademark laws and international treaties.
2. Governing Law, Jurisdiction, and Forum: This agreement is governed by the laws of the State of California, County of Ventura.
3. YOU MAY:
 - (a) Freely use, copy, and distribute the enclosed software only for use with Topaz Gem-Series tablets.
4. YOU MAY NOT:
 - (a) Use, distribute or modify the Topaz software for use with tablets not supplied by Topaz Systems, Inc.
 - (b) Reverse engineer, decompile, or dis-assemble, the software.
5. LIMITED WARRANTY:

Topaz Systems warrants to the original buyer only, that the media upon which the Software is recorded is free from defects in workmanship and material under normal use and service for a period of 30 days.
6. EXCLUSIVE REMEDY:

Topaz's entire liability and your exclusive remedy shall be, at Topaz's option, either (a) the repair or replacement of the software or (b) the refund of the price that was paid for the software package. The defective software, along with proof of payment, must be returned to Topaz within the terms of the warranty as set forth in this agreement.
7. LIMITATIONS ON DAMAGES:

In no event shall Topaz be liable for damages whatsoever, (including without limitation, damages for loss of profits, business interruption, loss of information, or other pecuniary loss) arising out of the use of or inability to use the software even it Topaz or it's suppliers, if any, have been advised have been advise of the possibility of such damages. In no event will Topaz's liability for any reason exceed the actual price paid for the license to use the specific program.
8. NO OTHER WARRANTIES:

With respect to the software, media, and written material, Topaz and it's suppliers, if any, disclaim all warranties, other than the above Topaz warranty, including but not limited to warranties merchantability or fitness for a particular use. Topaz does not warranty the software will meet or requirements or that the operations of the software will be uninterrupted or error free.
9. GOVERNING LAW AND GENERAL PROVISIONS:

If any part of this agreement is found void and unenforceable, it will not effect the validity of the balance of the agreement, which shall remain valid and enforceable according to it's terms. You agree that the software will not be shipped, transferred, or exported into any country or used in any manner prohibited by the United States Export Administration Act or any other export laws,

restrictions or regulations. This agreement shall automatically terminate upon failure of you to comply with its terms. This agreement may only be modified in writing signed by an officer of Topaz Systems, Inc. This agreement will not be governed by the United Nations convention on contracts for the International sale of goods, the application of which is expressly excluded.

Statement regarding patents, and warning against modification of Topaz products:

Topaz products are covered under US patents 5,120,908 and 5,122,623. Patent work is ongoing and pending. Use of Topaz's products in accordance with our instructions, to the best of our knowledge, does not infringe any patents. However, be aware that there are many patents out there, and modification of our products, or use of our products for other than their intended purpose, could run afoul of these patents. It is the responsibility of Topaz customers to honor relevant patents. U. S. Patent Nos. 5,544,255; 5,647,017; 5,818,955 and others are patents that users of Topaz products should consider if modifications are to be made to our products or if our products are to be used for other than their intended purpose. These patents may be obtained at <http://www.uspto.gov>. Only as examples, without first considering these patents, you should refrain from storing within the signature information (sig file or sig data):

1. A reason for signing or a statement of importance of the signature together with a set of measurements relating to the handwritten signature and either an indication of the signatory or means for comparing said measurements with a set of statistics of a genuine signature to obtain a similarity score.
2. A visual representation of a signature together with a document checksum, hash, or receipt, and the claimed identity of the signatory.
3. A set of measurements relating to biometric information together with a document checksum, hash, or receipt, and the claimed identity of the user.

The foregoing statement is not intended to be an interpretation of the patents or their application to any particular product or implementation.

It may be possible that Federal (FDA) and state digital signature regulations become violated if techniques in 1, 2, and 3 above are employed, so rest assured, Topaz software does not do any of them. Note the information below:

1. The FDA regulations state that "Signed electronic records shall contain information associated with the signing that clearly indicates all of the following ...printed name ... date and time ... The meaning ...". (see FDA guidelines section 11.50) The FDA regulations treat electronic records and electronic signatures as different entities (Section 11.70). Therefore, you should refrain from modifying the Topaz system on your own to store any of the name, date and meaning data in the signature file.
2. State regulations such as CA code of Regulations, Title 2, Division 7, Chapter 10 for example, require that the signature be bound to only the single message that is signed and not to any other message. Therefore you should refrain from modifying the Topaz system on your own to place a checksum in with the signature data and then adding other information in with the signature, and then encrypting them to a secret key. If you were to modify the Topaz system to do this, you would be binding the signature to multiple-messages which is prohibited by the state regulations.
3. State regulations require that the signature data be capable of verification by a forensic document examiner. (CA regulations paragraph 220003(b)(3)(B) for example). Some of the techniques in 1-3 above appear at odds with "lengthy process of handwriting analysis" required by state regulation (see state of CA FAQ section). With the Topaz system, you are assured of the most accurate forensic handwriting analysis results, since the Topaz .sig file is saving all of the original signature data.

In addition, there are potential security and refutability issues if too many things are crammed in with the signature. Leave the document data, reason for signing, receipt, and identity information in the document as GemTools is designed to work, and as complies with FDA and state regulations. That way, the signature is bound to all of the data, not just to part of the data. It is very easy to create all kinds of standard image files of the signature using GemTools software. Topaz signature & document security results from our patent pending methods of encrypting the .sig data directly to a document hash (not by putting the hash or receipt into the .sig file), and non-repudiation results from the proper use of (again patent pending) Topaz signature and document receipts. Please contact the factory if you have any additional questions or would like more information about your rights concerning patents as a customer or developer using Topaz products.

Important Notice:

This software or any or all additional documentation, guidelines, or examples do not constitute a warranty about the performance, security, or legal acceptability of GemTools software or the SigPlus control in any specific use or implementation. To the extent that GemTools or SigPlus are used to achieve regulatory or other specific objectives within an industry, you must consult competent experts or regulatory officials together with your own plan to achieve your desired business objectives using the Topaz tools.