eNBSP - NBioBSP

NITGEN&COMPANY Biometric Service Provider SDK

Programmer's Manual VB

SDK version 4.8x

Serial Number:

Specifications subject to change without notice.

INDEX

Chapter 1. Introduction

The eNBSP (NBioBSP) SDK provides feature rich, high-level functionality that can be integrated into any application requiring fingerprint authentication. NBioBSP technology is built on the NBioAPI™ specification, working seamlessly with the most durable, compact, and reliable optics-based fingerprint readers in the world.

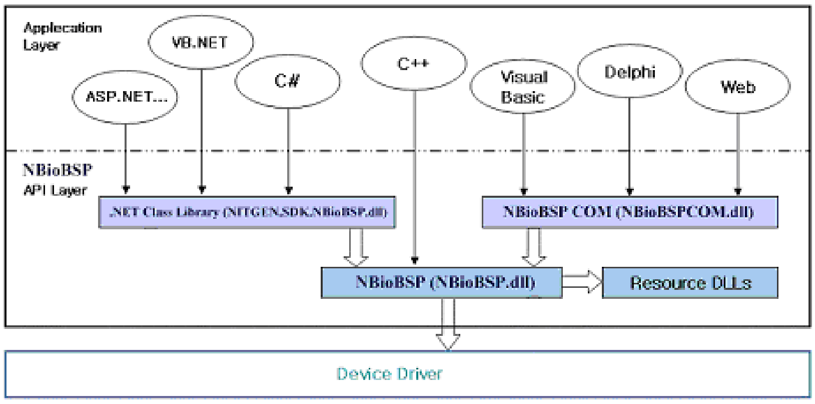
All NBioBSP SDK components contain the APIs needed for biometric authentication of multiple users and device functions. NBioBSP is equipped with self-contained User Interfaces for enrollment and verification, enabling software application developers to quickly and easily integrate fingerprint authentication into the application of their choice.

1.1 Support Modules

- NBioBSP.dll
  This is the main module of the NBioBSP that implements all of NITGEN's biometric functions including fingerprint enrollment and verification

- NBioBSPCOM.dll
  NBioBSP COM module based Microsoft COM Technology that facilitates easily integration of NBioBSP by developers using RAD tools or doing web development.

- NBioBSPCOMLib.dll
  It support only 64bit SDK



[Development model using NBioBSP SDK]

1.2 Sample programs

1.2.1 Support Samples (32bit SDK)

- Visual Basic
  BSPDemoVB – Basic function demo application for VB.
  DataExportDemoVB – Data Export / Import function demo application for VB.
  UITestVB – User Interface demo application for VB.
  IndexSerchVB – IndexSearch demo application for VB.
  BSPRollDemoVB – Roll fingerprint function demo application for VB.

- ASP
  Basic function demo application for ASP.

- C#
  BSPDemoCSharp – Basic function demo application for C#.
  UITestCSharp – User Interface demo application for C#.

- Delphi
  BSPDemoDP - Basic function demo application for Delphi.
  IndexSearchDP - IndexSearch demo application for Delphi.
  UITestDP - User Interface demo application for Delphi.

1.2.2 Support Samples (64bit SDK)

- C#
  BSPDemoCSharp – – Basic function demo application for C#.
  UITestCSharp – User Interface demo application for C#.

Chapter 2. Visual Basic Programming

This chapter describes Visual Basic programming using the NBioBSP COM module. The NBioBSP COM module is designed to help easily integration of NBioBSP by developers using RAD tools or doing web development.

The NBioBSP COM module does not support all functions of NBioBSP module. When developing web programs, fingerprint data must be handled in text format. They can be handled in binary or text format on Visual Basic or Delphi programming.

2.1 Module initialization and closure

2.1.1 Initializing the module

There are two ways to initialize the **NBioBSP COM** module: declare a **new object** or use the **CreateObject()** function. Using either method will have the same result.

- Method 1

```
Dim objNBioBSP As NBioBSPCOMLib.NBioBSP            'Declaration NBioBSPCOM Object
...
Set objNBioBSP = New NBioBSPCOMLib.NBioBSP
```

- Method 2

```
Dim objNBioBSP As NBioBSPCOMLib.NBioBSP            ' Declaration NBioBSPCOM Object
...
Set objNBioBSP = CreateObject("NBioBSPCOM.NBioBSP")
```

2.1.2 Declaring the child object

There are 7 child objects used in the NBioBSP COM module as follows.

- Declaring the evice object
  This object is used to open, close and configure device settings.

  ```
  Dim objDevice As IDevice          ‘ Declaration Device Object
  ...
  Set objDevice = objNBioBSP.Device
  ```

- Declaring the extraction object
  This object is used to capture and register fingerprint data.

  ```
  Dim objExtraction As IExtraction  ‘ Declaration Extraction Object
  ...
  Set objExtraction = objNBioBSP.Extraction
  ```

- Declaring the matching object
  This object is used to perform matching.

  ```
  Dim objMatching As IMatching      ‘ Declaration Matching Object
  ...
  Set objMatching = objNBioBSP.Matching
  ```

- Declaring the FPData object
  This object is used to convert and recreate fingerprint data.

  ```
  Dim objFPData As IFPData          ‘ Declaration FPData Object
  ...
  Set objFPData = objNBioBSP.FPData
  ```

- Declaring the FPImage object
  This object is used to extract and save fingerprint images.

  ```
  Dim objFPImage As IFPImage        ‘ Declaration FPImage Object
  ...
  Set objFPImage = objNBioBSP.FPImage
  ```

- Declaring the IndexSearch object
  This object is used to perform the IndexSearch Engine functions.

  ```
  Dim objIndexSearch As IIndexSearch          ‘ Declaration IndexSearch Object
  ...
  Set objIndexSearch = objNBioBSP.IndexSearch
  ```

- Declaring the NSearch object
  This object is used to perform the NSearch Engine functions.

  ```
  Dim objNSearch As INSearch        ‘ Declaration NSearch Object
  ...
  Set objNSearch = objNBioBSP.NSearch
  ```

2.1.3 Closing the module after use

Declare the object free when closing the application. Note that this is done automatically in Visual Basic when the application is closed.

```
Set objNBioBSP = nothing          ‘ Free NBioBSPCOM object
```

2.2 Device related programming

The device must be opened before it can be used. Use the **Enumerate** method to determine which device is linked to the system.

