

# **IntegriSign Capture SDK**

## **Developer's Guide**

# CONTENTS

<b>1. Introduction .....</b>	<b>1</b>
<b>2. IntegriSign Capture SDK Key Features .....</b>	<b>1</b>
<b>3. IntegriSign Capture SDK Integration Overview.....</b>	<b>1</b>
<b>4. Pre-requisites .....</b>	<b>2</b>
<b>5. Components of IntegriSign Capture SDK.....</b>	<b>2</b>
<b>6. SDK Deployment.....</b>	<b>2</b>
<b>7. Steps to use IntegriSign Capture SDK.....</b>	<b>3</b>
<b>8. IntegriSign Capture SDK API.....</b>	<b>3</b>
8.1 esW25COM Component .....	3
8.1.1 Properties .....	3
8.1.1.1 HashData .....	3
8.1.1.2 BackColor.....	3
8.1.1.3 ForeColor.....	3
8.1.1.4 Cross .....	4
8.1.1.5 SignDlgCaption .....	4
8.1.1.6 ShowSignerName .....	4
8.1.1.7 ImageThickness .....	4
8.1.1.8 ConnectedDevice .....	4
8.1.1.9 DisplayName .....	4
8.1.1.10 AvailableStartY .....	5
8.1.1.11 AvailableEndY .....	5
8.1.1.12 EnableAntiAliasing.....	5
8.1.1.13 SignThickness.....	5
8.1.2 Events supported.....	5
8.1.3 Signing Methods/Functions .....	6
8.1.3.1 SetSignerDetailsEx .....	6
8.1.3.2 GetSignerDetailsEx .....	6
8.1.3.3 StartSign .....	7
8.1.3.4 GetSignData.....	7
8.1.3.5 OpenSign .....	8
8.1.3.6 ClearSign .....	8
8.1.3.7 IsSigned .....	8
8.1.3.8 SaveToFile.....	9
8.1.3.9 VerifyContents.....	9
8.1.3.10 GetImageData.....	9
8.1.3.11 GetImageBytes .....	10
8.1.3.12 SetSignAppearanceOptionsEx.....	11
8.1.3.13 SetBkImage.....	11
8.1.3.14 GetDevices.....	12
8.1.3.15 setePadDevice.....	13
8.1.3.16 GenerateImageFromBase64.....	13
8.1.3.17 GenerateImageFromBinary .....	13
8.1.3.18 CloseConnection.....	14

8.1.3.19 SetSignerDetails .....	14
8.1.3.20 GetSignerDetails .....	14
8.1.3.21 SetSignAppearanceOptions .....	15
8.1.3.22 SetAffirmationText .....	16
8.1.3.23 SetUserInfo .....	16
8.1.3.24 SetBkImageEx .....	17
8.1.3.25 StartSignEx .....	19
8.1.3.26 SetSignRect .....	19

## **9. Deploying IntegriSign Capture SDK Components in Windows 2003 64-bit for applications running in IIS ..... 19**

## 1. Introduction

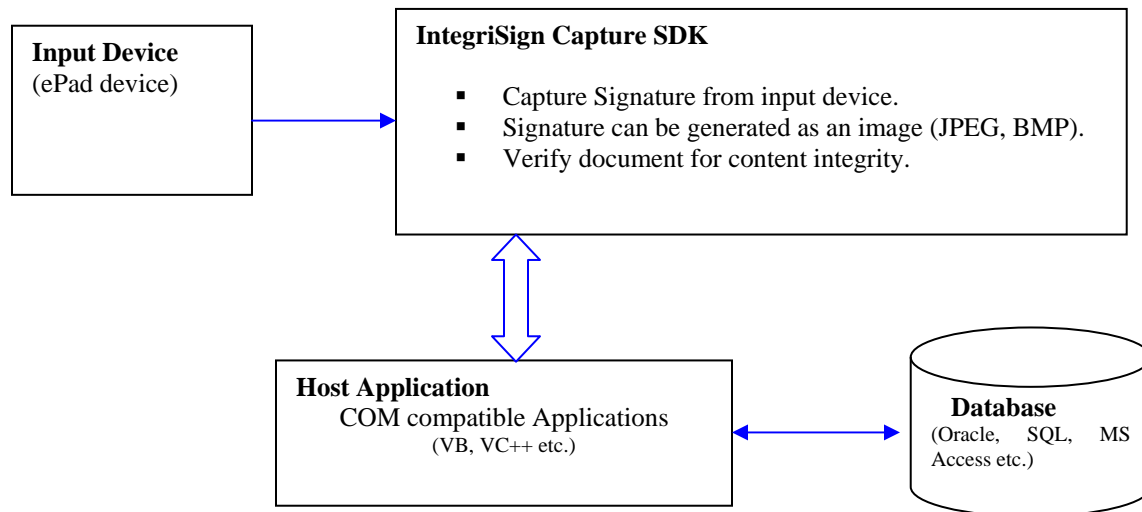
Thank you for choosing IntegriSign Capture SDK, the most comprehensive signature capture software SDK in the market. IntegriSign Capture SDK gives you the features and functionality to Sign and Secure electronic documents/e-mails of any COM compatible applications, using powerful authentication tools. IntegriSign Capture SDK comes with built-in support for handwritten electronic signatures. Electronic signature capture is done through ePad device.

