

ePadInk SDK (ActiveX)

Developer's Guide

CONTENTS

1. Introduction	1
2. ePadInk SDK (ActiveX) Key Features	1
3. ePadInk SDK (ActiveX) Integration Overview	1
4. Pre-requisites	1
5. Components of ePadInk SDK (ActiveX)	1
6. ePadInk SDK (ActiveX) Samples	1
7. Steps to use ePadInk SDK (ActiveX)	1
7.1 Working with display of images.	1
7.2 Controlling the backlight	2
7.3 Working with hotspots (rectangles).....	2
7.4 To capture the ink	2
7.5 To clear the inepad	2
8. ePadInk SDK (ActiveX) API	3
8.1 Properties	3
8.1.1 TextAlignment	3
8.1.2 Light	3
8.1.3 Penwidth	4
8.1.4 FontName	4
8.2 Events supported.....	4
8.3 Methods/Functions	5
8.3.1 SetEventRect	5
8.3.2 SetInkRgn	6
8.3.3 GetImage	6
8.3.4 SetBitmap	6
8.3.5 CreateScreen.....	7
8.3.6 OpenConnection	8
8.3.7 CloseConnection.....	8
8.3.8 Clear	9

1. Introduction

Thank you for choosing ePadInk SDK (ActiveX). ePadInk SDK (ActiveX) gives you the features and functionality to build applications that leverage the functionality of ePadInk device.

2. ePadInk SDK (ActiveX) Key Features

- Enables building of applications using ePadInk.
- Allows applications to use bi-directional features of ePadInk.

3. ePadInk SDK (ActiveX) Integration Overview

ePadInk APIs are bundled within the OCX (ePadInkCtrl.ocx) along with standard User Interface elements. The host application has to invoke these API functions to build applications using this SDK.

4. Pre-requisites

Optimum System Requirements to install and run ePadInk SDK (ActiveX)

Windows XP/2003/Vista/Windows 7/Windows 8/Windows 10 operating system

Pentium-class PC

ePad-Ink device

32 megabytes of RAM

20 megabytes of free disk space

5. Components of ePadInk SDK (ActiveX)

The following are the components required by ePadInk SDK (ActiveX) to integrate into any COM enabled application.

1. ePadInkCtrl.ocx – The core component responsible for the bi-directional functionality of ePadInk SDK (ActiveX).

6. ePadInk SDK (ActiveX) Samples

Samples built using ePadInk SDK (ActiveX) can be accessed from Start→Programs→IntegriSign Desktop -> SDKs ->ePadInk SDK (ActiveX) → Samples.

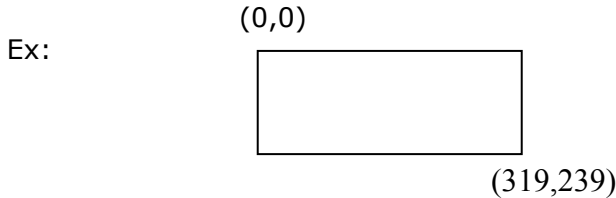
7. Steps to use ePadInk SDK (ActiveX)

Follow these steps to make your application use ePad-Ink bi-directional functionality.

7.1 Working with display of images.

1. A connection should be established with ePad-Ink using the *OpenConnection* API. This should be initial step for all the operations on ePad-Ink.

2. Set the bitmap image into the pad by specifying a screen-ID that can be referred later to display the required image. Also mention the layer on which the image should be displayed. (Refer to *SetBitmap* API)
3. We have 3 layers on ePad-Ink on which the images are displayed. The layer 0 is the one on which the user scribbles or interacts with ePad-Ink. This is always displayed. If layer-ID is 1 then the image overwrites the earlier image. If it is 2 then both the old and the new images are shown.

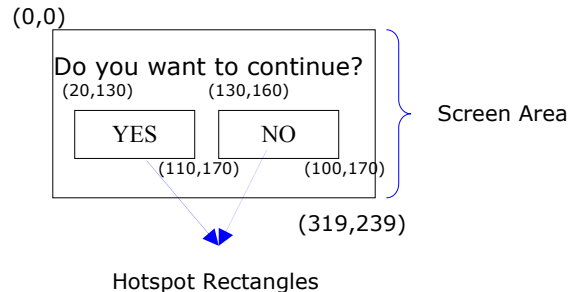


7.2 Controlling the backlight

The backlight on the ePad-Ink can be controlled using the SDK. The backlight state can be set to ON, OFF and AUTO depending on the requirement. When the backlight state is set to AUTO then the backlight turns off after user-defined number of seconds.

7.3 Working with hotspots (rectangles)

1. After establishing the connection, define the hotspots (rectangle areas) on the pad area by using the *SetEventRect* API.
2. Set the area size of the rectangles by specifying the co-ordinates w.r.t. the ePad screen area (319,239).
3. Specify an event-ID that can be referred to when *OnHandleData* event is raised.



4. When the hotspots are clicked, *OnHandleData* event is raised along with the Event-ID and Screen-ID. Depending on the functionality required the user can handle/override this event.

7.4 To capture the ink

1. After establishing the connection and optionally setting the bitmap image, you can set a whole/part of screen area to scribble.
2. This can be accomplished using *SetInkRgn* API specifying the area co-ordinates. This can be used to capture signatures from the ePadInk device.

7.5 To clear the inkpads

Use the clear API to clear the images in the inkpads layers. Specify the layer and the screen color you want to replace with.

8. ePadInk SDK (ActiveX) API

Following are the functions that can be used to build applications with ePad-Ink. These functions are available through the ePadInkCtrl component.