2.2.1 Listing devices

Before opening the device, use the **Enumerate** method to determine the number and type of devices linked to the PC. Once this is activated, the number of devices linked to the PC will appear in the **EnumCount** property and the ID for each device will appear in the **EnumDeviceID** property. **EnumDeviceID** is a LONG value array. **EnumDeviceID** is composed of the device names and their instance numbers.

NBioBSP sets default settings for the **FDU01** fingerprint recognition device at **0x02** and the **FDU11** fingerprint recognition device at **0x04.**

DeviceID = Instance Number + Device Name

If there is only one device for each type in the system, the instance number will be '0.' In this way, the device name has the same value as the device ID.

The following is an example of how to use the **Enumerate** method.

```
Set objDevice = objNBioBSP.Device              ' Set Device object
...
Call objDevice.Enumerate                       ' Enumerate devices
...
comboDevice.AddItem "Auto_Detect"
For DeviceNumber = 0 To objDevice.EnumCount
    comboDevice.AddItem objDevice.EnumDeviceName(DeviceNumber)
Next i
...
```

The **EnumDeviceID(DeviceNumber)** can be shown as inputting the number of the device for the **DeviceNumber** of the EnumDeviceID (DeviceNumber) property. For example, EnumDeviceID(0) will show the ID of the first device.

2.2.2 Initializing the device

The **Open** method is used to initialize the device for the **NBioBSP COM**. Device initialization must be done using the **Open** method before device related functions such as enrolling, verifying, and capturing will work properly.

In the event that you are unsure which devices have been installed, use the **Enumerate** method to determine what devices have previously been installed.

```
DeviceID = NBioBSP_DEVICE_ID_FDU01_0            ‘ First instance of FDU01 device
Call objDevice.Open(DeviceID)

If objDevice.ErrorCode = NBioBSPERROR_NONE Then
    ‘ Open device success ...
Else
    ‘ Open device failed ...
End If
```

The device can be set automatically using **NBioBSP_DEVICE_ID_AUTO_DETECT**.

```
objDevice.Open(NBioBSP_DEVICE_ID_AUTO_DETECT)
```

NBioBSP_DEVICE_ID_AUTO_DETECT use the latest opened device.

2.2.3 Closing the device

The **Close** method should be used to close the device. The same **DeviceID** used to call the **Open** method must be used again to call the **Close** method.

```
DeviceID = NBioBSP_DEVICE_ID_FDU01_0
Call objDevice.Open(DeviceID)
...
Call objDevice.Close(DeviceID)

If objDevice.ErrorCode = NBioBSPERROR_NONE Then
    ‘ Close device success ...
Else
    ‘ Close device failed ...
End If
```

The current device must be closed before opening another device.

2.3 Fingerprint enrollment

The **Enroll** method is used to enroll fingerprints. This method must be used after declaring the extraction object. All fingerprint data is used as the type of binary or encoded text found in the **NBioBSP** module. Fingerprint data will be entered into the **FIR** or **TextEncoedFIR** property upon successful enrollment. The **TextEncoedFIR** has String type value.

```
Dim szTextEncodeFIR As String
Dim szPayload As String
...
Set objExtraction = objNBioBSP.Extraction
Call objExtraction.Enroll(szPayload, null)

If objExtraction.ErrorCode = NBioBSPERROR_NONE Then
    ‘ Enroll success ...
    szTextEncodeFIR = objExtraction.TextEncodeFIR
    ‘ Write FIR data to file or DB
Else
    ‘ Enroll failed ...
End If
```

Fingerprint data will be stored as saving **TextEncodedFIR** to a file or DB.
Fingerprint data also can be retrieved in binary type as follows.

```
Dim biFIR() As Byte
...
ReDim biFIR(objExtraction.FIRLength) As Byte
biFIR = Space(objExtraction.FIRLength)
biFIR = objExtraction.FIR
```

2.4 Fingerprint verification

The **Verify** method performs fingerprint verification using the existing fingerprint data as a comparison with newly input fingerprints. This method must be used after declaring the Matching object. The result is saved as a value in the **MatchingResult** property: 1 for success, 0 for failed verification.

```
Dim storedFIRTextData As String
Dim szPayload As String
...
‘ Read stored FIR Data from File or DB.
...
Set objMatching = objNBioBSP.Matching
Call objMatching.Verify(storedFIRTextData)            ‘ TextEncodedFIR

If objMatching.MatchingResult = NBioAPI_TRUE then
    ‘ Verify success
    If objMatching.ExistPayload = NBioAPI_TRUE Then
        ‘ Exist
        szPayload = objMatching.TextEncodePayload
    Else
        ...
    End If
Else
    ‘ Verify failed
End if
```

2.5 Client / Server environment programming

Unlike standalone environments, the fingerprint enrollment and matching occur in separate places within the Client/Server environment. Fingerprints are generally enrolled in the client and later matched in the Server.

The **Enroll** method registers fingerprints while the **Capture** method verifies fingerprints.
The **VerifyMatch** method matches fingerprints in the Server through the use of previously registered fingerprints from the client.

