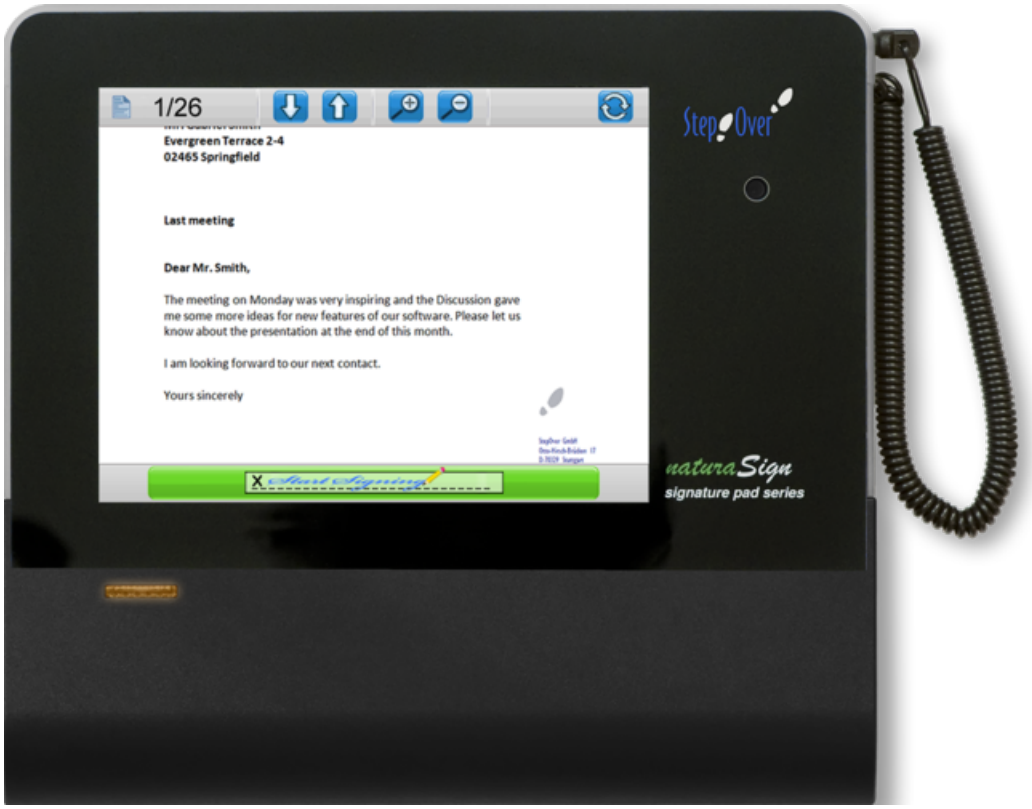
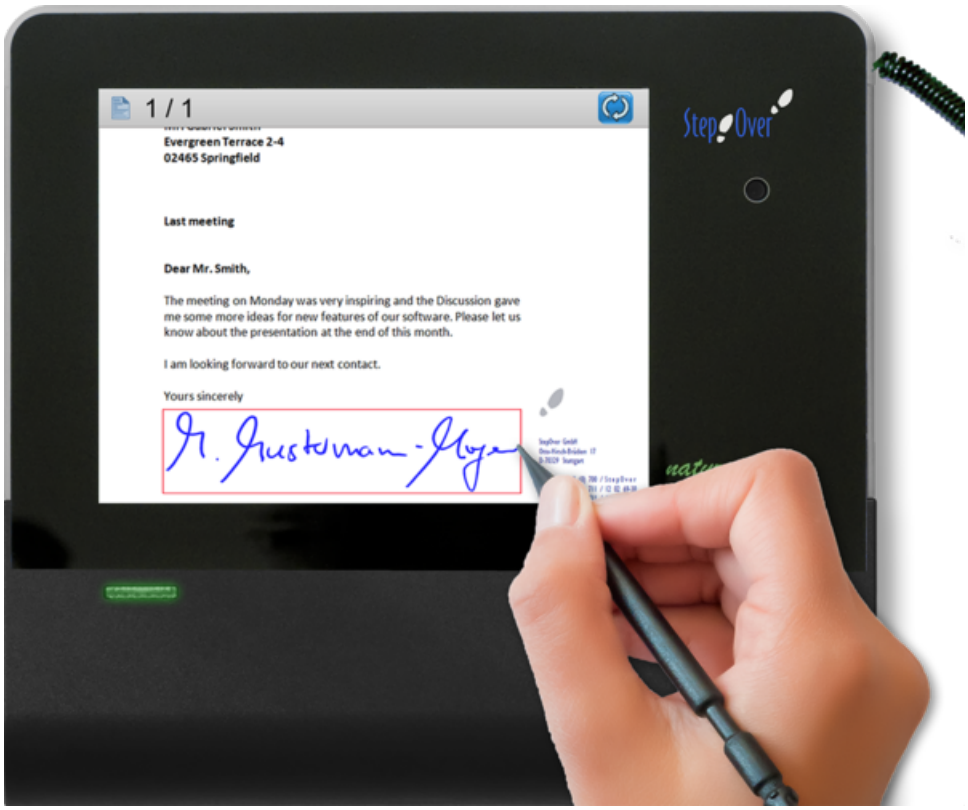
2. How to start