## 2. IntegriSign Capture SDK Key Features

- Enables applications with Electronic Handwritten Signature capability to capture the signature and secure the content using advanced encryption technologies.
- Allows signature capture through ePad device.
- Simultaneously captures the signer details, time-stamp of signing, location that provide validity to authenticated electronic content.
- Signed content integrity can be verified at any point of electronic transmission or storage.

## 3. IntegriSign Capture SDK Integration Overview

IntegriSign APIs are bundled within the DLL along with standard User Interface elements. When the host application invokes these API functions, to initiate the act of Signing [StartSign ()], the signature from the input device is captured along with other signer details. This signature data along with content hash [HashData] is encrypted and processed into BASE64 string format.



**Input Device:** ePad device is used for signature capture.

**Host Application:** Any COM compliant application like Visual Basic, VC++ etc.

**IntegriSign Capture SDK:** IntegriSign Capture SDK contains capture component (COM Object with User Interface) and supporting libraries. API functions are available to perform

different tasks such as to generate signature image [SaveToFile ()] and check content Integrity [VerifyContents ()].

**Database:** The captured signature data (Encrypted Base64 string) can be stored in the host database (Oracle, SQL, DB2, Sybase, MS Access) or in a file system. The stored data is interpreted by IntegriSign API function [OpenSign ()] to reproduce the signature at a later time.

## 4. Pre-requisites

Optimum System Requirements to install and run IntegriSign Capture SDK

Windows NT 4.0 (Build 1381 with Service Pack 4 or higher) operating system

Or

Windows 2000/XP/2003/Vista/Windows 7 operating system

Pentium-class PC

ePad device.

32 megabytes of RAM

10 megabytes of free disk space

## 5. Components of IntegriSign Capture SDK

The following are the components required by IntegriSign Capture SDK to integrate into any COM enabled application.

1. esW25COM.ocx – The core component responsible for signing of the electronic documents, setting the user preferences and binding the content of the documents to signatures.
2. esUtil.dll - Supporting library for IntegriSign Capture SDK.

## 6. SDK Deployment

To deploy the IntegriSign Capture SDK components, copy the following files from the SDK folder in the zip file into the client computer.

NOTE: - File 1 (esW25COM.ocx) can be copied into any of the folders on the client computer and must be registered. File 2 (esUtil.dll) must be copied into the windows system folder.

1. esW25COM.ocx (This ocx must be registered).
2. esUtil.dll

## 7. Steps to use IntegriSign Capture SDK

Follow these steps to make your application IntegriSign enabled.

1. Install the IntegriSign Capture SDK onto your system.
2. List the documents (forms) that are to be signed.
3. Add the Signature component (IntegriSign Core Component for Capture SDK) to the forms by referring to esW25COM.ocx.
4. Set the **HashData** property on the instance of esW25COM component wherever the content integrity is necessary. Typically it should be set before the **StartSign**, **VerifyContents** and **OpenSign** methods.
5. Implement other functionality for Capturing, Storing & Opening signatures taking the following SDK reference.

## 8. IntegriSign Capture SDK API

Following are the properties, methods and functions that can be used to enable an application with IntegriSign Capture functionality. These functions are available through the esW25COM component.

### 8.1 esW25COM Component

#### 8.1.1 Properties

##### *8.1.1.1 HashData*

##### **Description**

This property has to be set with the data that has to be authenticated. These values are taken from the present container viz. a form or document. To implement the content integrity functionality this property has to be set before the sign capturing process. This should also be set wherever you need to check the authenticity of the content. Typically this should be called before using the following methods.

- 1 StartSign
- 2 OpenSign
- 3 VerifyContents

*Note: When this property is set to null, hashing is not implemented.*

##### *8.1.1.2 BackColor*

##### **Description**

Sets the background color of the signature component.