2.5.1 Fingerprint enrollment

Use the **Enroll** method for fingerprint enrollment in the client.

```
Dim szTextEncodeFIR As String
Dim szPayload As String
...
Set objExtraction = objNBioBSP.Extraction
Call objExtraction.Enroll(szPayload, null)

If objExtraction.ErrorCode = NBioBSPERROR_NONE Then
    ‘ Enroll success ...
    szTextEncodeFIR = objExtraction.TextEncodeFIR
    ‘ Write FIR data to file or DB
Else
    ‘ Enroll failed ...
End If
```

2.5.2 Fingerprint verification

Use the **Capture** method for registering a fingerprint in the client. While the **Enroll** method allows several fingerprints to be enrolled and transferred as one FIR, the **Capture** method registers only one fingerprint. The Capture method must be used after declaring the Extraction object. Input the purpose of the Capture in the parameter; the values for the purpose, define in header files, are variable, but this method allows NBioAPI_FIR_PURPOSE_VERIFY only.

```
Dim szTextEncodeFIR As String
...
Set objExtraction = objNBioBSP.Extraction
Call objExtraction.Capture(NBioAPI_FIR_PURPOSE_VERIFY)

If objExtraction.ErrorCode = NBioBSPERROR_NONE Then
    ‘ Capture success ...
    szTextEncodeFIR = objExtraction.TextEncodeFIR
    ‘ Write FIR data to file or DB
Else
   ‘ Capture failed ...
End If
```

The **VerifyMatch** method takes two FIRs, the transmitted FIR data on a network and the previously enrolled FIR data, and matches between these two FIRs. See the **MatchingResult** property to check the verification result; 1 for success, 0 for failed verification. After successful verification, the method returns the payload.

```
Dim storedFIRTextData As String
Dim processedFIRTextData As String
Dim szPayload As String
...
‘ Get processed FIR Data from Client and Read stored FIR Data from File or DB.
...
Set objMatching = objNBioBSP.Matching
Call objMatching.VerifyMatch(processedFIRTextData, storedFIRTextData)

If objMatching.MatchingResult = NBioAPI_TRUE then
   ‘ Matching success
   If objMatching.ExistPayload = NBioAPI_TRUE Then
      ‘ Exist
      szPayload = objMatching.TextEncodePayload
   Else
      ...
   End If
Else
   ‘ Matching failed
End if
```

2.6 Using Payload

Including other data within the fingerprint data is called a **Payload**. Only encoded text type data can be used in the **NBioBSP** module as a **payload**.

2.6.1 Inserting payload into FIR

At the time of fingerprint enrollment, use the **Enroll** method to include **payload** with the fingerprint data. The **CreateTemplate** method can be used to insert **payload** into an existing FIR. The **Enroll** method will use the fingerprint data and **payload** to provide a parameter for later comparison.

```
Dim szTextEncodeFIR As String
Dim szPayload As String
...
szPayload = “Your Payload Data”
...
Set objExtraction = objNBioBSP.Extraction
Call objExtraction.Enroll(szPayload, null)

If objExtraction.ErrorCode = NBioBSPERROR_NONE Then
   ‘ Enroll success ...
   szTextEncodeFIR = objExtraction.TextEncodeFIR
   ‘ Write FIR data to file or DB
Else
   ‘ Enroll failed ...
End If
```

Use the **CreateTemplate** method to insert a **payload** into existing fingerprint data. The **CreateTemplate** method can also add new fingerprint data onto existing fingerprint data. Just as in the **Enroll** method, the new fingerprint data will be put into the **TextEncodedFIR** property. This method must be called after declaring the FPData object.

```
Dim storedFIRTextData As String
Dim newFIRTextData As String
Dim szPayload As String
...
szPayload = “Your Payload Data”
...
Set objFPData = objNBioBSP.FPData
Call objFPData.CreateTemplate(storedFIRTextData, null, szPayload)

If objFPData.ErrorCode = NBioBSPERROR_NONE Then
   ‘ CreateTemplate success ...
   newFIRTextData = objFPData.TextEncodeFIR
   ‘ Write FIR data to file or DB
Else
   ‘ CreateTemplate failed ...
End If
```

2.6.2 Extraction payload from FIR

Payload in fingerprint identification records (registered data) will only be extracted if matched using the **Verify** method or if the **VerifyMatch** method returns true.

Check the **IsPayload** property after matching to verify whether a **payload** exists. If **ExistPayload** is true, the **payload** will be shown in the **TextEncodedPayload** property.

```
Dim storedFIRTextData As String
Dim szPayload As String
...
' Read FIRText Data from File or DB.
...
Set objMatching = objNBioBSP.Matching
objMatching.Verify(storedFIRTextData)

If objMatching.MatchingResult = NBioAPI_TRUE Then
   ' Verify success
   If objNBioBSP.ExistPayload = NBioAPI_TRUE Then
      ' Exist payload
      szPayload = objMatching.TextEncodePayload
   End If
Else
   ' Verify failed
End if
```

Extracting payloads using the **VerifyMatch** method is the same as using the **Verify** Method. When calling VerifyMatch, as a first parameter, use data using compared and as a second parameter, use stored data.(Enrolled template).

The payload data only can be extracted from FIR data in Second parameter (enrolledFIRTextData). Although FIR data in first parameter includes payload, it is not retrieved.

```
...
Set objMatching = objNBioBSP.Matching
Call objMatching.VerifyMatch(capturedFIRTextData, enrolledFIRTextData)

if objMatching.MatchingResult = NBioAPI_TRUE then
   ' Verify success
   if objMatching.ExistPayload = NBioAPI_TRUE then
      ' Get payload
      szPayload = objMatching.TextEncodePayload
    End if
End if
...
```

2.7 Changing the NBioAPI User Interface

The **NBioBSP COM** module offers resource files for customization of the basic UI in English. Use the **SetSkinResource** method to load UI resources in languages other than English.

```
Dim szSkinPath
...
CommonDialog.ShowOpen

If CommonDialog.CancelError = False then
        szSkinPath = CommonDialog.FileName
        objNBioBSP.SetSkinResource(szSkinPath)
End if
```

Resource files must have an absolute path. Extra documents are offered for making customized UI's.