Introduction

The StepOver Signature API is an extension to the Signature Device Driver and allows to save the signature into a PDF document. This includes the signature image, the biometrical data, which will be linked to the document with HASH algorithms (recommended by Germany´s Federal Office of Information Security) and asymmetric encryption, which prevents manipulation and theft. The SignAPI requires a license for every workstation, on which it´s used. For testing purposes you can use it as demoversion, which shows a demo notification when loading a new document and a stamp over the signature image.

Another feature of the SignAPI is the document handling for the StepOver signature pads with colour displays. These devices support the DocumentViewing mode, in which you can see the document on the device display as well as the SignInDocument mode, in which you see the document area close to the signature position while signing on the device. Because the signing devices cannot handle PDF files directly, the SignAPI renders the PDF pages to images and sends them to the device. This means every time a user is clicking on a device button, the API reacts and sends a new image (for example with the next page) to the device.





Of course the SignAPI can also be used to check signed documents.

Linking the SignAPI with the DeviceAPI

To embed the signature and to render the document the 2 APIs need to be linked with each other. To add a signature 4 callback events (OnGetAesKey, OnGetDeviceCertificate, OnGetSignedDocHash, OnReadHighResBitmap) need to be implemented, which allow the SignAPI to access the signature data. The button handling also has to be communicated between the 2 APIs, because the DeviceAPI collects all button events (with OnDeviceButtonEx), which need to be forwarded to the SignAPI (OnDeviceLCDButtonHandler) for possible DocumentViewing mode actions. Within the SignAPI is a property named SignatureDevice, where the DeviceAPI objects need to be set.

Event	Information
OnGetAesKey	Required to link biometric data with the document
OnGetDeviceCertificate	Collects the device certificate
OnGetSignedDocHash	Required for the document hash dialog on the signature pad
OnReadHighResBitmap	Collects the signature image for the document

Loading Document and Documentviewing

The signing process starts with loading a document by using LoadDoc. Before doing this, you can set the temporary folder (SetTempPath), to where the SignAPI saves the working copys. LoadDoc returns *true*, if the document was loaded or *false* in case of an error. Most error cases occur due to an incomplete installation of the SignAPI, so please be sure that all files are installed and the *.ocx files are registered correctly. (Don´t forget the StepOverRegisterEngine.ocx!). If the error persists and the installation is correct, you should create log files and contact the StepOver Support (see 10. Troubleshooting). You can also check different properties of the loaded document.

Function Name	Information
LoadDoc	Function to load a PDF document into the SignAPI.
SetTempPath	Function to set the temp folder. By default the location is the user temp folder (%temp%)
GetPageNum	Function, which returns the total number of pages in the loaded document.