##### *8.1.1.3 ForeColor*

##### **Description**

Sets the color of the signature.

#### *8.1.1.4 Cross*

##### **Description**

Sets whether the cross-mark needs to be displayed on the signature when the content authentication failed. Two possible values are CROSS\_OFF and CROSS\_ON.

#### *8.1.1.5 SignDlgCaption*

##### **Description**

Sets caption for the signature capture dialog.

#### *8.1.1.6 ShowSignerName*

##### **Description**

Sets whether Signer Name field should be displayed in the signature capture dialog. Two possible values are TRUE or FALSE.

#### *8.1.1.7 ImageThickness*

##### **Description**

Sets the pen thickness to be used while generating the image of the signature. The values can range from 1 to 10. Default value for this property is 2.

#### *8.1.1.8 ConnectedDevice*

##### **Description**

Returns the Identifier for the currently connected ePad device. Currently it can be one of the following values.

{98C174D3-6D2A-4509-96DB-D5B34FA7A561}: ePadInk

{7FCD9512-8763-436E-8747-40972EE28EFD}: ePad

{C55C5D54-8A92-48AD-A32F-1FC58092A581}: ePadID

#### *8.1.1.9 DisplayName*

##### **Description**

Sets the text to be displayed on the signature object. This text appears when the signature object is not signed. If text needs to be displayed on multiple lines, insert “\n” wherever required.

#### *8.1.1.10 AvailableStartY*

##### **Description**

Gets the starting point on the Y co-ordinate system available for the Inking Region. The value returned by this property can be used when defining the Inking Region in the SetBkImageEx function. This property needs to be invoked after SetAffirmationText and SetUserInfo methods.

#### *8.1.1.11 AvailableEndY*

##### **Description**

Gets the ending point on the Y co-ordinate system available for the Inking Region. The value returned by this property can be used when defining the Inking Region in the SetBkImageEx function. This property needs to be invoked after SetAffirmationText and SetUserInfo methods.

#### *8.1.1.12 EnableAntiAliasing*

##### **Description**

Setting this property causes the signature drawing algorithm to use Anti-aliasing technique to smoothen the signature. SignThickness value will not be used when Anti-Aliasing is used.

#### *8.1.1.13 SignThickness*

##### **Description**

Setting this property causes the signature drawing algorithm to use the specified thickness to draw the signature. Anti-aliasing technique will not be used.

### **8.1.2 Events supported**

The developer may handle/override the following events from IntegriSign Component.

**DblClick**  
**MouseDown**  
**MouseMove**  
**MouseUp**

**SignCompleteStatus** (SignStatus as Long)

##### **Description**

Event indicating whether the signature field was signed or not. If the signing process is incomplete (i.e. cancelled) SignStatus is 0 else SignStatus is 1.



## **OnFirstTouch ()**

### ***Description***

Event fires when the user first touches the device (i.e. when the user starts inputting the signature).

## **8.1.3 Signing Methods/Functions**

### ***8.1.3.1 SetSignerDetailsEx***

#### ***Description***

The Signer details like Name, Organization etc. can be set using this method. This information is set so that it can be used as signer reference. This method, if implemented, should be called before the **StartSign** method to define 'who' is signing.

#### ***Parameters***

Name (In) – Data type String - Name of the signer

OrgUnit (In) – Data type String - Organization unit of the signer

OrgName (In) – Data type String - Organization name of the signer

City (In) – Data type String - Location of the signer

State (In) – Data type String – State

Country (In) – Data type String - Country of the signer

eMail (In) – Data type String – Signer eMail address

SignLocation (In) – Data type String - Location from where the signer is signing.

*Note: All the Parameters can take NULL values.*

#### ***Return Value***

This function returns nothing.

### ***8.1.3.2 GetSignerDetailsEx***

#### ***Description***

This method returns the Signer details stored in the signature component.

#### ***Parameters***

Name (In/Out) – Data type String - Name of the signer

OrgUnit (In/Out) – Data type String - Organization unit of the signer

OrgName (In/Out) – Data type String - Organization name of the signer

City (In/Out) – Data type String - Location of the signer

State (In/Out) – Data type String – State

Country (In/Out) – Data type String - Country of the signer

eMail (In/Out) – Data type String – Signer eMail address

SignLocation (In/Out) – Data type String - Location from where the signer is signing.

*Note: All the Parameters can take NULL values.*

### **Return Value**

Returns values are set through the parameters in the above order.

### **8.1.3.3 StartSign**

#### **Description**

Invokes Signature capture Dialog and user can start the “act of signing”.

#### **Parameters**

bs – Type ButtonStyle – To specify the type of buttons required.