Appendix A. COM Reference

A.1 NBioBSP Object

A.1.1 Property

long ErrorCode

Records the error code used in the last method or property. The value of 0 indicates success, and all other values represent an error condition.

BSTR ErrorDescription

Records the error description of text format, corresponding the error code.

VARIANT Device

An object containing a set of commands for controlling the device; enumerating devices, initializing/closing the device, and configuring the device settings.

VARIANT Extraction

Provides the functions that capture and process a fingerprint data.
It can also be used to change the UI option and save fingerprint image data.

VARIANT Matching

Provides the functions that verify a FIR data and return the result of the match.
It can also be used to change the UI option, and save fingerprint data.

VARIANT FPData

This object can be used to create a FIR data from two fingerprint data, and to insert a payload data into the FIR.

VARIANT FPImagae

This object can be used to achieve a fingerprint image data when capturing.

VARIANT NSearch

This object is to store a large number of fingerprint data into the memory DB, and to search a specific data in the memory.

VARIANT IndexSearch

This object is to store a large number of fingerprint data into the memory DB, and to search a specific data in the memory.

BOOL CheckValidityModule

This property checks the NBioBSP module validation. A value of TRUE means that the module is verified.

BSTR MajorVersion

Indicates the NBioBSP major version number.

BSTR MinorVersion

Indicates the NBioBSP minor version number. (2 digits)

BSTR BuildNumber

Indicates the NBioBSP build number.

A.1.2 Method

SetSkinResource (BSTR bszSkinPath)

Description

This method is used to apply a new skin resource to the NBioBSP module. The skin resource can be made for OEM users.

Parameter

bszSkinPath : A fullpath of the skin resource file.

Relation Property

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

A.2 Device Object

This object contains a set of commands for controlling the device; enumerating devices, initializing/closing the device, and configuring the device settings.

A.2.1 Property

long ErrorCode

Records the error code used in the last method or property. The value of 0 indicates success, and all other values represent an error condition.

BSTR ErrorDescription

Records the error description of text format, corresponding the error code.

long EnumCount

This property is to retrieve the number of devices attached to the system.
This value is set by the Enumerate method.

long EnumDeviceID(long nIndex)

Retrieves all device IDs attached to the system.
These values are set by the Enumerate method.

long EnumDeviceNameID(long nIndex)

Retrieves all device name IDs attached to the system.
These values are set by the Enumerate method.

long EnumDeviceInstance(long nIndex)

Retrieves instance of all devices attached to the system.
These values are set by the Enumerate method.

BSTR EnumDeviceName(long nIndex)

Retrieves name of all devices attached to the system.
These values are set by the Enumerate method.

BSTR EnumDeviceDescription(long nIndex)

Retrieves descriptions of all devices attached to the system.
These values are set by the Enumerate method.

BSTR EnumDeviceDll(long nIndex)

Retrieves device driver’s dll filename of all devices attached to the system.
These values are set by the Enumerate method.

BSTR EnumDeviceSys(long nIndex)

Retrieves device driver’s sys filename of all devices attached to the system.
These values are set by the Enumerate method.

long EnumDeviceAutoOn(long nIndex)

Retrieves auto on supporting of all devices attached to the system.
These values are set by the Enumerate method.

long EnumDeviceBrightness(long nIndex)

Retrieves brightness value of all devices attached to the system.
These values are set by the Enumerate method.

long EnumDeviceContrast(long nIndex)
Retrieves contrast value of all devices attached to the system.
These values are set by the Enumerate method.

long EnumDeviceGain(long nIndex)
Retrieves gain value of all devices attached to the system.
These values are set by the Enumerate method.

long OpenedDeviceID
Retrieves the last opened device ID using the Open method.

long GetDeviceName(long nDeviceID)
Retrieves the device name if a device ID is entered for the parameter.

long GetDeviceNumber(long nDeviceID)
Retrieves the device number if a device ID is entered for the parameter.
These values can be used for the USB devices only. It ranks in order of connected.

long MakeDeviceID(long nDeviceName, long nDeviceNumber)
Generates a device ID through the device name and device number.
The device ID can be used to open and close the device.

long ImageWidth(long nDeviceID)
Retrieves the image width of the fingerprint image that is captured from the device.

long ImageHeight(long nDeviceID)
Retrieves the image height of the fingerprint image that is captured from the device.

long Brightness(long nDeviceID)
Configures or retrieves the brightness value of the fingerprint image that is captured from the device.

long Contrast(long nDeviceID)
Configures or retrieves the contrast value of the fingerprint image that is captured from the device.

long Gain(long nDeviceID)
Configures or retrieves the gain value of the fingerprint image that is captured from the device.

long WindowStyle
This property is to select a window style of the NBioBSP dialogs.
It can be set to either of POPUP, INVISIBLE, or CONTINUOUS. A value of CONTINUOUS is not used on the NBioBSP COM module.

BOOL WindowOption(long nOption)
This property is to select a window style of the NBioBSP dialogs.
It can be set to either of NO_FPIMG, TOPMOST, NO_WELCOME, or NO_TOPMOST.

long ParentWnd
This property is not used.

long FingerWnd
This property can be used to set a Windows fingerprint image control, when using the NO_FPIMG window style.
Specify the handle of the control to display the image.

BSTR CaptionMsg

This property is used to set a text message to be displayed, when CANCEL button is selected on the enrollment dialog.

BSTR CancelMsg

This property is used to set a text caption to be displayed, when CANCEL button is selected on the enrollment dialog.

BSTR FPForeColor

This property is used to set a color of fingerprint image to be displayed, when using the INVISIBLE window style or displaying images on the custom control.