GetSigNum	Function, which returns the number of signatures, which are already in the loaded document.
GetPdfFormFieldsXml	Function to read formular values from the document.
SetPdfFormFieldValue	Function to set formular values in the document.
ViewDocOnDevice	Function to display the document on the device
ViewDocOnDeviceDefaultZoom	Function to display the document on the device, including the option to set zoom level
GetPage	Function which returns you a rendered pdf page of the loaded document

If you have a signature pad with colour display, you can use the DocumentViewing mode to show the document on the signature device. There are 2 functions to use the DocumentViewing mode (ViewDocOnDevice and ViewDocOnDeviceDefaultZoom) and the only difference between them, is just the option to set a starting page. The DocumentViewing mode also supports a button to start the SigningMode, which allows the user to trigger the signing procedure directly from the signature pad (Event OnLCDSignButton).

Signing the document

The placement specification of a signature needs to be done with PDF coordinates. You need the top/left as well as the botton/right corner and also the page number. It´s possible to convert centimeters/inches to PDF points: usually a PDF file has 72 points per inch, which means the size of a normal PDF page is normally 595x842 and the top/left corner has the value 0x0. You need the PDF coordinates for the different AddSignature functions and if want to use the SignInDocument Mode also for the SetSignDocMode function (the values should be the same, otherwise the signature will not be at the same position, which the signer saw while signing on the pad).

If the documents are dynamic and it´s not possible to use static coordinates, the function SearchKeyString can be use to find an ancor point inside the document. The allows you to read a specific string in the document, but for this, the PDF must contain readable text and may not be a scanned file. The return value of the function is an array with all positions of the searched string in the document. This also works for "invisible" text (white text on white background) or very small text. The idea is that you search for a very specific string, which is close to the signature fields, to calculate the values you need for you signature placement.

For adding a Signature you need to activate the Signature Mode of the device with the DeviceAPI function startCapture, when the signier is finished the signature can be added to the document. For that you need to call the AddSignature or AddSignatureField (in case you want to store the signature into an exsiting Signature Field). In case the Return value is not 1, you need to check if you added all the Callback Events and if the placement informations are correct. The function AddSignatureImage allows you to insert an image into the PDF document, this image can be Stamp or you can use this function to add additional informations into the document. The image is added as a Signature, which means you can use this functionallty also with an already signed document without making the exsiting signature invalid.

Function Name	Information
SearchKeyString	Function, which returns all coordinate sets for a searched string in the document.
GetSigFields	Function, which returns the coordinates of all signature fields (signed and unsigned). The unsigned field coordinates can be used for SetSignDocMode.
SetSignDocMode	Function to set the signing postion in the document for the SignInDocument Mode (needs to be called before startCapture).
SetLcdSignZoom	Function to change the zoom level of the SignInDocument Mode (needs to be called before SetSignDocMode).
SetSignatureFrameParams	Function to change the configuration of the signature rectangle for the SignInDocument Mode (needs to be called before SetSignDocMode).
AddSignature	Function to add a signature to the PDF document. This function will insert a new SignatureField into the document.
AddSignatureField	Function to add a signature to an empty signature field (In which case the placement with coordinates is not nessecary).
AddSignatureImage	Function to add an image as signature to the document.
TimeCenterURL	Property to define a TimeStampServer to get time stamp for the signature (by default the local system time is used).
DeviceModel	Property to set the Device Name (visible in the DigitalSignature).
DeviceSerial	Property to set the Device Serial (visible in the DigitalSignature).

Saving and closing a document

After the last signature, the signature pad can be set back to the StandBy Mode by using the DeviceAPI function stopRead and the document can be saved.

The SignAPI has two function to save the document: SaveDoc will save the loaded document to its original path and filename, while SaveDocTo saves the signed document to another path and with a new filename.

The last step is to close the document with CloseDoc, which will release the document from the SignAPI and will erase temporary file.

Function Name	Information
SaveDoc	Function to save the document to it's original location.
SaveDocTo	Function to save the document to a different location.
CloseDoc	Function to unload the document from the SignAPI.