There are three Button Styles available viz., BT\_OK, BT\_OK\_CLEAR and BT\_OK\_CANCEL\_CLEAR (default button style) and BT\_NONE.

int signDlgReq – To specify whether a separate signature capture dialog is to be displayed or not. If signing dialog is not shown, scribbling on the pad directly reflects on the signature control. This parameter is optional and if not specified, signing dialog is shown.

#### **Return Value**

StartSign method returns following values depending on the hotspots clicked.

BT\_PRESSED\_OK

BT\_PRESSED\_CANCEL

### **8.1.3.4 GetSignData**

#### **Description**

This function returns the signature data in an encrypted Base64string Format. If not signed, it returns a null string.

#### **Parameters**

This takes no inputs.

#### **Return Value**

Signature data in encrypted Base64String format.

#### [\*8.1.3.5 OpenSign\*](#)

##### **Description**

Use this function to open and view the signature from the encrypted Base64String on the signature component.

##### **Parameters**

StrData – An IntegriSign signature string of encrypted Base64String Format.

##### **Return Value**

This function returns nothing.

*Note: Method **GetSignData** was used originally to generate encrypted base64string from the signature component.*

#### [\*8.1.3.6 ClearSign\*](#)

##### **Description**

This clears the signature on the component, if any.

##### **Parameters**

This takes no inputs.

##### **Return Value**

This function returns nothing.

#### [\*8.1.3.7 IsSigned\*](#)

##### **Description**

Indicates whether the component is signed or not.

##### **Parameters**

This takes no inputs

##### **Return Value**

Returns a Long.

0 – The component is not signed

1 – The component is signed.

#### [8.1.3.8 SaveToFile](#)

##### **Description**

Saves the signature as an image file (bmp, jpeg, gif) in the set location.

##### **Parameters**

FileName – Data type String - Complete file Path where the bmp image is to be saved.

nHeight – Data type Integer - Height of the bmp image in pixels.

nWidth – Data type Integer - Width of the bmp image in pixels.

FileType - Data type FILETYPE – Indicates type of file (image) storage.  
(BMP = 0, JPEG = 1, GIF=2)

ImageQuality – Data type Integer – For image quality (jpeg). An Optional parameter.

GIFTransparency – Data type Integer – For Non-transparent GIF the value is 0 and for transparent GIF the value is 1. An Optional parameter.

*Note: ImageQuality should be between 0 and 100. If the parameter is not set or is set to zero ImageQuality is taken as 80 by default.*

##### **Return Value**

This returns nothing.

#### [8.1.3.9 VerifyContents](#)

##### **Description**

Performs the hashing operation and returns the content integrity status. **HashData** property has to be set before calling this method.

##### **Parameters**

This takes no inputs.

##### **Return Value**

0 – Hashed content has not changed since signing.

1 – Hashed content was modified after signing.

2 – Hashing was not implemented.

#### [8.1.3.10 GetImageData](#)

**Description**

Returns encoded (Base64) Signature image data for storage as string.

**Parameters**

nHeight – Data type Integer - Height of the bmp image in pixels.

nWidth – Data type Integer - Width of the bmp image in pixels.

FileType - Data type FILETYPE – Indicates type of file (image) storage.  
(BMP = 0, JPEG =1, GIF=2)

ImageQuality – Data type Integer – For image quality (jpeg). An Optional parameter.

GIFTransparency – Data type Integer – For Non-transparent GIF the value is 0 and for transparent GIF the value is 1. An Optional parameter.

*Note: ImageQuality should be between 0 and 100. If the parameter is not set or is set to zero ImageQuality is taken as 80 by default.*

**Return Value**

ImageData – Data type String - Encoded (Base64) Signature image data for storage.

### [8.1.3.11 GetImageBytes](#)

**Description**

Returns Signature image data for storage as a Byte array.

**Parameters**

nHeight – Data type Integer - Height of the bmp image in pixels.

nWidth – Data type Integer - Width of the bmp image in pixels.

FileType - Data type FILETYPE – Indicates type of file (image) storage.  
(BMP = 0, JPEG =1, GIF=2)

ImageQuality – Data type Integer – For image quality (jpeg). An Optional parameter.

GIFTransparency – Data type Integer – For Non-transparent GIF the value is 0 and for transparent GIF the value is 1. An Optional parameter.

*Note: ImageQuality should be between 0 and 100. If the parameter is not set or is set to zero ImageQuality is taken as 80 by default.*

**Return Value**

ImageBytes – Data type ByteArray - Signature image bytes for storage.