BSTR FPBackColor

This property is used to set a color of fingerprint background to be displayed, when using the INVISIBLE window style or displaying images on the custom control.

BOOL DisableFingerForEnroll(long nFingerID)

This property can be used to set specific fingers to be enabled or disabled, when using the Enroll method. A value of TRUE for a finger ID means disabling the finger.

BOOL CheckFinger

This property can be used to check if a finger is placed on the sensor.
Only valid for USB devices and device driver version 4.1.0.1 or higher. It returns TRUE when a finger is on.

A.2.2 Method

Open(long nDeviceID)

Description

This method is to initialize the device.

Parameters

nDeviceID : The device ID to be opened. If there are more than one device attached to the system, the device ID can be generated by using the MakeDeviceID method.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

Close(long nDeviceID)

Description

This method is to close the device opened by the Open method.

Parameters

nDeviceID : The device ID to be closed. The device IDs currently opened can be read by using the OpenedDeviceID property.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

Enumerate()

Description

This method is used to enumerate devices attached to the system.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

EnumCount : Indicates the number of devices attached to the system.

EumDeviceID(long nIndex) : Indicates device IDs retrieved.

Adjust()

Description

This method is used to configure the brightness of the device.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

A.3 Extraction Object

This object provides the functions that capture and process a fingerprint data. It can also be used to change the UI option and save fingerprint image data.

A.3.1 Property

long ErrorCode

Records the error code used in the last method or property. The value of 0 indicates success, and all other values represent an error condition.

BSTR ErrorDescription

Records the error description of text format, corresponding the error code.

BSTR TextEncodeFIR

Retrieves the text encoded FIR data.

long FIRLength

Retrieves the size, in bytes, of the FIR data.

VARIANT FIR

Retrieves the FIR data in binary type. The buffer in size of the FIRLength must be allocated before use of the methods using this property.

long FIRFormat

Retrieves the Format value, of the FIR data.

long MaxFingerForEnroll

This property is used to set the maximum number of fingers to be enrolled, when using the Enroll method.

long SamplesPerFinger

This property is used to set the number of samples (of each finger) to be enrolled, when using the Enroll method.
This value is fixed to 2.

long DefaultTimeout

This property is used to set the timeout of a fingerprint capture, when using the Capture, or the Verify method.

long EnrollImageQuality

This property is used to set the image quality criterion for a successful capture, when using the Enroll method.
This value must be set between 30 and 100.

long VerifyImageQuality

This property is used to set the image quality criterion for a successful capture, when using the Verify method.
This value must be set between 0 and 100.

long IdentifyImageQuality

This property is not used.

long SecurityLevel

Indicates the security level set for fingerprint recognition. Values range from 1 (lowest) to 9 (highest). The default is 5 (normal).

long WindowStyle

This property is to select a window style of the NBioBSP dialogs.
It can be set to either of POPUP, INVISIBLE, or CONTINUOUS. A value of CONTINUOUS is not used on the NBioBSP COM module.

BOOL WindowOption(long nOption)

This property is to select a window style of the NBioBSP dialogs.
It can be set to either of NO_FPIMG, TOPMOST, NO_WELCOME, or NO_TOPMOST.

long ParentWnd

This property is not used.

long FingerWnd

This property can be used to set a Windows fingerprint image control, when using the NO_FPIMG window style. Specify the handle of the control to display the image.

BSTR CaptionMsg

This property is used to set a text message to be displayed, when CANCEL button is selected on the enrollment dialog.

BSTR CancelMsg

This property is used to set a text caption to be displayed, when CANCEL button is selected on the enrollment dialog.

BSTR FPForeColor

This property is used to set a color of fingerprint image to be displayed, when using the INVISIBLE window style or displaying images on the custom control.

BSTR FPBackColor

This property is used to set a color of fingerprint background to be displayed, when using the INVISIBLE window style or displaying images on the custom control.

BOOL DisableFingerForEnroll(long nFingerID)

This property can be used to set specific fingers to be enabled or disabled, when using the Enroll method. A value of TRUE for a finger ID means disabling the finger.

A.3.1 Method

Capture (/*[in, optional]*/ long nPurpose)

Description

This method captures samples for the purpose specified.

Parameters

nPurpose : A value indicating the purpose of the fingerprint data capture. This value is optional, only 1 is used at this version.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

TextEncodeFIR : Text encoded FIR data.

FIRLength : Size of the FIR data.

FIR : FIR data newly captured.

RollCapture (/*[in, optional]*/ long nPurpose)

Description

This method Roll captures samples for the purpose specified.

Parameters

nPurpose : A value indicating the purpose of the fingerprint data capture. This value is optional, only 1 is used at this version.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

TextEncodeFIR : Text encoded FIR data.

FIRLength : Size of the FIR data.

FIR : FIR data newly captured.

Enroll (VARIANT payload, /*[in, optional]*/ VARIANT storedFIR)

Description

This method captures fingerprint data from the attached device for the purpose of enrollment.

Parameters

payload : A user defined data that will be wrapped inside the newly created template.

This data can be read after a successful match from the Verify or the VerifyMatch method.

Payload can be used in binary or string array.

storedFIR : Optionally, the FIR data to be adapted.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

TextEncodeFIR : Text encoded FIR data.

FIRLength : Size of the FIR data.

FIR : FIR data newly enrolled.

A.4 Matching Object

This object provides the functions that verify a FIR data and return the result of the match. It can also be used to change the UI option, and save fingerprint data.

A.4.1 Property

long	ErrorCode

Records the error code used in the last method or property. The value of 0 indicates success, and all other values represent an error condition.

BSTR	ErrorDescription

Records the error description of text format, corresponding the error code.

BOOL	MatchingResult

Indicates the result of a match.

BOOL	ExistPayload

Indicates if the payload data is included in the FIR.