8.1 Properties

8.1.1 TextAlignment

Description

You can set the alignment of the text displayed on the ePad-Ink device using this property. Possible values are Vertical_Top and Vertical_Center.

Parameters

newVal (In) – Alignment

Alignment can be set as

Vertical_Top – 0

Vertical_Center – 1

Return Value

This function returns nothing.

8.1.2 Light

Description

The Back Light State on the surface of ePad-Ink can be set using this method.

Parameters

newVal (In) – State

State can be set as

S_OFF – Back Light in OFF state - 0

S_ON – Back Light in ON state - 1

S_Auto – Automatic Back Light switch off state - 2

Return Value

This function returns nothing.

8.1.3 Penwidth

Description

Sets the pen width.

Parameters

penwidth (In) – integer – The width to be set to.

Return Value

This function returns nothing.

8.1.4 FontName

Description

Sets the name of the font to be used to display the text on the ePad-Ink device. Default FontName is "Courier".

Parameters

newVal (In) – FontName

FontName can be set as

ARIAL = 0,
COURIER = 1,
MICROSOFT_SANS_SERIF = 2,
TAHOMA = 3,
TIMES_NEW_ROMAN = 4,
VERDANA = 5

Return Value

This function returns nothing.

8.2 Events supported

The developer may handle/override the following event from ePadInkCtrl Component.

OnHandleData

Description

OnHandleData event is raised along with the Event-ID and Screen-ID. Depending on the functionality required the user can handle/override this event.

Parameters

X (In) – Long – X co-ordinate of the device.

Y (In) – Long - Y co-ordinate of the device.

ScreenID – Long - Identifier for the current screen.

EventID – Long – Identifier to the button clicked.

Note: When a pre-defined template is selected, the EventIDs are:

OK	= 1
CANCEL	= 2
CLEAR	= 3
YES	= 4
NO	= 5
PREV	= 6
NEXT	= 7
ACCEPT	= 8
DECLINE	= 9

To create pre-defined screens based on templates, refer to the CreateScreen method.

8.3 Methods/Functions

8.3.1 SetEventRect

Description

The Area co-ordinates on the Ink Pad to be activated to respond to an event can be set using this method.

Parameters

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

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

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

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

EventID – Integer – ID of the Event to occur on click in the set area.

Return Value

This function returns nothing.

[8.3.2 SetInkRgn](#)

Description

This method can be used to set the Area co-ordinates on the Ink Pad where Ink needs to be visible when user scribbles in the area.

Parameters

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

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

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

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

Return Value

This function returns nothing.

[8.3.3 GetImage](#)

Description

Stores the drawing in the scribbled area as an image in the specified location.

Parameters

bstrPath (In) – BSTR - Complete file Path where the image is to be stored.

Return Value

This function returns nothing.

[8.3.4 SetBitmap](#)

Description

Sets the bitmap image to be displayed on the ePad-Ink device.

Parameters

bstrPath (In) – BSTR - Complete file Path where the image is present.

Layer (In) – Integer – Layer in which bitmap is to be set.

ScreenID (In) – Integer – Identifier for the current screen. The value set here will be returned in the OnHandleData event to identify the active bitmap.

Return Value

This function returns nothing.

Note: The bitmap image should be flipped vertically as shown in the sample applications.

8.3.5 CreateScreen

Description

This method allows you to create screens based on predefined templates where you can specify type of the screen, button type, text to be displayed, font size and the identifier for the screen. When displaying text, number of characters per screen is dependent on the font size selected. The approximate number of characters for each font size type is provided in the tables below.

Parameters

ScreenType (In) – Integer – To specify one of the two screen types, Text-only and Input. Input screen type can be used to accept responses in addition to passing text on the ePad-Ink.

Available Screen Types:

ST_TEXTONLY = 0,
ST_INPUT = 1

ButtonType (In) – Integer – To specify the type of button to be displayed.

Available Button Types:

BT_OK = 0,
BT_OK_CANCEL = 1,
BT_OK_CLEAR = 2,
BT_OK_CANCEL_CLEAR = 3,
BT_YES_NO = 4,
BT_YES_NO_CANCEL = 5,
BT_NEXT = 6,
BT_PREV_NEXT = 7,
BT_PREV_NEXT_CLEAR = 8,
BT_ACCEPT_DECLINE = 9,
BT_NONE = 10

Text (In) – Integer – Text to be displayed on the ePad-Ink device.

FontSize – Font size of the text to be displayed can be of three types viz., Small, Medium and Large. Refer to the table below for more information.

ScreenID – Integer – To specify an identifier for the screen.

Return Value

0 – Failure, 1- Success.

Number of characters per screen when text is displayed

Text-only Screen type

FontSize	No. of lines	Approx. String length
Small	11	250
Medium	7	98
Large	5	50

Input Screen type

FontSize	No. Of lines	Approx. String length
Small	5	115
Medium	3	42
Large	2	20

8.3.6 OpenConnection

Description

Activates ePad-Ink and establishes the connection between the application and the ePad-Ink device.

Parameters

No parameters.

Return Value

Returns True if connection succeeds else False.

8.3.7 CloseConnection

Description

Closes the connection between the application and the Ink Pad device.

Parameters

No parameters.

Return Value

This function returns nothing.

8.3.8 Clear

Description

Clears the ePad-Ink device screen.

Parameters

Layer (In) – Integer – Bitmap layer to be cleared.

Color (In) – Integer – Color to be set when cleared. 0 – Black, 1 – White.

Return Value

This function returns nothing.