#### [\*8.1.3.12 SetSignAppearanceOptionsEx\*](#)

##### **Description**

This function lets you specify the information to be displayed and printed along with the signature.

##### **Parameters**

PenColor (In) – Data type Long

ShowName (In) – Data type Integer

ShowDate (In) – Data type Integer

ShowLogo (In) – Data type Integer

ShowLocation (In) – Data type Integer

ShowLabels (In) – Data type Integer

*For all the above parameters a value of 1 means true and 0 means false.*

##### **Return Value**

This function does not return any value.

#### [\*8.1.3.13 SetBkImage\*](#)

##### **Description**

You can specify the bitmap and the coordinates for ink region (when user scribbles) using SetBkImage method. This method should be called before StartSign method.

##### **Parameters**

Filename (In) – Data type String – Complete path of the image file (including name) to be set for display on the ePadInk device.

*The selected bitmap should be a monochrome image with 320 X 200 resolution.*

LeftX (In) – Data type Integer - Left X co-ordinate of the area

TopY (In) – Data type Integer - Top Y co-ordinate of the area.

RightX (In) – Data type Integer - Right X co-ordinate of the area.

BottomY (In) – Data type Integer - Bottom Y co-ordinate of the area.

##### **Return Value**

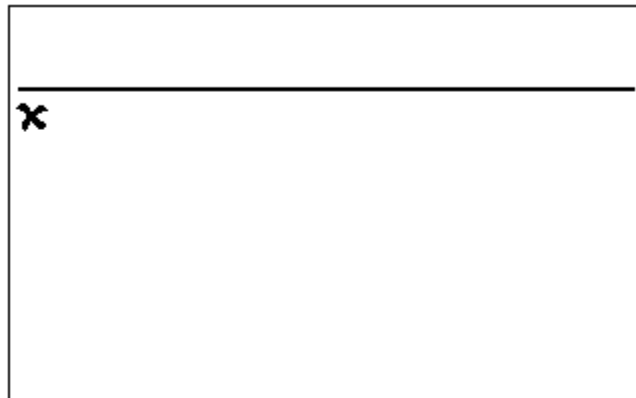
This function does not return any value.

### **Image Orientation:**

As explained above, the image should be a monochrome image with 320 \* 200 pixels and it should be flipped vertically. For example, if you want the image to be displayed on the device as



the original image should be as shown below.



**NOTE:** It is recommended to use the SetBkImageEx method instead of the SetBkImage method. SetBkImage method is maintained to support legacy applications developed using versions of IntegriSign Desktop before 7.5.4.

#### ***8.1.3.14 GetDevices***

##### ***Description***

Enumerates the all the installed device providers and returns an array containing their information i.e. ClassID and Description.

##### ***Parameters***

No input parameters

##### ***Return Value***

String Array, containing the list of providers.

#### *8.1.3.15 setePadDevice*

##### **Description**

Sets the device for initiating the act-of-signing. The device to be set is selected from the list of installed providers.

##### **Parameters**

ClassID of the device as String.

##### **Return Value**

No return value.

#### *8.1.3.16 GenerateImageFromBase64*

##### **Description**

Generates the signature image from the supplied Base64 image string.

##### **Parameters**

ImageStr – Image data as Base64 string.

FilePath – File Path including the name where the image needs to be saved.

##### **Return Value**

Long – Indicates whether the image is saved successfully.

0 – Success

1 – Failure

#### *8.1.3.17 GenerateImageFromBinary*

##### **Description**

Generates the signature image from the supplied binary data.

##### **Parameters**

ImageData – Image data as a byte array.

FilePath – File Path including the name where the image needs to be saved.

##### **Return Value**

Long – Indicates whether the image is saved successfully.

0 – Success

1 – Failure



### *8.1.3.18 CloseConnection*

#### **Description**

Closes the connection with the ePad device.

#### **Return Value**

No Return Value.

### *8.1.3.19 SetSignerDetails*

#### **Description**

The Signer details like Name, Organization etc. can be set using this method. This information is set so that it can be used as signer reference. This method, if implemented, should be called before the **StartSign** method to define 'who' is signing.

#### **Parameters**

Name (In) – Data type String - Name of the signer

OrgUnit (In) – Data type String - Organization unit of the signer

OrgName (In) – Data type String - Organization name of the signer

City (In) – Data type String - Location of the signer

State (In) – Data type String – State

Country (In) – Data type String - Country of the signer

eMail (In) – Data type String – Signer eMail address

SignLocation (In) – Data type String - Location from where the signer is signing.

bReserved(In) – Data type String - reserved parameter.