BSTR	TextEncodePayload

Retrieves the text encoded payload data.

long	PayloadLength

Indicates the size, in bytes, of the payload.

VARIANT	Payload

Retrieves the payload data of binary type.

The buffer in the size of the PayloadLength must be allocated before use of the methods for the payload.

long	HitNum

Indicates the result of a CompareTwo Method

Long	MatchScore

Indicates the result of a CompareTwo Method

long	MaxFingerForEnroll

This property is used to set the maximum number of fingers to be enrolled, when using the Enroll method.

long	SamplesPerFinger

This property is used to set the number of samples (of each finger) to be enrolled, when using the Enroll method.

This value is fixed to 2.

long	DefaultTimeout

This property is used to set the timeout of a fingerprint capture, when using the Capture, or the Verify method.

long	EnrollImageQuality

This property is used to set the image quality criterion for a successful capture, when using the Enroll method.

This value must be set between 30 and 100.

long	VerifyImageQuality

This property is used to set the image quality criterion for a successful capture, when using the Verify method.

This value must be set between 0 and 100.

long IdentifyImageQuality

This property is not used.

long SecurityLevel

Indicates the security level set for fingerprint recognition. Values range from 1 (lowest) to 9 (highest). The default is 5 (normal).

long WindowStyle

This property is to select a window style of the NBioBSP dialogs.

It can be set to either of POPUP, INVISIBLE, or CONTINUOUS. A value of CONTINUOUS is not used on the NBioBSP COM module.

BOOL WindowOption(long nOption)

This property is to select a window style of the NBioBSP dialogs.

It can be set to either of NO_FPIMG, TOPMOST, NO_WELCOME, or NO_TOPMOST.

long ParentWnd

This property is not used.

long FingerWnd

This property can be used to set a Windows fingerprint image control, when using the NO_FPIMG window style.

Specify the handle of the control to display the image.

BSTR CaptionMsg

This property is used to set a text message to be displayed, when CANCEL button is selected on the enrollment dialog.

BSTR CancelMsg

This property is used to set a text caption to be displayed, when CANCEL button is selected on the enrollment dialog.

BSTR FPForeColor

This property is used to set a color of fingerprint image to be displayed, when using the INVISIBLE window style or displaying images on the custom control.

BSTR FPBackColor

This property is used to set a color of fingerprint background to be displayed, when using the INVISIBLE window style or displaying images on the custom control.

BOOL DisableFingerForEnroll(long nFingerID)

This property can be used to set specific fingers to be enabled or disabled, when using the Enroll method.

A value of TRUE for a finger ID means disabling the finger.

A.4.2 Method

Verify(VARIANT storedFIR)

Description

This method captures fingerprint data from the attached device, and compares it against the storedFIR.

Parameters

storedFIR : The FIR to be verified against.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

MatchingResult : Indicates the result of the match.

ExistPayload : Indicates the flag of the payload existence.

TextEncodePayload : Text encoded payload data.

PayloadLength : Size of the payload data.

Payload : Payload data retrieved.

VerifyMatch (VARIANT processedFIR , VARIANT storedFIR)

Description

This method performs a verification (1-to-1) match between FIRs; the processedFIR and the storedFIR.

Parameters

processedFIR : The FIR to be verified.

storedFIR : The FIR to be verified against.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

MatchingResult : Indicates the result of the match.

ExistPayload : Indicates the flag of the payload existence.

TextEncodePayload : Text encoded payload data.

PayloadLength : Size of the payload data.

Payload : Payload data retrieved.

CompareTwo(VARIANT processedFIR, VARIANT storedFIR)

Description

This method performs a verification (1-to-1) match between FIRs; the processedFIR and the storedFIR.

Parameters

processedFIR : The FIR to be verified.

storedFIR : The FIR to be verified against.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

MatchingResult : Indicates the result of the match.

HitNum : Hit number between FIRs.

MatchScore : Match score between FIRs.

A.5 FPData Object

This object can be used to create a FIR data from two fingerprint data, and to insert a payload data into the FIR.

A.5.1 Property

long ErrorCode

Records the error code used in the last method or property. The value of 0 indicates success, and all other values represent an error condition.

BSTR ErrorDescription

Records the error description of text format, corresponding the error code.

long TotalFingerCount

Indicates the number of fingers converted to a different type of minutiae from the FIR data.

long FingerID(long nIndex)

Indicates the finger IDs of fingerprint data converted from the FIR data.
Indexes range from 0 to the TotalFingerCount - 1.

long SampleNumber

Indicates the number of samples for a finger.

long FPDataSize(long nFingerID)

Indicates the size, in bytes, of the processed fingerprint data.
The nFingerID must be entered as the parameter to retrieve the size values.

long FPSampleDataSize(long nFingerID, long nSampleNum)

Indicates the size, in bytes, of the processed fingerprint data.

long FPData(long nFingerID, long nSampleNum)

Retrieves the fingerprint data.
The nFingerID and nSampleNum must be entered as the parameter to retrieve the binary fingerprint data.

BSTR TextEncodeFIR

Retrieves the text encoded FIR data.

long FIRLength

Retrieves the size, in bytes, of the FIR data.

VARIANT FIR

Retrieves the FIR data in binary type.
The buffer in size of the FIRLength must be allocated before use of the methods using this property.

long QuailtyInfo(long nFingerID, long nSampleNum)

Retrieves the quality information of fingerprint data.
The nFingerID and nSampleNum must be entered as the parameter to retrieve the fingerprint quality value.

A.5.2 Method

Export(VARIANT storedFIR, long nDesFPDataType)

Description

This method is to convert the FIR data to different data format.

Parameters

storedFIR : The FIR data to be converted.

nDesFPDataType : A value indicating the type of exportation.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

FingerID : Identifies the finger IDs retrieved.

SampleNumber : Identifies the sample number of a finger.

FPDataSize : Size of the fingerprint data.

FPData : Fingerprint data exported.

Import(BOOL bInitialize, long nFingerID, long nPurpose, long nSrcFPDataType,
    long nFPDataSize, VARIANT FPData1, /*[in, optional]*/ VARIANT FPData2)

Description

This method is to convert a different data format of minutiae to the FIR.

Parameters

bInitialize : A value indicating a flag for the FIR creation style.

A value of TRUE means to create a new FIR data, while a value of FALSE means to append to the existing FIR data.

nFingerID : A value indicating the finger ID.

nPurpose : A value indicating the purpose of import the data.

nSrcFPDataType : The type of the minutiae data.

nFPDataSize : Size, in bytes, of the minutiae data to be converted.

FPData1 : The first minutiae data to be converted.

FPData2 : Optionally. The second minutiae data to be converted. Only if the SampleNumber is 2, this value will be used.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

TextEncodeFIR : Text encoded FIR data.

FIRLength : Size of the FIR data.

FIR : FIR data imported.

CreateTemplate(VARIANT capturedFIR, VARIANT storedFIR, VARIANT payload)

Description

This method takes a FIR containing raw fingerprint data for the purpose of creating a new enrollment template.

Parameter

capturedFIR : The FIR data

storedFIR : The FIR data

Payload : A user defined data that will be wrapped inside the newly created template.

Relation Property

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

TextEncodeFIR : Text encoded FIR data.

FIRLength : Size of the FIR data.

FIR : FIR data imported.

ImportImage(long ImageWidth, long ImageHeight, VARIANT RawImage)

Description

This method is to import the FIR data of RawImage

Parameter

ImageWidth : A value indicating the Width of RawImage.

ImageHeight : A value indicating the Height of RawImage.

RawImage : The Raw data to be imported.

Relation Property

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

TextEncodeFIR : Text encoded FIR data.

FIRLength : Size of the FIR data.

FIR : FIR data imported.

CheckQuality(VARIANT storedFIR, VARIANT auditFIR)

Description

This method is to get the quality value of fingerprint data.

Parameters

storedFIR : The FIR data to be checked quality.

auditFIR : The audit data to be checked quality.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

QualityInfo: quality information of fingerprint data.

A.6 FPImage Object

This object can be used to achieve a fingerprint image data when capturing.

A.6.1 Property

long ErrorCode

Records the error code used in the last method or property. The value of 0 indicates success, and all other values represent an error condition.

BSTR ErrorDescription

Records the error description of text format, corresponding the error code.

long TotalFingerCount

Indicates the number of fingers converted from the FIR data.

long FingerID(long nIndex)

Indicates the finger IDs of fingerprint data converted from the FIR data.
Indexes range from 0 to the TotalFingerCount - 1.

long ImageWidth

Indicates the image width in pixel.

long ImageHeight

Indicates the image height in pixel.

VARIANT RawData(long nFingerID, /*[in, optional]*/ long nSampleNumber)

Retrieves the fingerprint data in binary type.
The nFingerID must be entered as the first parameter.
The nSampleNumber is not used, enter 0.

long ConvertImageWidth

Indicates the result of a ConvertWsqToRaw Method

long ConvertImageHeight

Indicates the result of a ConvertWsqToRaw Method

BSTR TextEncodeAuditData

Retrieves the text encoded Audit data.

Long AuditdataLength

Retrieves the size, in bytes, of the Audit data.

VARIANT Auditdata

Retrieves the Audit data in binary type.
The buffer in size of the AuditdataLength must be allocated before use of the methods using this property.

A.6.2 Method

Export()

Description

This method is to retrieve fingerprint image data captured by using the Enroll or Capture method.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

TotalFingerCount : Indicates the number of fingers exported.

FingerID : Indicates the finger IDs exported.

ImageWidth : Indicates the image width.

ImageHeight : Indicates the image height.

RawData : Fingerprint data in raw type.

ExportEx(VARIANT auditData)

Description

This method is to retrieve fingerprint image data from the auditData.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

TotalFingerCount : Indicates the number of fingers exported.

FingerID : Indicates the finger IDs exported.

ImageWidth : Indicates the image width.

ImageHeight : Indicates the image height.

RawData : Fingerprint data in raw type.

Save(BSTR bszImgFilePath, long nImageType, long nFingerID, /*[in, optional]*/ long nSampleNumber)

Description

This method is to save fingerprint image data into a file specified.

Parameters

bszImgFilePath : The fullpath to save fingerprint image

nImageType : A value indicating the image type.

nFingerID : Indicates the finger ID to be saved.

nSampleNumber : Indicates the sample number to save. This value is not used.

Relation Property

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

ConvertWsqToRaw(long WsqLen, VARIANT WsqImage)

Description

This method is to convert the Wsa data to Raw data format.

Parameter

WsqLen : A value indicating the size of WsqImage.

WsqImage : The Wsq data to be converted.

Relation Property

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

Return Value

VARIANT RawImage

Retrieves the Raw data in binary type.

ConvertRawToWsq(long ImageWidth, long ImageHeight, VARIANT RawImage, float fQuality)

Description

This method is to convert the Raw data to Wsq data format.

Parameter

ImageWidth : A value indicating the Width of RawImage.

ImageHeight : A value indicating the Height of RawImage.

RawImage : The Raw data to be converted.

fQuality : Wsq Quality (0.1 ~ 7.0)

The default is 0.75 (15:1), If a lower fQuality is selected, the Image distortion will be increased.

Relation Property

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

Return Value

VARIANT WsqImage

Retrieves the Wsq data in binary type.

A.7 IndexSearch Object

This object is to store a large number of fingerprint data into the memory DB, and to search a specific data in the memory. Compared to the NSearch engine, the IndexSearch engine can be used for large database, but smaller volume than the NSearch engine.