*Note: All the Parameters can take NULL values.*

#### **Return Value**

This function returns nothing.

**NOTE:** It is recommended to use the SetSignerDetailsEx method instead of the SetSignerDetails method. SetSignerDetails method is maintained to support legacy applications.

### *8.1.3.20 GetSignerDetails*

#### **Description**

This method returns the Signer details stored in the signature component.

**Parameters**

Name (In/Out) – Data type String - Name of the signer

OrgUnit (In/Out) – Data type String - Organization unit of the signer

OrgName (In/Out) – Data type String - Organization name of the signer

City (In/Out) – Data type String - Location of the signer

State (In/Out) – Data type String – State

Country (In/Out) – Data type String - Country of the signer

eMail (In/Out) – Data type String – Signer eMail address

SignLocation (In/Out) – Data type String - Location from where the signer is signing.

bReserved (In/Out) – Data type String – reserved parameter.

*Note: All the Parameters can take NULL values.*

**NOTE:** It is recommended to use the `GetSignerDetailsEx` method instead of the `GetSignerDetails` method. `GetSignerDetails` method is maintained to support legacy applications.

**8.1.3.21 SetSignAppearanceOptions**

**Description**

This function lets you specify the information to be displayed and printed along with the signature.

**Parameters**

PenColor (In) – Data type Long

ShowName (In) – Data type Integer

ShowDate (In) – Data type Integer

ShowLogo (In) – Data type Integer

sReserved (In) – Data type Integer. A reserved parameter.

ShowLocation (In) – Data type Integer

ShowLabels (In) – Data type Integer

*For all the above parameters a value of 1 means true and 0 means false.*

**Return Value**

This function does not return any value.

NOTE: It is recommended to use the SetSignAppearanceOptionsEx method instead of the SetSignAppearanceOptions method. SetSignAppearanceOptions method is maintained to support legacy applications.

#### *8.1.3.22 SetAffirmationText*

##### **Description**

Sets the affirmation text to be displayed on the ePadInk device during signature capture.

##### **Parameters**

Affirmation Text – Type String

##### **Return Value**

Returns the error number as Integer.

0 – No error

11 – Text length is too long

12 – More number of lines are required to display the text than supported 6 lines.

13 – No Data

Note: Number of lines supported for Affirmation Text on the ePadInk device is 6. If affirmation text needs to be displayed in multiple lines as intended by the application, insert “\n” wherever required. If “\n” is not found in the text, the algorithm in the component breaks the data into multiple lines accordingly.

#### *8.1.3.23 SetUserInfo*

##### **Description**

Sets the User Information to be displayed below the signing area on the ePadInk device during signature capture.

##### **Parameters**

User Info – Type String

##### **Return Value**

Returns the error number as Integer.

0 – No error

11 – Text length too long

12 – More number of lines are required to display the text than supported 2 lines.

13 – No Data

Note: Number of lines supported for User Information on the ePadInk device is 2. If User Information needs to be displayed in multiple lines as intended by the application, insert “\n” wherever required. If “\n” is not found in the text, the algorithm in the component breaks the data into multiple lines accordingly.

#### *8.1.3.24 SetBkImageEx*

##### **Description**

Sets the bitmap and the coordinates for ink region (when user scribbles) using SetBkImageEx method. This method should be called before StartSignEx method and after SetAffirmationText, SetUserInfo methods. If the last optional parameter FolderName is not specified, by default the folder is assumed as Windows system folder.

##### **Parameters**

Filename (In) – Data type String – Image file name excluding path. Currently only BMP images are supported.

StartX (In) – Data type Integer - Starting X co-ordinate of the area- Value can be between 0 and 320.

StartY (In) – Data type Integer – Top Left Y co-ordinate of the area. Value should be in between the values obtained from properties [AvailableStartY](#) and [AvailableEndY](#).

EndX (In) – Data type Integer - Ending X co-ordinate of the area- Value can be between 0 and 320.

EndY (In) – Data type Integer – Bottom Right Y co-ordinate of the area. Value should be in between the values obtained from properties [AvailableStartY](#) and [AvailableEndY](#). The maximum height is 199 pixels.

The above four parameters define the Inking Region on the screen.

FolderName (In) – Optional Parameter – Folder Path of the Image file if it is other than Windows System folder.

##### **Return Value**

Returns the error number as Integer.

- 0 - No error
- 1 - The specified Image File not found
- 2 - Invalid Image

- 3 - The specified image is not a monochrome image.
- 4 - Image height is greater than the available inking area.
- 5 - Image Width greater than the available inking area.
- 6 - Invalid ink region specified.

### **Image Orientation:**

#### **ePad-ink:**

The image should be a monochrome image. The colors in the image should be inverted. For example, if you want the image to be displayed on the device as



the original image should be as shown below.



#### **ePadLS:**

The image is displayed on the device as-is. It can contain colors.

#### **ePad Pro Devices:**

The image is displayed on the device as-is, but it should be a monochrome image.

### **ePad-vision:**

Currently ePad-vision does not support this method. Even if this method used, the image specified would not be displayed on the device.

#### ***8.1.3.25 StartSignEx***

##### ***Description***

Invokes the “act of signing”.

##### ***Parameters***

*BUTTON\_TYPE\_EX bTypeEx* – Type ButtonStyle –

Possible values for this parameter are

*BT\_One\_Left = 0*  
*BT\_One\_Righ = 1*  
*BT\_One\_Center = 2*  
*BT\_Two\_Left = 3*  
*BT\_Two\_Right = 4*  
*BT\_Two\_Cente = 5*  
*BT\_Three = 6*

Captions – String containing the Captions for the buttons. The names of the buttons are to be separated by “~”. Example : “Cancel~Clear~Accept”

##### ***Return Value***

StartSignEx method returns the name of the button that was clicked by the User. For Example, Cancel or Clear or Accept.

#### ***8.1.3.26 SetSignRect***

##### ***Description***

Sets the percentage of the signature object to be used for drawing the signature.

##### ***Parameters***

WidthPercent – Percentage of Signature Object Width to be used. Default value is 80.

HeightPercent – Percentage of Signature Object Height to be used. Default value is 80.

##### ***Return Value***

No Return Value.

## **9. Deploying IntegriSign Capture SDK Components in Windows 2003 64-bit for applications running in IIS**

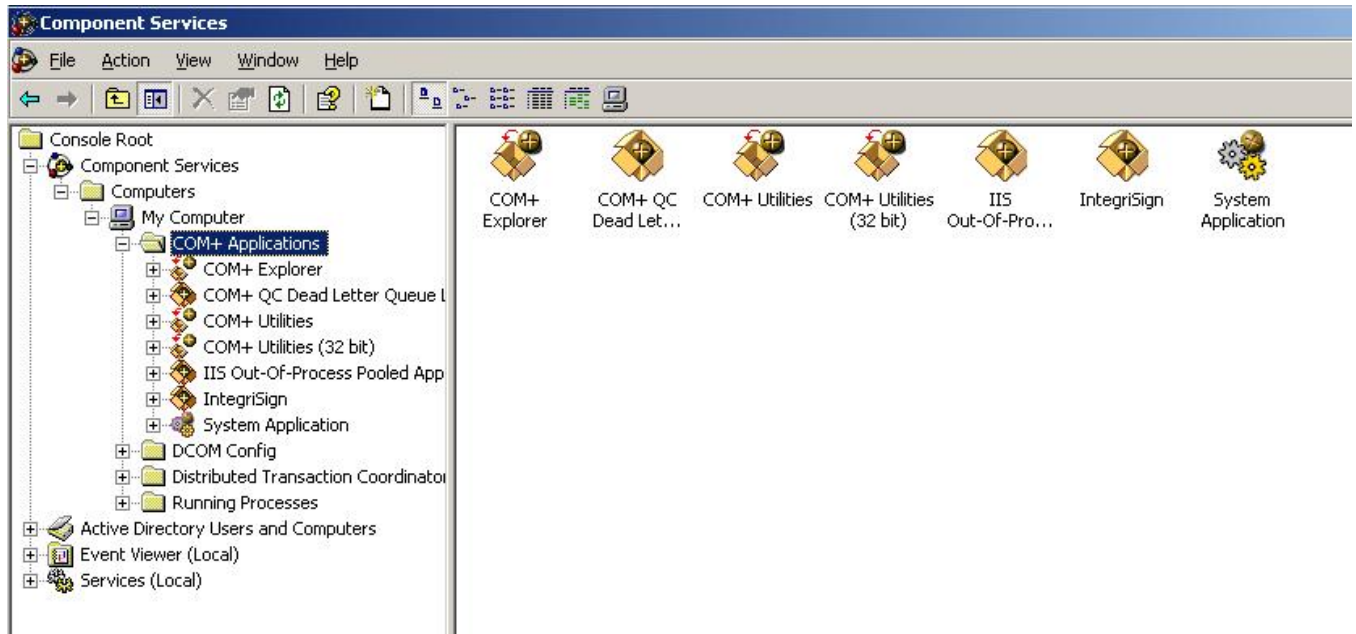
This section explains the additional steps required for using the IntegriSign Capture SDK core component in applications running in IIS on 64 bit Windows 2003 Operating system.