A.7.1 Property

long ErrorCode

Records the error code used in the last method or property. The value of 0 indicates success, and all other values represent an error condition.

BSTR ErrorDescription

Records the error description of text format, corresponding the error code.

long Count

Indicates the number of fingers searched.

long MaxCandidatenumber

This property is used to set the maximum number of candidates.

long GetDataCountFromDB

This property is used to read the number of data in the memory DB.

BOOL CheckDataExistFromDB(long nUserID, long nFingerID, long nSampleNumber)

This property can be used to check if a specific data exists in the memory DB.

long UserID

Retrieves the user ID as a result of identification.

long MaxSearchTime

This property is used to set the maximum search time.

A.7.2 Property(CandidateList Object)

This object is to acquire the result of fingerprint enrollment and identification. This object, created as a collection, can be used without declaration.

long UserID

Indicates the user ID that is in number.

long FingerID

Indicates the finger ID.

long SampleNumber

Indicates the sample number, 0 or 1.

A.7.3 Method

AddFIR (VARIANT FIR, long nUserID)

Description

This method is to add a FIR data into memory DB.

Parameters

FIR : A FIR data to be added into the memory DB.

nUserID : The user ID of the FIR. Must be in number.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

CandidateList Object : Contains the result of enrollment. The ConfidenceLevel is not included for enrollment.

RemoveData(long nUserID, long nFingerID, long nSampleNumber)

Description

This method is to delete a specific fingerprint data from the memory DB.

It can be used to delete a single fingerprint data from a user.

Parameters

nUserID : The user ID of the data to be deleted. Must be in number.

nFingerID : The finger ID to be deleted.

nSampleNumber : The sample number to be deleted. Must be 0 or 1.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

RemoveUser(long nUserID)

Description

This method is to delete all fingerprint data of a user.

Parameters

nUserID : The user ID to be deleted. Must be in number.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

IdentifyUser(VARIANT storedFIR, long nSecuLevel)

Description

This method is to identify the fingerprint data as a result, TRUE or FALSE.

Parameters

storedFIR : The FIR data to be identified.

nSecuLevel : The security level for identification.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

UserID : The user ID identified.

ClearDB()

Description

This method is to clear the memory DB.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

SaveDBToFile(BSTR bszFilePath)

Description

This method is to save the memory DB into a file in disk.

It can be used before an application is closed to keep the memory DB into a file.

Parameters

bszFilePath : The fullpath to save the memory DB.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

LoadDBFromFile(BSTR bszFilePath)

Description

This method is to load the FDB file (saved by the SaveDBToFile method) into the memory DB.

Parameters

bszFilePath : The fullpath to load the FDB file.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

A.8 NSearch Object

This object is to store a large number of fingerprint data into the memory DB, and to search a specific data in the memory.

For more information, please refer to the NSearch engine manual.

A.8.1 property

long ErrorCode

Records the error code used in the last method or property. The value of 0 indicates success, and all other values represent an error condition.

BSTR ErrorDescription

Records the error description of text format, corresponding the error code.

long Count

Indicates the number of fingers searched.

long MaxCandidatenumber

This property is used to set the maximum number of candidates.

long GetDataCountFromDB

This property is used to read the number of data in the memory DB.

BOOL CheckDataExistFromDB(long nUserID, long nFingerID, long nSampleNumber)

This property can be used to check if a specific data exists in the memory DB.

long UserID

Retrieves the user ID as a result of identification.

A.8.2 property (CandidateList Object)

This object is to acquire the result of fingerprint enrollment and identification.

long UserID

Indicates the user ID that is in number.

long FingerID

Indicates the finger ID.

long SampleNumber

Indicates the sample number, 0 or 1.

long ConfidenceLevel

Indicates the confidence level that means the matching score. Values range from 1 to 9 (closest).

A.8.3 Method

AddFIR (VARIANT FIR, long nUserID)

Description

This method is to add a FIR data into memory DB.

Parameters

FIR : A FIR data to be added into the memory DB.

nUserID : The user ID of the FIR. Must be in number.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

CandidateList Object : Contains the result of enrollment.

RemoveData(long nUserID, long nFingerID, long nSampleNumber)

Description

This method is to delete a specific fingerprint data from the memory DB.

Parameters

nUserID : The user ID of the data to be deleted. Must be in number.

nFingerID : The finger ID to be deleted.

nSampleNumber : The sample number to be deleted. Must be 0 or 1.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

RemoveUser(long nUserID)

Description

This method is to delete all fingerprint data of a user.

Parameters

nUserID : The user ID to be deleted. Must be in number.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

SearchData(VARIANT storedFIR)

Description

This method is to search all data in the memory DB and list up the candidates.

Parameters

storedFIR : A FIR data to be searched.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

CandidateList Object : Contains the result of searching.

IdentifyUser(VARIANT storedFIR, long nSecuLevel)

Description

This method is to identify the fingerprint data as a result, TRUE or FALSE.

Parameters

storedFIR : The FIR data to be identified.

nSecuLevel : The security level for identification.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

UserID : The user ID identified.

ClearDB()

Description

This method is to clear the memory DB.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

SaveDBToFile(BSTR bszFilePath)

Description

This method is to save the memory DB into a file in disk.

Parameters

bszFilePath : The fullpath to save the memory DB.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

LoadDBFromFile(BSTR bszFilePath)

Description

This method is to load the FDB file (saved by the SaveDBToFile method) into the memory DB.

Parameters

bszFilePath : The fullpath to load the FDB file.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.

ImportIndexSearchDB(BSTR bszFilePath)

Description

This method is to load the FDB file (saved by the IndexSearch Engine) into the memory DB.

Parameters

bszFilePath : The fullpath to load the FDB file.

Relation Properties

ErrorCode : Indicates the result of the method. The value NBioAPIERROR_NONE indicates success, and all other values represent an error condition.

ErrorDescript : Description of the error code.