**Deployment Files required:**

1. Copy esW25COM.ocx into any folder on the machine and register it using regsvr32 utility.
2. Copy esUtil.dll to SysWOW64 folder under Windows folder (typically "C:\Windows\SysWOW64").

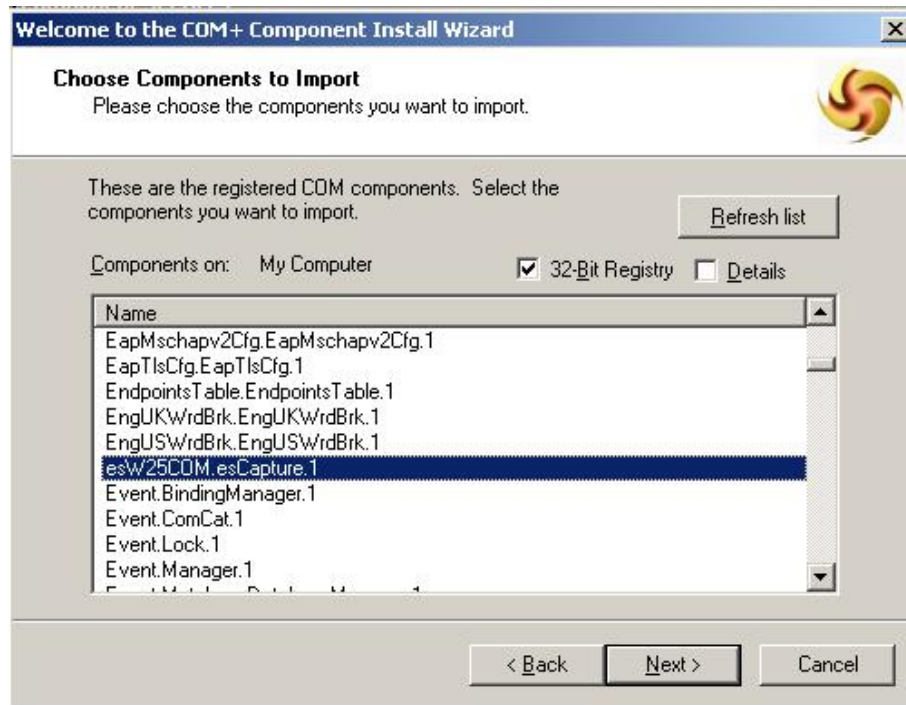
**Configuring the above esW25COM.ocx using Component Services to enable the component for running applications under IIS:**

1. Open Component Services from "Control Panel -> Administrative Tools".
2. Navigate to "Component Services -> My Computer -> COM+ Applications".
3. Right Click on "COM+ Applications" and select "New -> Application".
4. Click "Next" in the "COM+ Application Installation Wizard".
5. Click on "Create an empty application".
6. Enter a name (eg. IntegriSign) for the application and then click on "Next". Server application gets selected as default and do not change this option.
7. In the following Set Application Identity screen, click on "Next" without changing any selection. "Interactive User – currently logged on user" will be selected by default.
8. In the "Add Application Roles" select "CreatorOwner" and then click on "Next".
9. In the following screen (i.e. Add Users to Roles) expand "CreatorOwner" and select "Users". Click on "Add" button.
10. In the displayed window (i.e. Select Users or Groups), click on "Advanced" and then select "Find Now" in the displayed window.
11. In the displayed User List, select "Network Service" and click on "OK". Click on "OK" in the Select Users or Groups window too.
12. Click on "Next" in the Add Users to Roles window.
13. Finish the Wizard.
14. One will now be able to see the newly created Application under COM+ applications as shown below.



15. Now, select IntegriSign -> Components. Right click on "Components" and select "New -> Component". This will bring up the COM+ Component Install Wizard. Click on "Next".
16. In the displayed window (i.e. Import or Install a component) select "Import component(s) that are already registered".
17. In the displayed window (i.e. Choose components to Import), select the check box "32-Bit registry" and then select "esW25COM.esCapture.1" as shown below. Click on "Next" to continue.





18. Finish the COM+ Component Install Wizard by clicking on “Finish”.

Now one can run the applications under IIS without any problem. Once the above process is completed, one will not see the error “Retrieving the COM class factory for component with CLSID {84C046A7-4370-4D91-8737-87C12F4C63C5} failed due to the following error: 80040154” when using .NET as the platform.